

# Zuverlässige numerische Berechnungen mit dem Spigot-Ansatz

## Dissertation

zur Erlangung des akademischen Grades  
Doktoringenieur (Dr.-Ing.)

vorgelegt an der  
Technischen Universität Dresden  
Fakultät Informatik

eingereicht von  
**Dipl.-Inform. Dang-Khoa Do**  
geboren am 16. Juli 1970 in Thai Binh

Gutachter: Prof. Dr.-Ing. habil. Erwin P. Stoschek  
Technische Universität Dresden  
Prof. Dr. rer. nat. habil. Immo O. Kerner  
Technische Universität Dresden  
Prof. Dr. rer. nat. habil. Karl Hantzschmann  
Universität Rostock

Tag der Verteidigung: 20.09.2005  
Dresden im September 2006



# Danksagung

Vor allem bedanke ich mich herzlichst bei meinem Doktorvater, Herrn Prof. Dr. -Ing. habil. Stoschek, der die interessante Problematik des Spigot-Algorithmus als Gegenstand der Dissertation vorgeschlagen hat, mir bei der konkreten Ausgestaltung des Themas weitergehende Freiheit gewährt hat, meine Begeisterung und mein Selbstvertrauen durch anspruchsvolle Aufgaben der Reihentransformation geweckt hat, meine Ideen durch zahlreiche Gespräche und Diskussionen gefördert hat und mich durch seinen Optimismus stets ermutigte.

Herrn Prof. Dr. rer. nat. habil. Hebenstreit und Herrn Prof. Dr. rer. nat. habil. Kerner danke ich ganz herzlich dafür, dass sie mich durch ihre Interessen an meiner Forschung motiviert haben und mir viele nützliche Hinweise und Anregungen bezüglich meiner ersten wissenschaftlichen Publikation in einer international referierten Zeitschrift gegeben haben.

Herrn Prof. Dr. sc. nat. Gerhard Maeß danke ich für die Möglichkeit eines mathematischen Kolloquiums an der Universität Rostock, wodurch ich Teilergebnisse meiner Forschung präsentieren konnte.

Der Redaktion und den Gutachtern der internationalen Zeitschrift *Reliable Computing* danke ich für viele wertvolle, wegweisende Hinweise im Hinblick auf einen angemessenen Stil in der wissenschaftlichen Arbeit.

Meine Promotion wurde mit einem Stipendium des Freistaates Sachsen finanziell gefördert, wodurch die wesentlichen Ergebnisse der Dissertation sowie meine erste Publikation entstanden sind. Dafür möchte ich mich bedanken.

Die Anfertigung der Dissertationsschrift erfolgte während einer Beschäftigung am *Institut für Thermodynamik und Technische Gebäudeausrüstung der TU Dresden*. Die großzügige Unterstützung durch die Kollegen und das ausgezeichnete Arbeitsklima hier sind entscheidend für den Erfolg. Ich möchte mich bei dem Leiter und den Kollegen des Instituts herzlich bedanken. Mein besonderer Dank gilt Dr.-Ing. Tobias Zschunke als Leiter der Arbeitsgruppe. Er versteht es ausgezeichnet, seine Mitarbeiter zu leiten und zu motivieren, um persönliche und kollektive Ziele in Einklang zu bringen und beide optimal zu erreichen. Mein besonderer Dank richtet sich auch an Herrn Kühne von der Fakultät Verkehrswissenschaft als fakultätsübergreifender Organisator meiner Projektarbeit am Institut für viele organisatorische Unterstützungen.

Ich möchte den liebevollen Dank gegenüber meinem Vater Tham-Hoa Do und meiner Mutter Thi-Sam Dao zum Ausdruck bringen. Mit allen notwendigen Anstrengungen und Überwindung aller Schwierigkeiten haben meine Eltern mir eine sorgenfreie Kindheit ermöglicht sowie meine Interessen und Begabungen für Mathematik von Kindheit an gefördert. Meine liebe Mutter ist kurz vor der Fertigstellung der Dissertationsschrift gestorben, dennoch habe ich von ihr für die Arbeit noch viel geistige Energie bekommen. Zu Lebzeiten hat sie meine Interessen für Mathematik insbesondere durch die Bücher von G. Polya gefördert.

Ich bin vielen Personen dankbar, die daran beteiligt waren, mir die sprachlichen Mängel in den Entwurfsversionen der Dissertation kenntlich zu machen und Verbesserungsvorschläge zu geben. Das sind Eva Wagner, Dr.-Ing. Jens Meinert, Mike Neuman, Dr.-Ing. Thomas Hinze und Steffen Simon. Sie haben als Muttersprachler bzw. Sachverständiger die Arbeit durchgesehen und mir viele wertvolle Hinweise gegeben.

Bei meinen Geschwistern, bei vielen Freunden und besonders bei meiner Geliebten Thu-My Le (die inzwischen meine Frau geworden ist) bedanke ich mich für die emotionalen Unterstützungen während der Anfertigung der Dissertationsschrift. Der liebevolle Umgang von Thu-My hat mir

es ermöglicht, nach dem Tod meiner Mutter das emotionale Gleichgewicht allmählich wiederzufinden sowie meinen Stress abzubauen.

Mein herzlicher Dank richtet sich natürlich auch an die Gutachter für ihr Interesse an meiner Forschung, dafür, dass sie sich Zeit und Geduld für die Begutachtung der Dissertation genommen haben und dafür, dass sie mir wertvolle Hinweise gegeben, mit denen ich einige im Begutachtungsexemplar vorkommende Mängel nun aufheben kann. Mit den Hinweisen von Herrn Prof. Kerner werden Beispiel 2.79 um die Überwachung der tatsächlich größten benötigten Zahl ergänzt sowie Satz 2.78 und Lemma 2.81 korrigiert. Mit den Hinweisen von Herrn Prof. Hantschmann werden einige Flüchtigkeitsfehler beseitigt (insbesondere bezüglich Beispiel 4.27) und Abschnitt 7.3 aktualisiert.

Ich werde jedem Leser für sein Interesse an der vorliegenden Dissertation dankbar sein. Bemerkungen, Kritiken, Verbesserungsvorschläge, Anregungen und Hinweisen auf Abhandlungen mit ähnlichen, verwandten und weiterführenden Inhalten bzw. über alternative Vorgehensweisen sind an die E-Mail-Adresse [kdspigot@web.de](mailto:kdspigot@web.de) willkommen und sehr erwünscht.

# Vorwort

Der *Spigot-Ansatz* ist eine elegante Alternative, numerische Werte zuverlässig, effizient und mit beliebiger Genauigkeit zu berechnen. Der *Spigot-Algorithmus* ist dabei der Kern des Spigot-Ansatzes.

Das Paradigma des Spigot-Ansatzes lautet: die symbolisch-algebraische Transformation so weit wie möglich einzusetzen und numerische Berechnungen erst im letzten Schritt ohne Rundung auszuführen. Dementsprechend wird zuerst für den zu berechnenden numerischen Wert eine gut konvergierende Reihe mit rationalen Gliedern durch symbolisch-algebraische Methoden hergeleitet; dann wird für eine gewünschte Genauigkeit eine Teilsumme ausgewählt; anschließend werden aus der Teilsumme Wertanteile iterativ extrahiert, die in der Zielbasis „kodiert“ sind. Die Extraktion wird mit elementarer Integer-Arithmetik ausgeführt und der absolute Fehler kann exponentiell in Abhängigkeit von der Anzahl der Iterationen beschränkt werden.

Im Kontext des Spigot-Ansatzes hat „Spigot“ die Bedeutung „Berechnung durch sukzessive Extraktion von Wertanteilen“: die Wertanteile werden extrahiert, als ob sie durch einen Hahn (englisch: spigot) gepumpt werden. Diese Vorgehensweise gilt für beide Teilprozesse der Berechnung, sowohl bei der Herleitung einer Reihe für den zu berechnenden Wert als auch bei der Extraktion der in einer gewünschten Zielbasis kodierten Wertanteile aus der Reihe. Der erste Teilprozess erfordert schöpferische Arbeit, dabei sind mathematisch interessante und tiefgreifende Ansätze der Reihenentwicklung und Reihentransformation — gegebenenfalls auch mit Computerunterstützung — einzusetzen. Der zweite Teilprozess lässt sich mit dem (numerischen) Spigot-Algorithmus ausführen, der in der vorliegenden Dissertation formalisiert und in das Konzept des Spigot-Ansatzes eingebettet wird. Der Spigot-Algorithmus ist somit der Ausgang und zugleich der Kern der vorliegenden Dissertation.

Die absolute Korrektheit numerischer Berechnungen mit dem Spigot-Ansatz kann im Prinzip formal bewiesen werden. Für die Realisierung sehr hoher Genauigkeiten sind nur elementare arithmetische Operationen mit kleinen ganzen Zahlen notwendig, die implementierungsunabhängig sind und ohne Rundung von jedem Computer ausgeführt werden können. Das Problem der Fehlerfortpflanzung, das viele theoretisch elegante Verfahren in der Praxis der numerischen Berechnung unbrauchbar machen kann, wird mit dem Spigot-Ansatz in Grenzen gehalten bzw. vollständig vermieden. *Mit dem Spigot-Ansatz erreichen maschinell ausgeführte numerische Berechnungen das Höchstmaß an Zuverlässigkeit.*

Die Idee des Spigot-Algorithmus wurde zum ersten Mal von Sale ([Sal68]) zur Berechnung der Dezimalziffern von  $e$  in der Fachliteratur veröffentlicht. Der Spigot-Algorithmus als ein allgemeines Verfahren für die numerische Summation von Reihen ist durch Abdali ([Abd70]) angegeben worden. Rabinowitz und Wagon ([RW95]) haben Spigot-Berechnungen für  $\pi$  unter der Benutzung der Ergebnisse der Reihentransformation von Gosper ([Gos74]) hergeleitet. Stoschek [Sto97, Baustein 6, Tabelle 3] hat Ideen zur Herleitung von Spigot-Berechnungen für eine Sammlung von Werten skizziert, darunter  $\sqrt{2}$ ,  $\sqrt[3]{2}$  und  $\ln 2$ . In [Sto97] findet man auch den Spigot-Algorithmus in Form einer Turing-Maschine.

Aus [Gos74] und [RW95] lässt sich vermuten, dass der Terminus "Spigot" als Indikation für die „Berechnung durch sukzessive Extraktion von Wertanteilen“ von Gosper (vergl. [Gos74, S. 10]) eingeführt wurde, der dann von Rabinowitz und Wagon ([RW95]) übernommen wird. Mit [RW95] ist der Begriff Spigot-Algorithmus (engl.: spigot algorithm) in der Fachliteratur bekannt.

Die vorliegende Dissertation stellt sich folgende Aufgaben:

1. Formalisierung des Spigot-Algorithmus, um dessen Anwendbarkeit für Reihen zu überprüfen bzw. nachzuweisen und dann bei der Anwendung Aussagen über die Genauigkeit von (maschinell ausgeführten) numerischen Berechnungen mathematisch exakt herzuleiten.
2. Systematisierung der gegenwärtigen Ansätze, Methoden und Techniken der Reihenentwicklung und Reihentransformation, um die schöpferische Arbeit der Herleitung von spigotgeeigneten Reihen zu unterstützen.
3. Herleitung von typischen und/oder interessanten Spigot-Berechnungen.
4. Sammlung der potenziellen Verknüpfungspunkte von praktisch-technischen Problemlösungen zu dem Spigot-Algorithmus.

□

Zur Realisierung der oben genannten Aufgaben ist der Inhalt wie folgt gegliedert:

- ☞ Zuerst werden im vorbereitenden Kapitel die technisch-methodischen Grundlagen aufgestellt, die eine systematische Vorgehensweise sowohl bei der Ausarbeitung als auch bei der Darstellung des Inhalts der vorliegenden Dissertation unterstützen sollen.
- ☞ Kapitel 1 führt die Idee des Spigot-Algorithmus intuitiv ein, legt die Motivation für eine Weiterentwicklung dar und führt den grundlegenden begrifflichen Zusammenhang Spigot-Algorithmus/Spigot-Berechnung/Spigot-Ansatz aus.
- ☞ In Kapitel 2 werden die formalen Grundlagen für den effektiven und korrekten Einsatz des Spigot-Algorithmus geschaffen, wodurch aus dem Spigot-Algorithmus eine Methode für die numerische Berechnung von Reihen mit beliebiger Genauigkeit und mathematisch exakt begründeter Zuverlässigkeit gewonnen wird.
- ☞ Kapitel 3 beinhaltet die Beispiele für die Reihenentwicklung mit Hilfe charakteristischer Funktionalgleichungen.
- ☞ Kapitel 4 systematisiert die klassischen Grundbausteine für die Manipulation von Reihen und Folgen nach einem möglichst einheitlichen Gesichtspunkt.
- ☞ Kapitel 5 systematisiert die Methoden der Reihentransformation von Markoff, Zeilberger und Wilf nach einem einheitlichen Gesichtspunkt.
- ☞ Kapitel 6 formalisiert die Methode der Reihentransformation von Gosper.
- ☞ Kapitel 7 entwickelt eine Methode für die Wurzelberechnung nach dem Spigot-Ansatz.
- ☞ Kapitel 8 entwickelt eine Methode für die Logarithmenberechnung nach dem Spigot-Ansatz.
- ☞ Der Ausblick in Kapitel 9 führt Schlussbemerkungen zu den Ergebnissen der vorliegenden Dissertation, einige Möglichkeiten der Weiterentwicklung des Spigot-Ansatzes, sowie einige Verknüpfungspunkte des Spigot-Ansatzes zu anderen Problemkreisen aus.

- 
- ☞ Anhang A systematisiert die wichtigsten, gut bekannten Kenntnisse über reelle Zahlen und Reihen mit Literaturverweis und ohne Herleitung, um formale Bezugspunkte für die Herleitung und Transformation von Reihen bereitzustellen.
  - ☞ Anhang B führt einige Einzelheiten im Umgang mit Maple und der Technik des Beweisans aus. □

Wesentlich neue Forschungsergebnisse sind in Kapitel 2, Kapitel 7 und Kapitel 8 enthalten. Die Ideen und Methoden aus Kapitel 3 können eventuell als Ausgangspunkt für die Entwicklung neuer Methoden zur Berechnung von Funktionen benutzt werden. Mit Kapitel 4, Kapitel 5 und Kapitel 6 werden die Ansätze und Methoden der Reihentransformation gründlich systematisiert. Diese Systematisierung kann eventuell als eine Grundlage für die Computerisierung der Reihentransformation benutzt werden.

Es wurde angestrebt, sowohl in der Ausarbeitung als auch in der Darstellung der Dissertation systematisch und anschaulich vorzugehen. Dies betrifft insbesondere die Symbolik und die Beweisführung. Sowohl mathematische Strenge als auch Anschaulichkeit werden zum Ziel gesetzt.

# Inhaltsverzeichnis

<b>Technisch-methodische Grundlagen</b>	<b>1</b>
0.1 Abkürzungen . . . . .	1
0.2 Vereinheitlichung der Schreibweisen von mathematischen Begriffen . . . . .	1
0.3 Bezeichnungen . . . . .	3
0.4 Gleichheit . . . . .	4
0.5 Folgennotation . . . . .	5
0.6 Definieren, Spezifizieren und Herleiten . . . . .	6
<b>I Der Spigot-Algorithmus als der Kern des Spigot-Ansatzes</b>	<b>9</b>
<b>1 Einleitung</b>	<b>10</b>
1.1 Die Zahl $e$ und die Entdeckung des Spigot-Algorithmus . . . . .	10
1.1.1 Die naive Methode und ihre Mängel . . . . .	10
1.1.2 Die Idee der Spigot-Berechnung . . . . .	11
1.2 Besonderheit der Spigot-Berechnung von $\pi$ . . . . .	15
1.3 Motivation zur Weiterentwicklung . . . . .	17
<b>2 Formale Grundlagen für den Einsatz des Spigot-Algorithmus</b>	<b>20</b>
2.1 FNEs und der Spigot-Algorithmus . . . . .	20
2.1.1 Struktur, Notation, Manipulation und Wert-Eigenschaften von FNEs . . . . .	21
2.1.2 FNEs . . . . .	21
2.1.3 Normierte Form von FNEs . . . . .	28
2.1.4 Der Normierungsalgorithmus und seine Korrektheit . . . . .	31
2.1.5 Spigot-Approximationen und Spigot-Algorithmus . . . . .	35
2.1.6 Spigot-Algorithmus . . . . .	36
2.1.7 Obere Schranke der Zahlen bei der Ausführung des Spigot-Algorithmus . . . . .	41
2.2 INEs und Spigot-Berechnung reeller Zahlen . . . . .	44
2.2.1 Struktur, Konvergenzverhalten und Manipulation von INEs . . . . .	45



2.2.2	Formale Werkzeuge für die Untersuchung und Bewerkstelligung der Eigenschaften von Spigot-Berechnungen durch INE . . . . .	47
2.3	Ressourcenverbrauch der Spigot-Berechnung . . . . .	53
2.4	Variationen des Spigot-Algorithmus . . . . .	55
2.5	Formalisierung der Spigot-Berechnung von $\sqrt{2}$ nach Stoschek . . . . .	57
2.5.1	Die zugrunde liegende INE, ihre Restschranke und Spigot-Schranke . . . . .	57
2.5.2	Einfache numerische Berechnung zur Basis 10 . . . . .	59
2.5.3	Berechnung mit Potenzüberbrückung zur Basis 10 . . . . .	61
2.6	Zusammenfassung . . . . .	62
<b>II</b>	<b>Techniken der Herleitung und Beschleunigung von Reihen</b>	<b>63</b>
<b>3</b>	<b>Reihenentwicklung durch charakteristische Funktionalgleichung</b>	<b>64</b>
3.1	Exponentialreihe . . . . .	65
3.2	Logarithmische Reihe . . . . .	66
3.3	Die Funktion $\arctan$ und die Gregory-Leibniz-Reihe für $\pi$ . . . . .	69
3.4	Binomialreihe . . . . .	70
3.5	Reihen mit durch Bernoullische Zahlen dargestellten Koeffizienten . . . . .	72
<b>4</b>	<b>Grundbausteine für die Manipulation von Folgen und Reihen</b>	<b>74</b>
4.1	Differenz . . . . .	75
4.2	Doppelreihe . . . . .	78
4.2.1	Grundschemata und Varianten der Konvergenzbedingung . . . . .	78
4.2.2	Beispiele . . . . .	80
4.3	Wertverschiebung . . . . .	83
4.3.1	Zweistellige Durchschnittsbildung . . . . .	83
4.3.2	Wertverschiebung als Verallgemeinerung der Durchschnittsbildung . . . . .	85
4.3.3	Kummersche Transformation als Spezialfall der Wertverschiebung . . . . .	86
<b>5</b>	<b>Markoffsche Differenzrechnung und WZ-Methode</b>	<b>90</b>
5.1	Markoffsche Rechtecktransformation . . . . .	91
5.2	Zeilbergersche Variante der Dreieckstransformation . . . . .	94
5.3	Wilfsche Methode . . . . .	97
5.4	Über Gosper- und Zeilberger-Algorithmus . . . . .	101
5.5	Diskussion . . . . .	103
<b>6</b>	<b>Gospersche Reihentransformation</b>	<b>105</b>
6.1	Formalisierung der Gosperschen Methode . . . . .	106

6.1.1	$\mathcal{R}$ -Ausdruck . . . . .	106
6.1.2	Spaltungsfunktion und schrittweise Extraktion von Wertanteilen aus einer Reihe . . . . .	109
6.1.3	Kombinationsquotient und explizite Formel für Extraktionssumme . . . . .	111
6.1.4	Anwendungsformen und Anwendungsheuristik . . . . .	113
6.2	Anwendungsbeispiele . . . . .	116
6.2.1	Reihen für $\pi$ . . . . .	116
6.2.2	Reihen für $\zeta(2)$ . . . . .	117
6.2.3	Reihen für $\zeta(3)$ . . . . .	118
<b>III Anwendung und offene Probleme</b>		<b>119</b>
<b>7</b>	<b>Wurzelberechnung</b>	<b>120</b>
7.1	Eigenschaften des Ausdrucks $\frac{p}{q}(1 - \frac{u}{v})^{-\frac{1}{k}}$ mit $0 < u < v \in \mathbb{N}$ . . . . .	121
7.2	Überführung von $\sqrt[k]{\frac{r}{s}}$ in $\frac{p}{q}(1 - \frac{u}{v})^{-\frac{1}{k}}$ . . . . .	122
7.2.1	Zurückführung der Berechnung eines Hilfsfaktors auf die Berechnung von $\lfloor \sqrt[k]{n} \rfloor$ . . . . .	123
7.2.2	Die Berechnung von $\lfloor \sqrt[k]{n} \rfloor$ mit $n \in \mathbb{N}$ . . . . .	125
7.3	Schlussbemerkungen und numerische Beispiele . . . . .	128
<b>8</b>	<b>Logarithmenberechnung</b>	<b>131</b>
8.1	INEs, Restschranken und Spigot-Schranken für Logarithmenberechnung . . . . .	132
8.2	Spigot-Berechnung von $k \times \ln 2$ und $\ln \left(1 + \frac{p}{q}\right)$ . . . . .	135
8.3	Prozeduren zur Berechnung von $\ln Q$ . . . . .	136
8.4	Schlussbemerkungen und numerische Beispiele . . . . .	139
<b>9</b>	<b>Ausblick</b>	<b>140</b>
<b>A</b>	<b>Grundlegendes über infinitesimale Rechnungen und Potenzreihen</b>	<b>142</b>
A.1	Die Vollständigkeit der Menge $\mathbb{R}$ . . . . .	142
A.2	Konvergenzverhalten von Folgen und Reihen reeller Zahlen . . . . .	143
A.3	Konvergenz von Funktionenfolgen und Funktionenreihen . . . . .	148
A.4	Konvergenzverhalten von Potenzreihen . . . . .	149
A.5	Das Rechnen mit Potenzreihen . . . . .	151
<b>B</b>	<b>Verschiedenes</b>	<b>153</b>
B.1	Beispiel für den Beweis von Nebenbedingungen . . . . .	153
B.2	Grundlegendes für die Benutzung des Gosper-Algorithmus unter Maple . . . . .	153

---

B.3 Grundlegendes für die Benutzung des Zeilberger-Algorithmus unter Maple . . . .	154
<b>Literaturverzeichnis</b>	<b>156</b>
<b>Symbolverzeichnis</b>	<b>159</b>
<b>Index</b>	<b>162</b>

# Verzeichnis der Herleitungen

1.1	Überführung von $e_7$ in eine geschachtelte Summe . . . . .	12
1.2	Die erste Runde der Extraktion dezimaler Wertanteile aus $e_7$ . . . . .	13
1.3	Die zweite Runde der Extraktion dezimaler Wertanteile aus $e_7$ . . . . .	14
1.4	Die erste Runde der Extraktion dezimaler Wertanteile aus $\pi_9^{\text{GLE}}$ . . . . .	16
2.1	Einzelne Schritte der Konstruktion der FNE für $e_7$ laut Definition 2.1 . . . . .	23
2.2	Das Erhalten des Wertes der Teilsumme an der Position $pos - 1$ nach der Versetzung des Übertrags an der Position $pos$ . . . . .	29
2.3	Das Erhalten der Beziehung zwischen Approximationssumme und Restausdruck durch Induktion . . . . .	37
2.4	Zum Abschluss des Beweises der grundlegenden Invarianz des Spigot-Algorithmus	38
2.5	Die ine-Darstellung $\overset{\text{sim}}{\text{INE}}_{\sqrt{2}}$ für $\sqrt{2}$ . . . . .	58
2.6	Die ifs-Restschranke $\overset{\text{sim}}{\underset{\text{r-ifs}}{\Delta}}(\overset{\text{sim}}{\text{IFS}})_{\sqrt{2}}(n)$ . . . . .	59
4.1	Geschlossene Summation der Reihe $\sum_{n=0}^{\infty} (n+1)x^n$ mit dem Ansatz der Doppelreihe	80
4.2	Zur Verbesserung der Konvergenz von $\zeta(2)$ mit dem Ansatz der Doppelreihen .	81
4.3	Das Schema der Doppelreihe für die allgemeine Form der Eulerschen Transformation	82
4.4	Zur Umformung der Reihe für $\ln 2$ in (3.22) mit zweistelliger Durchschnittsbildung	85
6.1	Umformung eines endlichen $\mathcal{R}$ -Ausdrucks . . . . .	108
6.2	Ein Schritt der Gosperschen Extraktion . . . . .	110
6.3	Termquotienten und Kombinationsquotienten . . . . .	112
6.4	Überführung der Gosperschen Extraktionssumme in einen $\mathcal{R}$ -Ausdruck . . . . .	113
7.1	Eine Restschranke für (7.2b) . . . . .	122
8.1	Eine Restschranke für (8.2b) . . . . .	133

# Algorithmenverzeichnis

1.1	Extraktion der Dezimalziffern von $e_7$ . . . . .	15
1.2	Extraktion der dezimalen Wertanteile von $\pi_9^{\text{GLE}}$ . . . . .	17
2.1	Normierungsalgorithmus . . . . .	34
2.2	Extraktionsalgorithmus . . . . .	39
2.3	Ergebnisnormierungsalgorithmus . . . . .	40
2.4	Spigot-Algorithmus mit Einbeziehung der Unterprogramme . . . . .	41
2.5	Spigot-Algorithmus . . . . .	42
2.6	Spigot-Algorithmus mit Potenzüberbrückung . . . . .	55
2.7	Potenzrückführungsalgorithmus . . . . .	56
7.1	Berechnung von $\lfloor \sqrt{n} \rfloor$ mit Heron's Iteration . . . . .	126
7.2	Berechnung von $\lfloor \sqrt[k]{n} \rfloor$ mit Newton's Verfahren . . . . .	128
8.1	Berechnung von $\lfloor \log_2 Q \rfloor$ . . . . .	137

# Tabellenverzeichnis

1	Zur Vereinheitlichung der Symbolik. . . . .	2
1.1	Naive Berechnung von $e$ mit dem Genauigkeitsmaß $\epsilon$ auf der Grundlage von(1.1)	11
7.1	Spigot-Berechnung von $\sqrt[k]{\frac{101}{2}} \mid 2 \leq k \leq 5$ bis zu 20 Nachkommaziffern. . . . .	129
7.2	Verschiedene BINE für die Berechnung von $\sqrt{2}$ nach Schema 7.13 . . . . .	130
8.1	Numerische Beispiele der Berechnung von $\ln Q = k \times \ln 2 + \ln(1 + \frac{p}{q})$ mit absolutem Fehler kleiner als $\frac{1}{10^l}$ für $l = 40$ . . . . .	139

# Technisch-methodische Grundlagen

In diesem vorbereitenden Kapitel werden die technisch-methodischen Grundlagen aufgestellt, die eine systematische Vorgehensweise sowohl bei der Ausarbeitung als auch bei der Darstellung des Inhalts der vorliegenden Dissertation unterstützen sollen.

## 0.1 Abkürzungen

Die Wirkung des Einsatzes von Abkürzungen ist bekannt. Im Folgenden sind die Regeln für die Bildung und Handhabung von Abkürzungen in der vorliegenden Dissertation vereinbart.

*Bemerkung 0.1* (Abkürzungen). Eine Abkürzung wird in der Regel anhand des vollständigen englischen Fachausdrucks gebildet (um „internationale Aussagekraft“ zu gewährleisten) und ist grammatisch wie folgt zu handhaben:

1. Das Genus wird aus dem vollständigen Ausdruck in der deutschen Sprache übernommen.
2. Alle Kasi haben die gleiche Form.
3. Der Plural wird durch das Anhängen eines „s“ gebildet.

Abkürzungen können in fetter oder normaler Schrift erscheinen, stehen aber immer **gerade**, um sich typographisch von Variablen zu unterscheiden.  $\square$

## 0.2 Vereinheitlichung der Schreibweisen von mathematischen Begriffen

Die Symbole für einige allgemein bekannte mathematische Begriffe weisen in der Fachliteratur prinzipielle Ähnlichkeiten, aber auch kleine Unterschiede auf. Tabelle 1 soll deshalb dazu dienen, eine einheitliche Symbolik für die vorliegende Dissertation festzulegen. Symbole, die schon in der Fachliteratur einheitlich erscheinen, z. B. + oder  $\leq$ , sind in der Tabelle nicht ausgeführt.

*Bemerkung 0.2* (Trennungszeichen zwischen Ausdruck und zusätzlicher Information). Als Trennungszeichen wird das Zeichen | benutzt, z. B.:  $[c_i]_{i=k}^{k+n} \mid c_i \in \mathbb{Z}$ . Zum selben Zweck wird wahlweise auch das Wort 'mit' eingesetzt. z. B.:  $\ln(1 + \frac{p}{q})$  **mit**  $0 < p \leq q \in \mathbb{Z}$ .  $\square$

<b>Zahlenmengen</b>	
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{Z}$	Menge der ganzen Zahlen
$\mathbb{Z}^+$	Menge der positiven ganzen Zahlen
$\mathbb{Q}$	Menge der rationalen Zahlen
$\mathbb{Q}^+$	Menge der positiven rationalen Zahlen
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}^+$	Menge der positiven reellen Zahlen
<b>Arithmetische Funktionen</b>	
$x \div y$	Ganzzahlige Division
$x \bmod y$	Ganzzahlige Restbildung
$\times$	Multiplikation
$\lfloor x \rfloor$	Ganzer Teil von $x$ , die größte ganze Zahl, die $x$ nicht übersteigt
$\{x\}$	Gebrochener Teil von $x$ , $\{x\} \equiv x - \lfloor x \rfloor$
$\lceil x \rceil$	Die kleinste ganze Zahl größer oder gleich $x$ , $x$ aufgerundet
<b>Mengen</b>	
$x \in X$	$x$ ist Element von $X$
$X \subset Y$	$X$ ist Teilmenge von $Y$
$X \times Y$	Kartesisches Produkt der Menge $X$ und der Menge $Y$
$X \setminus Y$	Differenz der Menge $X$ und der Menge $Y$
<b>Indexmengen</b>	
$\mathbb{Z}_{\geq i_0}$	$\{z \in \mathbb{Z} \mid z \geq i_0\}$
$\mathbb{Z}_{i_0 \rightarrow i_1}$	$\{z \in \mathbb{Z} \mid i_0 \leq z \leq i_1\}$
<b>Aussagen</b>	
$\Rightarrow$	Logische Implikation
$\forall$	Für alle
<b>Abbildungen</b>	
$f : X \longrightarrow Y$	$f$ ist eine Funktion der Menge $X$ in die Menge $Y$
$x \longmapsto y$	Das Element $y$ ist das Bild des Elements $x$ durch eine Abbildung
<b>Algorithmen</b>	
$:=$	Wertzuweisung
<b>Sonstiges</b>	
$E C$	Ausdruck   Nebenbedingung
$\underbrace{(\square)}_{\text{EqE}}$	Gleichungsausdruck

Tabelle 1: Zur Vereinheitlichung der Symbolik.

*Bemerkung 0.3* (Gleichungsausdruck). Manchmal ist es vorteilhaft, für einen Teilausdruck eine nummerierte Gleichung einzubeziehen. Unter anderem wird dadurch die Darstellung kürzer. Dieser Mechanismus ist analog zu dem des Aufrufs von Unterprogrammen in der Programmierung.

Man betrachte als Beispiel den Ausdruck  $\text{APR}_{\text{ine}}^{\text{fne}} \left( \underbrace{(2.72)}_{\text{EqE}}, n \right)$ , der in (2.73) vorkommt. Dabei steht  $\underbrace{(2.72)}_{\text{EqE}}$  für den Ausdruck in (2.72). Die Referenz auf eine Gleichung zum Zweck der Benutzung als Teilausdruck wird mit  $\underbrace{(\square)}_{\text{EqE}}$  eingeleitet. Für einen konkreten Einsatz wird  $\square$  durch die



Numerierung der Gleichung ersetzt. □

*Bemerkung 0.4* (Beseitigung von Mehrdeutigkeiten durch Einbeziehen des Kontextes). Manche Symbole sind erst durch Einbeziehen des Kontextes eindeutig,  $\times$  kann z. B. wahlweise für arithmetische Multiplikation oder kartesisches Produkt von Mengen stehen. □

*Bemerkung 0.5* (Indexmengen). Mit  $\mathbb{Z}_{\geq i_0}$  wird die Menge

$$\{z \in \mathbb{Z} \mid z \geq i_0\}$$

und mit  $\mathbb{Z}_{i_0 \rightarrow i_1}$  wird die Menge

$$\{z \in \mathbb{Z} \mid i_0 \leq z \leq i_1\}$$

bezeichnet. Die beiden Mengen werden häufig als Indexmengen verwendet. □

### 0.3 Bezeichnungen

Bezeichnungen sollen aussagekräftig sein und die Einordnung der zu bezeichnenden Begriffe unterstützen. Dazu werden in diesem Abschnitt Regeln erläutert, die der vorliegenden Dissertation zugrunde liegen.

*Bemerkung 0.6* (Allgemeine Regeln für Bezeichnungen).

1. Eine Bezeichnung hat stets einen *Hauptteil*, der gegebenenfalls durch *Übersetzung*, *Untersetzung*, *Indizierung* usw. ergänzt wird.
2. Die Bestandteile einer Bezeichnung sollen möglichst als Eselsbrücken dienen. Eine Buchstabenfolge, die als Bestandteil von Bezeichnungen vorkommt, soll demnach einen relevanten Begriff andeuten und möglichst aus der entsprechenden Abkürzung abgeleitet werden.
3. Das gesamte Bild aller Bestandteile einer Bezeichnung soll möglichst der Bedeutung des zu bezeichnenden Sachverhaltes entsprechen. □

*Bemerkung 0.7* (Bezeichnung von Funktionsnamen, Operatoren und Konstanten). Die Regeln in Bemerkung 0.6 sind zu beachten. Die Buchstabenfolge des Hauptteils soll demnach möglichst den Namen oder die Bedeutung der zu bezeichnenden Funktion, Operation oder Konstanten andeuten. Außerdem soll in Anlehnung an die Konvention von L<sup>A</sup>T<sub>E</sub>X für Operatornamen der **Hauptteil in senkrechter Schrift** erscheinen. □

*Beispiel 0.8.* Um den Aufbau einer Bezeichnung zu illustrieren, betrachten wir im Folgenden den Funktionsnamen  $\overset{\text{spigot}}{\underset{\text{fine}}{\text{APN}}}$ , der aus der Notation  $\overset{\text{spigot}}{\underset{\text{fine}}{\text{APN}}}(NE, g, i)$  in Definition 2.36 entnommen wird.

Mit der Bezeichnung  $\overset{\text{spigot}}{\underset{\text{fine}}{\text{APN}}}$  wird der Sachverhalt 'Spigot-Approximation von endlich geschachtelten Summen' bezeichnet. Die Bezeichnung kommt wie folgt zustande:

- Der Hauptteil APN deutet 'normierte Approximation' an, dabei ist die Kurzform APN aus dem Englischen 'normed approximation' für 'normierte Approximation' gebildet.

- Die Buchstabenfolge der Übersetzung  $\text{fne}$  ist aus der Abkürzung FNE abgeleitet und deutet an, dass ein Argument der Funktion eine endlich geschachtelte Summe (Finitely Nested Expression) ist.
- Die Übersetzung  $\text{spigot}$  deutet an, dass die Funktion einen Aspekt der Spigot-Berechnung widerspiegelt.

□

*Bemerkung 0.9* (Bezeichnung von Variablen). Zur Bezeichnung von Variablen sind zusätzlich zu Bemerkung 0.6 noch folgende Regeln zu beachten:

1. Der *Hauptteil erscheint immer in kursiver Schrift* (der Konvention für Variablennamen von L<sup>A</sup>T<sub>E</sub>X folgend) und soll möglichst den Typ andeuten.
2. Indizierungen dienen wahlweise zur Unterscheidung mehrerer Variablen eines Typs oder zur Modellierung von Arrays.

□

*Beispiel 0.10* (Variablen für geschachtelte Summe). NE ist die Abkürzung für geschachtelte Summe. Den Konventionen in Bemerkung 0.6 und Bemerkung 0.9 entsprechend werden  $NE$ ,  $NE_1$ ,  $NE^{\text{norm}}$  als Variablen für geschachtelte Summen verwendet. Die Übersetzung  $\text{norm}$  deutet den Kontext der Normierung an.

□

## 0.4 Gleichheit

Der Begriff der Gleichheit gehört zu den semantisch wichtigsten Begriffen in der Mathematik. Unter anderem basiert die Definition eines neuen Begriffes meist auf einer entsprechenden Stufe der Gleichheit. In diesem Abschnitt werden Bemerkungen über die Stufen und die Erscheinungsformen von Gleichheit in der vorliegenden Dissertation ausgeführt.

*Bemerkung 0.11* (Stufen der Gleichheit).

1. Die *Wertgleichheit* zwischen mathematischen Ausdrücken wird mit  $\overline{\text{val}}$  bezeichnet. Es gelten z. B. sowohl

$$1 + 1 \overline{\text{val}} 1 + 1$$

als auch

$$1 + 1 \overline{\text{val}} 2 .$$

2. Die *mathematische Ausdrucksgleichheit* wird mit  $\overline{\text{expr}}$  bezeichnet. Es gelten z.B

$$1 + 1 \overline{\text{expr}} 1 + 1$$

und

$$1 + 1 + 1 \overline{\text{expr}} (1 + 1) + 1$$

bei der linken Assoziativität der Addition, aber nicht

$$1 + 1 \overline{\text{expr}} 2 .$$

3. Die *Meta-Gleichheit*, d. h. die Gleichheit nicht nur zwischen mathematischen Ausdrücken, wird mit  $\equiv$  bezeichnet. Die Notation

$$[1, 2, 3] \equiv [1, 2, 3]$$

ist zulässig und bezeichnet die Gleichheit zwischen zwei Listen. Dagegen ist die Notation

$$[1, 2, 3] \stackrel{\text{expr}}{=} [1, 2, 3]$$

nicht zulässig, wenn vorher Listen nicht als mathematische Ausdrücke eingeführt worden sind.

Die Meta-Gleichheit impliziert die mathematische Ausdrucksgleichheit und diese wiederum impliziert die Wertgleichheit. Die Meta-Gleichheit entspricht etwa der Gleichheit nach dem bekannten Leibnizschen Prinzip: Zwei Dinge sind gleich, wenn sie *bezüglich aller ihrer Eigenschaften* gleich sind<sup>1</sup>. Bei den anderen Stufen der Gleichheit handelt es sich um eine entsprechende Fokussierung/Einschränkung der Betrachtungen auf bestimmte relevante Eigenschaften/Aspekte.  $\square$

*Bemerkung 0.12* (Erscheinungsformen der Gleichheit).

1. Wenn eine spezifische Stufe der Gleichheit hervorzuheben ist, wird die entsprechende Bezeichnung laut Bemerkung 0.11 eingesetzt.
2. Die Bedeutung von  $=$  ist kontextabhängig und im entsprechenden Kontext mit  $\stackrel{\text{val}}{=}$ ,  $\stackrel{\text{expr}}{=}$  oder  $\equiv$  gleichzusetzen.
3. Eine Definition kann mit  $\stackrel{\text{def}}{=}$  eingeleitet werden. Die Stufe der Gleichheit sowie die Richtung der Definition soll dabei aus dem Kontext erschließbar sein.

$\square$

## 0.5 Folgennotation

Folgen — sowohl endliche als auch unendliche — sind sowohl aus der Mathematik als auch aus der Informatik bekannt. In der Programmierung kommen Folgen häufig als Listen vor. In der vorliegenden Dissertation lassen sich viele Begriffe auf Folgen und viele Operationen auf die Manipulation von Folgen zurückführen. Eine Möglichkeit für die Notation von Folgen, die eine bequeme Notation von solchen Begriffen und Operationen anbieten soll, wird nun vereinbart.

*Bemerkung 0.13* (Folge mit expliziter Angabe der Indexgrenzen). Mit  $[E]_{i=n_1}^{n_2}$  wird eine Folge mit *Indexvariable*  $i$ , *Indexuntergrenze*  $n_1$  und *Indexobergrenze*  $n_2$  angesprochen, d. h. eine Folge der Form  $[x_{n_1}, x_{n_1+1}, \dots, x_{n_2}]$ . Im Folgenden werden die Regeln der Notation erläutert:

1. **Konsistenz der Indexgrenzen.** Es muss  $n_1 \leq n_2$  gelten.
2. **Fallunterscheidung zwischen konkreter Liste und allgemeinem Ausdruck.**  $E$  steht **entweder** für einen *allgemeinen Ausdruck* über die Variable  $i$  **oder** eine *konkrete Liste* von genau  $m_2 - n_1 + 1$  Elementen, die durch Komma getrennt sind.

<sup>1</sup>Vergl. [BB92, Kapitel III].

3. **Konsistenz und Semantik der Folgenverkettung.** Genau dann, wenn  $n_3 = n_2 + 1$ , kann aus

$$[E_1]_{i=n_1}^{n_2} \equiv [x_{n_1}, x_{n_1+1}, \dots, x_{n_2}]$$

und

$$[E_2]_{i=n_3}^{n_4} \equiv [x_{n_3}, x_{n_3+1}, \dots, x_{n_4}]$$

die Folge

$$[E_1]_{i=n_1}^{n_2} \overset{+}{seq} [E_2]_{i=n_3}^{n_4} \equiv [x_{n_1}, x_{n_1+1}, \dots, x_{n_4}]$$

gebildet werden.

□

*Beispiel 0.14* (Beispiele für gültige und ungültige Notationen von Folgen).

1. Die Notation  $[1]_{i=3}^2$  ist ungültig, weil die untere Indexgrenze größer als die obere Indexgrenze ist. Dagegen sind  $[1]_{i=3}^3$  und  $[1]_{i=4}^4$  gültig.
2. Mit  $[1, 2]_{i=3}^4$  ist eine Folge mit konkreter Elementenliste gegeben. Die Notation  $[1, 2]_{i=3}^3$  ist ungültig, weil die Angabe der unteren und oberen Indexgrenze nicht in Übereinstimmung mit der Anzahl der Elemente steht:  $3 - 3 + 1 \neq 2$ .
3. Die Notation  $[a_i, b_i]_{i=1}^2$  ist ungültig, weil es nicht zu bestimmen ist, ob eine konkrete Liste oder ein allgemeiner Ausdruck vorliegt (Komma auf der ersten syntaktischen Ebene als Trennzeichen für Elemente in der konkreten Liste und gleichzeitig das Vorhandensein der Indexvariable  $i$  in den Elementen). Mit  $[(a_i, b_i)]_{i=1}^2$  ist dagegen die Folge  $[(a_1, b_1), (a_2, b_2)]$  gemeint (Komma auf der zweiten syntaktischen Ebene im Bestandteil von Elementen).

□

*Bemerkung 0.15* (Folge mit ausschließlicher Angabe der Indexvariablen). Mit  $[E]_i$  wird eine Folge mit Indexvariable  $i$  angesprochen. Die Indexgrenzen sind für die betroffene Betrachtung nicht von Interesse. Der Ausdruck in eckigen Klammern  $E$  ist dabei als eine Funktion über  $i$  zu interpretieren.

□

## 0.6 Definieren, Spezifizieren und Herleiten

Der Inhalt der vorliegenden Dissertation wird im Wesentlichen durch Definieren, Spezifizieren und Herleiten aufgebaut. Im Folgenden werden allgemeine Bemerkungen über solche Vorgänge ausgeführt.

*Bemerkung 0.16* (Prädikatenlogik als Grundlage der Formulierung). Für die Formulierung von mathematischen Aussagen werden die Ausdrucksmittel der Prädikatenlogik benutzt. Beweise werden mit traditionellen Techniken des logischen Schließens der Prädikatenlogik geführt, die im Wesentlichen auf *Induktion* sowie *Unifikation* und *Termersetzung* beruhen. Im Rahmen der Studienarbeit [Do94] hat sich der Autor mit den Techniken des logischen Schließens der Prädikatenlogik höherer Ordnung intensiv auseinandergesetzt; der Inhalt von [Do94] baut dabei auf [Rei93] auf.

□

*Bemerkung 0.17* (Notation für Substitution und Verschiebung). Die Begriffe sind aus der Prädikatenlogik bekannt bzw. leicht definierbar. Hier wird eine einheitliche Art der Notationen vereinbart.

1. Eine Substitution ist mit einer Notation der Form  $E_1 \xrightarrow{\text{sub}} E_2$  versehen.  $E_1$  steht für das, was im originalen Ausdruck (eventuell auch mehrfach) vorkommt, und  $E_2$  bezeichnet den Ausdruck, durch den jedes Vorkommen von  $E_1$  zu ersetzen ist, um das Ergebnis der Substitution zu bilden. Für den einfachsten Fall ist  $E_1$  eine Variable im originalen Ausdruck. Wenn dies nicht zutrifft, muss der originale Ausdruck zuerst zu einem Ausdruck über  $E_1$  umgeschrieben werden. Die Semantik und die Anwendung dieser Notation kommen beispielsweise durch

$$E_1 \underset{E_1 \xrightarrow{\text{sub}} E_2}{=} E_2$$

deutlich zum Ausdruck.

2. Verschiebung ist ein Spezialfall von Substitution. Eine Verschiebung ist mit einer Notation der Form  $E_1 \xrightarrow{\text{shft}} K$  versehen, wobei  $E_1$  eine Variable und  $K$  eine Konstante bezeichnet.  $E_1 \xrightarrow{\text{shft}} K$  ist äquivalent zu  $E_1 \xrightarrow{\text{sub}} E_1 - K$ . Diese Notation ist vor allem für die Indexverschiebung gedacht. Dazu muss  $K \in \mathbb{Z}$  gelten. Die Semantik und die Anwendung dieser Notation kommen beispielsweise durch

$$\sum_{i=0}^n \underset{i \xrightarrow{\text{shft}} i_1}{=} \sum_{i=i_1}^{n+i_1}$$

deutlich zum Ausdruck.

□


*Bemerkung 0.18* (Ganzzahlige Rückwärtsinduktion). Die ganzzahlige Rückwärtsinduktion ist eine Variante der vollständigen Induktion, die in der vorliegenden Dissertation zur Spezifikation der Teilprozeduren des Spigot-Algorithmus häufig angewandt wird. Im Folgenden wird sie kurz charakterisiert.

1. **Die Definition durch ganzzahlige Rückwärtsinduktion.** Es seien  $k$  und  $m$  mit  $k \in \mathbb{Z}$ ,  $m \in \mathbb{Z}$ ,  $k < m$  gegeben. Wenn ein Begriff  $N(n)$  für  $n = m$  definiert und für alle  $i \mid k \leq i < m$  die Vorschrift der Definition von  $N(i)$  aus  $N(i+1)$  bekannt ist, so ist  $N(n)$  für alle  $n \mid k \leq n \leq m$  definiert.
2. **Der Beweis durch ganzzahlige Rückwärtsinduktion.** Es seien  $k$  und  $m$  mit  $k \in \mathbb{Z}$ ,  $m \in \mathbb{Z}$ ,  $k < m$  gegeben. Wenn eine Eigenschaft  $P(n)$  für  $n = m$  gilt und für alle  $i \mid k \leq i < m$  die Gültigkeit von  $P(i)$  aus der Gültigkeit von  $P(i+1)$  hergeleitet werden kann, so gilt  $P(n)$  für alle  $n \mid k \leq n \leq m$ .

□

*Bemerkung 0.19* (Spezifikation und Verifikation von Berechnungsvorschriften). Die Definition einer eigenen Logik für die Spezifikation von Berechnungsvorschriften, wie etwa Floyd-Hoare-Logik in [HS85], würde den Rahmen der vorliegenden Dissertation sprengen. Deshalb werden die Berechnungsvorschriften in Pseudo-Form spezifiziert. Mathematische Exaktheit und Strenge werden jedoch dadurch erreicht, dass die Datenwerte eines Algorithmus durch ein System von mathematischen Begriffen modelliert werden, deren Eigenschaften mit mathematischer Strenge hergeleitet sind. Ein Algorithmus wird dadurch verifiziert, dass der Zusammenhang zwischen

dessen Datenwerten und den zu modellierenden mathematischen Begriffen aufgezeigt wird. Das Entscheidende für die Verifikation ist dabei die Zuordnung einer grundlegenden Invarianz zu einer mathematischen Gleichung. Man siehe Bemerkung 2.27 für detaillierte Konventionen und Kommentar 2.28 für ein Beispiel der Verifikation.  $\square$

*Bemerkung 0.20* (Darstellung des logischen Schließens mit Herleitungen). Das logische Schließen wird in der vorliegenden Dissertation häufig anhand von *Herleitungen* durchgeführt. Eine Herleitung ist eine abbildungsartige Zusammenstellung von Gleichungen und Formeln, die in einer semantischen Beziehung zueinander stehen. In der Regel ist eine Herleitung eine Kette von Ausdrücken, die in ein und derselben transitiven Relation (Gleichung oder Ungleichung) stehen. Dabei wird die Markierung  (gegebenenfalls mit Beschriftung und Verweis versehen) dazu benutzt, signifikante Stellen eines Ausdrucks zu kennzeichnen, insbesondere den Teilausdruck, der für den aktuellen Transformationsschritt relevant ist, um den nächsten Ausdruck zu bilden. Alle Herleitungen in der vorliegenden Dissertation werden im Verzeichnis der Herleitungen aufgelistet. Die Darstellung des logischen Schließens mit Herleitungen ist eine Besonderheit im Stil der Beweisführung der vorliegenden Dissertation in der Hoffnung, Beweise kurz, übersichtlich, verständlich und prägnant darzustellen.  $\square$

Teil I

**Der Spigot-Algorithmus als der Kern  
des Spigot-Ansatzes**

# Kapitel 1

## Einleitung

In diesem Kapitel werden anhand von Beispielen aus der Mathematikgeschichte die charakteristischen Aspekte der Spigot-Berechnung sowie offene Probleme diskutiert. Dadurch wird die Motivation für eine Weiterentwicklung offengelegt. Die inhaltliche Grundlage für die Formalisierung im nächsten Kapitel wird bereitgestellt.

Die Konstanten  $e$  und  $\pi$  haben eine große Bedeutung für die gesamte Mathematik, und wir werden in diesem Kapitel sehen, dass sie auch bei der Entstehung der Idee des Spigot-Algorithmus die entscheidenden Schritte ausmachen.

### 1.1 Die Zahl $e$ und die Entdeckung des Spigot-Algorithmus

Die Idee der Spigot-Berechnung wurde bei der Berechnung von  $e$  mit Hilfe der Gleichung

$$e = \sum_{k=0}^{\infty} \frac{1}{k!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots \quad (1.1)$$

entdeckt. Zur Motivation der Idee der Spigot-Berechnung wird zunächst eine naive Methode für die Berechnung von  $e$  betrachtet, die ohne tiefgreifende Überlegungen direkt aus (1.1) abgeleitet ist.

#### 1.1.1 Die naive Methode und ihre Mängel

Bei der naiven Berechnung von  $e$  werden die Terme der Reihen in (1.1) sukzessiv von links nach rechts bis zu einem Genauigkeitsmaß berechnet und addiert. Unter Beachtung der Eigenschaft, dass der  $(i - 1)$ -te Term das  $i$ -fache des  $i$ -ten Term ist, lässt sich die Berechnung in Tabelle 1.1 skizzieren, wobei mit  $t[i]$  und  $s[i]$  die Werte des  $i$ -ten Terms bzw. der  $i$ -ten Teilsumme von (1.1) bezeichnet werden.

Die naive Berechnung weist folgende Mängel auf:

- **Begrenzung der Genauigkeit durch das benutzte Zahlensystem.** Um beispielsweise ein Genauigkeitsmaß von  $10^{-100}$  zu erreichen, muss der Datentyp, in dem  $s$  und  $t$  deklariert sind, imstande sein, solche Werte aufzunehmen. Ein solcher Datentyp ist in den meisten Programmiersprachen nicht vorhanden und muss bei Bedarf mit großem Aufwand implementiert werden.



Schritt	Aktion
0	$t[0] := 1;$ $s[0] := 1;$
1	$t[1] := \frac{t[0]}{1};$ $s[1] := s[0] + t[1];$
2	$t[2] := \frac{t[1]}{2};$ $s[2] := s[1] + t[2];$
3	$t[3] := \frac{t[2]}{3};$ $s[3] := s[2] + t[3];$
4	$t[4] := \frac{t[3]}{4};$ $s[4] := s[3] + t[4];$
5	$t[5] := \frac{t[4]}{5};$ $s[5] := s[4] + t[5];$
$\vdots$	$\dots$ $\dots$
n	$t[n] := \frac{t[n-1]}{n};$ $s[n] := s[n-1] + t[n];$

Die Approximation  $s[n] \approx e$  wird ausgegeben, wenn  $t[n] < \epsilon$  gilt.

Tabelle 1.1: Naive Berechnung von  $e$  mit dem Genauigkeitsmaß  $\epsilon$  auf der Grundlage von (1.1)

- **Informationsverlust durch Rundung und Fehlerfortpflanzung.** Auch wenn ein Datentyp mit erforderlicher Genauigkeit vorhanden ist, liefert die Division in den meisten Fällen keine mathematisch exakten Ergebnisse. Wenn beispielsweise  $s$  und  $t$  in einem dezimalen oder hexadezimalen System deklariert sind, gibt es schon bei der Berechnung von  $t[3]$  Informationsverlust durch Rundung. Dieser Informationsverlust beeinträchtigt die Berechnungen aller  $t[i]$  mit  $i > 3$ . *Dadurch sind mathematisch exakte Aussagen über den absoluten Fehler der Berechnung schwer oder gar unmöglich herzuleiten.*

### 1.1.2 Die Idee der Spigot-Berechnung

Bei dem Versuch, die Mängel der naiven Berechnung von  $e$  zu meistern, hat Sale die Idee der Spigot-Berechnung entdeckt und in [Sal68] veröffentlicht. In diesem Abschnitt wird die Idee von Sale in eigener Ausdrucksform des Autors mit zusätzlichen Bemerkungen detailliert wiedergegeben, um eine solide inhaltliche Grundlage für die Formalisierung im nächsten Kapitel zu erhalten.

Zur Berechnung von  $e$  ist zuerst eine ausreichende Approximationsumme

$$e_n = \sum_{k=0}^n \frac{1}{k!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!} \quad (1.2)$$

auszuwählen; dann werden aus dieser Summe die Wertanteile in dezimalen Portionen extrahiert.

Aus algorithmischer Sicht können die dezimalen Wertanteile jeweils mit ihren zugehörigen Koeffizienten versehen werden, und wenn alle diese Koeffizienten kleiner als 10 sind, formen sie tatsächlich die Ziffern einer Zahl in dezimaler Darstellung. In diesem Fall kann von einem Prozess der Extraktion der Dezimalziffern geredet werden.

Der *Spigot-Algorithmus* beinhaltet im Wesentlichen die Extraktion der Wertanteile aus einer endlichen Summe und wird im Folgenden näher betrachtet.

*Kommentar 1.1* (Die Arbeitsweise des Spigot-Algorithmus am Beispiel der Berechnung der Dezimalziffern von  $e_7$ ).

- Zuerst wird  $e_7$  nach dem Distributionsgesetz wie in Herleitung 1.1 in eine *geschachtelte*

$$\begin{aligned}
e_7 &= \underbrace{\frac{1}{0!} + \frac{1}{1!}} + \underbrace{\frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!}} \\
&= 2 + \underbrace{\frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \frac{1}{6!} + \frac{1}{7!}} \\
&= 2 + \frac{1}{2} \left( 1 + \frac{1}{3} + \frac{1}{3 \cdot 4} + \frac{1}{3 \cdot 4 \cdot 5} + \frac{1}{3 \cdot 4 \cdot 5 \cdot 6} + \frac{1}{3 \cdot 4 \cdot 5 \cdot 6 \cdot 7} \right) \\
&= 2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} + \frac{1}{4 \cdot 5} + \frac{1}{4 \cdot 5 \cdot 6} + \frac{1}{4 \cdot 5 \cdot 6 \cdot 7} \right) \right) \\
&= 2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} + \frac{1}{5 \cdot 6} + \frac{1}{5 \cdot 6 \cdot 7} \right) \right) \right) \\
&= 2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} \left( 1 + \frac{1}{6} + \frac{1}{6 \cdot 7} \right) \right) \right) \right) \\
&= 2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 1 + \frac{1}{7} \right) \right) \right) \right) \right).
\end{aligned}$$

Herleitung 1.1: Überführung von  $e_7$  in eine geschachtelte Summe

*Summe* umgeformt. Als Ergebnis entsteht

$$2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 1 + \frac{1}{7} \right) \right) \right) \right) \right). \quad (1.3)$$

Diese Umformung dient nur zum Verständnis der Arbeitsweise des Spigot-Algorithmus, für sie ist keine numerische Rechnung erforderlich. Die ganzen Zahlen vor dem Zeichen „+“ werden *Koeffizienten*, die gebrochenen Zahlen vor der öffnenden Klammer „(“ werden *Faktoren* der geschachtelten Summe genannt.

- Der *freie Koeffizient* 2 von (1.3) wird als der ganze Teil von  $e$  angenommen.
- Mit dem *Restausdruck*

$$\frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 1 + \frac{1}{7} \right) \right) \right) \right) \right) \quad (1.4)$$

von (1.3) kann nun der Prozess der „Extraktion der Wertanteile in dezimalen Portionen aus dem Restausdruck“ wiederholt angewandt werden. Eine Runde der Extraktion bekommt als Input einen Restausdruck und erzeugt neben dem entsprechenden Koeffizienten wieder einen Restausdruck, der als Input für die nächste Runde benutzt werden kann.

Die *erste Runde der Extraktion*, in der die erste dezimale Nachkommaziffer von  $e_7$  aus (1.4) extrahiert wird, ist in Herleitung 1.2 anschaulich dargestellt. Diese Runde wird im Folgenden näher betrachtet.

Alle Koeffizienten von (1.4) werden mit 10 multipliziert, so dass der Term

$$\frac{1}{2} \left( 10 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 10 + \frac{1}{5} \left( 10 + \frac{1}{6} \left( 10 + \frac{10}{7} \right) \right) \right) \right) \right) \quad (1.5)$$

$$\begin{aligned}
e_7 &= \underbrace{2}_{\text{freier Koeffizient}} + \underbrace{\frac{1}{2}\left(1 + \frac{1}{3}\left(1 + \frac{1}{4}\left(1 + \frac{1}{5}\left(1 + \frac{1}{6}\left(1 + \frac{1}{7}\right)\right)\right)\right)\right)}_{\text{Restausdruck (1.4) als Ausgang der Berechnung}} \\
&= 2 + \frac{1}{10} \underbrace{\left[10 \cdot \left(\frac{1}{2}\left(1 + \frac{1}{3}\left(1 + \frac{1}{4}\left(1 + \frac{1}{5}\left(1 + \frac{1}{6}\left(1 + \frac{1}{7}\right)\right)\right)\right)\right)\right)}_{(1.5)} \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 10 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 10 + \frac{1}{5} \left( 10 + \frac{1}{6} \left( 10 + \frac{10}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 10 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 10 + \frac{1}{5} \left( 10 + \frac{1}{6} \left( 11 + \frac{3}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 10 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 10 + \frac{1}{5} \left( 11 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 10 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 12 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 10 + \frac{1}{3} \left( 13 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \frac{1}{2} \left( 14 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right] \\
&= 2 + \frac{1}{10} \left[ \underbrace{7}_{\text{extrahiert}} + \underbrace{\frac{1}{2} \left( 0 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right)}_{\text{Restausdruck (1.7) für nächste Runde in Tabelle 1.3}} \right] \\
&= \underbrace{2.7}_{\text{Resultat}} + \frac{1}{10} \cdot \left[ \frac{1}{2} \left( 0 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right]
\end{aligned}$$

Umformungen sind laut Regel (1.6) auszuführen.

Herleitung 1.2: Die erste Runde der Extraktion dezimaler Wertanteile aus  $e_7$

entsteht, dann wird (1.5) laut der Regel

$$\underbrace{a + \frac{1}{b}(c + \langle \square \rangle)} \quad \text{zu} \quad \left( a + (c \div b) \right) + \frac{1}{b} \left( (c \bmod b) + \langle \square \rangle \right), \quad (1.6)$$

wobei

1.  $\langle \square \rangle$  für den unveränderten Teil steht, der bei einer konkreten Anwendung der Regel mit einem Teilausdruck zu füllen ist,

2. die Ausdrücke  $(a + (c \div b))$  und  $(c \bmod b)$  numerisch auszuwerten sind

und

3. die Markierung  $\underbrace{\hspace{2cm}}$  nur eine visuelle Hilfestellung leistet, um die zu ändernden Stellen des Inputs kenntlich zu machen,

sukzessiv von rechts nach links modifiziert. Die Wertgleichheit zwischen dem Input und dem Output der Regel (1.6) ist durch die elementaren Rechenregeln der Arithmetik leicht nachvollziehbar. Demnach bleibt der gesamte Wert von (1.5) unverändert, während die

$$\begin{aligned}
e_7 &= \underbrace{2.7}_{\text{bisherige Resultat}} + \frac{1}{10} \cdot \left[ \underbrace{\frac{1}{2} \left( 0 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right)}_{\text{Restausdruck (1.7) aus Tabelle 1.2 als Ausgang der Berechnung}} \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ 10 \cdot \underbrace{\left( \frac{1}{2} \left( 0 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \right)}_{\text{Restausdruck (1.7) aus Tabelle 1.2 als Ausgang der Berechnung}} \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 0 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 10 + \frac{1}{6} \left( 50 + \frac{30}{7} \right) \right) \right) \right) \right) \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 0 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 10 + \frac{1}{6} \left( 54 + \frac{2}{7} \right) \right) \right) \right) \right) \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 0 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 19 + \frac{1}{6} \left( 0 + \frac{2}{7} \right) \right) \right) \right) \right) \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 0 + \frac{1}{3} \left( 10 + \frac{1}{4} \left( 3 + \frac{1}{5} \left( 4 + \frac{1}{6} \left( 0 + \frac{2}{7} \right) \right) \right) \right) \right) \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 3 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 3 + \frac{1}{5} \left( 4 + \frac{1}{6} \left( 0 + \frac{2}{7} \right) \right) \right) \right) \right) \right] \\
&= 2.7 + \frac{1}{100} \cdot \left[ \underbrace{1}_{\text{extrahiert}} + \underbrace{\frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 3 + \frac{1}{5} \left( 4 + \frac{1}{6} \left( 0 + \frac{2}{7} \right) \right) \right) \right) \right)}_{\text{Restausdruck für nächste Runde}} \right] \\
&= \underbrace{2.71}_{\text{Resultat}} + \frac{1}{100} \cdot \left[ \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 3 + \frac{1}{5} \left( 4 + \frac{1}{6} \left( 0 + \frac{2}{7} \right) \right) \right) \right) \right) \right]
\end{aligned}$$

Umformungen sind laut Regel (1.6) auszuführen.

Herleitung 1.3: Die zweite Runde der Extraktion dezimaler Wertanteile aus  $e_7$

Wertanteile infolge der Modifizierung von rechts nach links durchwandern und sich schließlich auf dem am weitesten links stehenden Summanden — *dem freien Koeffizienten* — konzentrieren, der dann mit der ersten dezimalen Nachkommaziffer von  $e_7$  zu identifizieren ist und extrahiert wird.

Der Restausdruck nach der Extraktion des freien Koeffizienten am Ende einer Runde kann als Input für eine weitere Runde benutzt werden, so dient der Restausdruck

$$\frac{1}{2} \left( 0 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 0 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 5 + \frac{3}{7} \right) \right) \right) \right) \right) \quad (1.7)$$

der ersten Runde als Input für die zweite Runde der Extraktion, die dann ähnlich wie die erste Runde verläuft und in Herleitung 1.3 dargestellt ist. Die Ausführung weiterer Runden bis zu einer beliebigen Anzahl geschieht in der gleichen Art und Weise. Die Extraktion der Dezimalziffern aus  $e_7$  bis zu der Anzahl `len` ist in Algorithmus 1.1 als Pseudocode dargestellt.  $\square$

Nun wenden wir uns der Geschichte der Spigot-Berechnung zu. Die Grundidee wurde von Sale entdeckt und in [Sal68] verbal-intuitiv formuliert, von der ausgehend ein ALGOL 60 Programm für die Extraktion der Dezimalziffern von  $e$  entwickelt wurde. Dabei wurde angenommen, dass die extrahierten Koeffizienten die Zahl 9 nicht übersteigen und folglich Dezimalziffern sind. Diese Tatsache wird erst durch Rabinowitz und Wagon in [RW95] bewiesen.

---

**Algorithmus 1.1** Extraktion der Dezimalziffern von  $e_7$ 


---

```

1: DECLARE AS: extractDecDigFromE7 (len, Out Result)
2: Reg[0] := 2
3: for pos = 1 to 6 step 1 do
4:   Reg[pos] := 1
5: end for
6: Result[0] := Reg[0]
7: for ite = 1 to len step 1 do
8:   carry := 0
9:   for pos = 6 to 1 step -1 do
10:    help := 10 × Reg[pos] + carry
11:    Reg[pos] := help mod (pos + 1)
12:    carry := help ÷ (pos + 1)
13:   end for
14:   Result[ite] := carry
15: end for

```

---

Abdali gab in [Abd70] den Spigot-Algorithmus in Pseudo-Form sowie einen Bericht über eine Beispielrechnung von  $\sin 0.6$  an. Der Pseudocode ist kurz und exakt. Die gesamte Darstellung ist aber zu knapp, um die Arbeitsweise des Algorithmus nachvollziehen zu können. Die Aussage über die Genauigkeit der Beispielrechnung war ohne Beweis ausgeführt.

Rabinowitz und Wagong haben in [RW95] die Spigot-Berechnung von  $e$  überarbeitet und darauf aufbauend eine Spigot-Berechnung von  $\pi$  hergeleitet. Die für den Spigot-Algorithmus relevanten Eigenschaften eines Faktorsystems (in der originalen Arbeit *base* genannt) werden dabei zum ersten Mal untersucht.

In [Sto97] findet man die Notation des Spigot-Algorithmus in Form einer Turing-Maschine sowie Hinweise zur Herleitung von Spigot-Berechnungen für mehrere mathematische Konstanten, darunter  $\sqrt{2}$ ,  $\sqrt[3]{2}$  und  $\ln 2$ .

Der Spigot-Algorithmus stellt — wie in [RW95] bemerkt ist — eine verallgemeinerte Form der Basis-Konvertierung (vergl. [Knu97, Abschnitt 4.4.A]) dar. Dementsprechend beinhaltet Kommentar 1.1 die Konvertierung von  $e_7$  aus der inhomogenen Basis  $(\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7})$  in die homogene Basis  $(\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \dots)$ .

## 1.2 Besonderheit der Spigot-Berechnung von $\pi$

Die Spigot-Berechnung von  $\pi$  in [RW95] geht von der Gleichung

$$\pi = 2 \sum_{k=0}^{\infty} \frac{k!}{\prod_{j=0}^k (2j+1)} = 2 + 2 \frac{1}{3} + 2 \frac{1 \cdot 2}{3 \cdot 5} + 2 \frac{1 \cdot 2 \cdot 3}{3 \cdot 5 \cdot 7} + \dots \stackrel{\text{def}}{=} \pi^{\text{GLE}} \quad 1 \quad (1.8)$$

aus.

Analog zur Berechnung von  $e$  wird zuerst eine Teilsumme  $\pi_n^{\text{GLE}}$  von (1.8) ausgewählt, dann wird  $\pi_n^{\text{GLE}}$  in eine geschachtelte Summe überführt, und anschließend werden aus der geschachtelten Summe dezimale Wertanteile sukzessiv extrahiert. Wegen der Eigenschaften von  $\pi^{\text{GLE}}$  sind folgende Probleme zu berücksichtigen:

---

<sup>1</sup> Die Bezeichnung oben rechts  $\text{GLE}$  von  $\pi^{\text{GLE}}$  weist darauf hin, dass die Reihe aus der Gregory-Leibnitz-Reihe durch Anwendung der Euler-Transformation hergeleitet ist.

$$\begin{aligned}
\pi_9^{\text{GLE}} &= 2 + \underbrace{\frac{1}{3}(2 + \frac{2}{5}(2 + \frac{3}{7}(2 + \frac{4}{9}(2 + \frac{5}{11}(2 + \frac{6}{13}(2 + \frac{7}{15}(2 + \frac{8}{17}(2 + \frac{9}{19}(2))))))))))}_{\text{Restausdruck als Ausgang der Berechnung}} \\
&= 2 + \frac{1}{10} \underbrace{[10 \cdot \frac{1}{3}(2 + \frac{2}{5}(2 + \frac{3}{7}(2 + \frac{4}{9}(20 + \frac{5}{11}(2 + \frac{6}{13}(2 + \frac{7}{15}(2 + \frac{8}{17}(2 + \frac{9}{19}(2))))))))))]_{\text{Restausdruck als Ausgang der Berechnung}} \\
&= 2 + \frac{1}{10} \\
&\quad \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(20 + \frac{5}{11}(20 + \frac{6}{13}(20 + \frac{7}{15}(20 + \frac{8}{17}(20 + \frac{9}{19}(20)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(20 + \frac{5}{11}(20 + \frac{6}{13}(20 + \frac{7}{15}(20 + \frac{8}{17}(29 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(20 + \frac{5}{11}(20 + \frac{6}{13}(20 + \frac{7}{15}(28 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(20 + \frac{5}{11}(20 + \frac{6}{13}(27 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(20 + \frac{5}{11}(32 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(20 + \frac{4}{9}(30 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(20 + \frac{3}{7}(32 + \frac{4}{9}(3 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(20 + \frac{2}{5}(32 + \frac{3}{7}(4 + \frac{4}{9}(3 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot \frac{1}{3}(32 + \frac{2}{5}(2 + \frac{3}{7}(4 + \frac{4}{9}(3 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&= 2 + \frac{1}{10} \cdot ( \underbrace{10}_{\text{Korrektur!}} \\
&\quad + \frac{1}{3}(2 + \frac{2}{5}(2 + \frac{3}{7}(4 + \frac{4}{9}(3 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1)))))))))) \\
&\quad \underbrace{\hspace{10em}}_{\text{Restausdruck für nächsten Prozess}} \\
&= 3 + \frac{1}{10} \cdot (\frac{1}{3}(2 + \frac{2}{5}(2 + \frac{3}{7}(4 + \frac{4}{9}(3 + \frac{5}{11}(10 + \frac{6}{13}(1 + \frac{7}{15}(13 + \frac{8}{17}(12 + \frac{9}{19}(1))))))))))
\end{aligned}$$

Umformungen sind laut Regel (1.9) auszuführen.

Herleitung 1.4: Die erste Runde der Extraktion dezimaler Wertanteile aus  $\pi_9^{\text{GLE}}$

---

**Algorithmus 1.2** Extraktion der dezimalen Wertanteile von  $\pi_9^{\text{GLE}}$ 


---

```

1: DECLARE AS: extractDecValFromPiGLE9 (num, Out Result)
2: for pos = 0 to 9 step 1 do
3:   Reg[pos] := 2
4: end for
5: Result[0] := Reg[0]
6: for ite = 1 to num step 1 do
7:   carry := 0
8:   for pos = 9 to 1 step -1 do
9:     help := 10 × Reg[pos] + carry
10:    Reg[pos] := help mod (2 × pos + 1)
11:    carry := pos × (help ÷ (2 × pos + 1))
12:   end for
13:   Result[ite] := carry
14: end for

```

---

1. **Neue Regel für Wertverschiebung.** Die Faktoren der geschachtelten Summe haben in diesem Fall eine allgemeinere Form  $\frac{a}{b}$  mit zwei Parametern  $a$  und  $b$ , demzufolge ist die Regel für die „Verschiebung der Wertanteile von rechts nach links“ zu der folgenden Form zu modifizieren:

$$\underbrace{c + \frac{a}{b}(d + \langle \square \rangle)} \quad \text{zu} \quad (c + a \times (d \div b)) + \frac{a}{b}((d \bmod b) + \langle \square \rangle) \quad (1.9)$$

Dieses Problem ist mit einer Modifizierung der Berechnungsvorschrift einfach zu lösen: Man vergleiche dazu die Anweisung 12 in Algorithmus 1.1 mit der Anweisung 11 in Algorithmus 1.2.

2. **Stellenwertüberlauf.** Extrahierte Koeffizienten können die Basis 10 übersteigen, wodurch Nacharbeitung/Korrektur notwendig ist.

Das Problem des Stellenwertüberlaufs ist schon bei der ersten Runde der Extraktion der dezimalen Wertanteile aus  $\pi_9^{\text{GLE}}$  zu erkennen, die in Herleitung 1.4 dargestellt ist<sup>2</sup>. Die Überführung von  $\pi_9^{\text{GLE}}$  in die als Ausgangspunkt dienende geschachtelte Summe wird dabei ausgelassen; sie verläuft analog zu Herleitung 1.1.

In [RW95] wird zur Korrektur das Konzept der dort genannten „*predigit*“ eingeführt. Dieses Konzept ist aber speziell auf  $\pi$  zugeschnitten und wird deshalb in der vorliegenden Dissertation nicht weiter betrachtet. Eine einfache und allgemeine Lösung wird dadurch erreicht, dass die extrahierten Koeffizienten erst am Ende durch Ergebnismodulierung<sup>3</sup> korrigiert werden.

### 1.3 Motivation zur Weiterentwicklung

*Bemerkung 1.2* (Positive Merkmale von Spigot-Berechnungen im Hinblick auf Zuverlässigkeit). Aus der in Kommentar 1.1 dargelegten Arbeitsweise des Spigot-Algorithmus lassen sich folgende positive Merkmale von Spigot-Berechnungen im Hinblick auf Zuverlässigkeit erkennen:

---

<sup>2</sup> Herleitung 1.4 ist dabei ähnlich wie Herleitung 1.2 und Herleitung 1.3 zu interpretieren.

<sup>3</sup> Vergl. Definition 2.21, Korollar 2.32, Definition 2.36 und Algorithmus 2.3

1. **Maschinenunabhängige Berechnung ohne Informationsverlust und Fehlerfortpflanzung.** Die notwendigen elementaren Operationen sind laut Herleitung 1.2 nur Addition, ganzzahlige Division und ganzzahlige Restbildung, die von jedem Computer mathematisch exakt ohne Rundung ausführbar sind.
2. **Anscheinend einfache und sichere Kontrolle der Genauigkeit bis zu einem beliebigen Maß.** Bei der Berechnung einer endlichen Summe wird eine beliebig gewünschte Genauigkeit durch eine ausreichende Anzahl Iterationen der Extraktion bewerkstelligt. Die Berechnung einer unendlichen Summe ist auf die Berechnung einer endlichen Summe zurückgeführt.
3. **Berechnung mit kleinen ganzen Zahlen für hohe Genauigkeit.** Für die Realisierung eines Genauigkeitsmaßes von  $\frac{1}{n!}$  sind z. B. im Fall der Berechnung von  $e$  Operationen nur mit ganzen Zahlen notwendig, die scheinbar nicht wesentlich größer als  $n$  sind.
4. **Potenziell mathematisch exakt begründete Zuverlässigkeit.** Als Ergebnis der oben genannten Punkte ist die Möglichkeit höchstwahrscheinlich, die Genauigkeit einer Berechnung mathematisch exakt zu begründen. Nach der Auffassung des Autors stellt diese Art und Weise *das Höchstmaß* an Zuverlässigkeit dar.

□

*Bemerkung 1.3* (Einschränkung des effektiven und korrekten Einsatzes von Spigot-Berechnungen durch den Mangel an formalen Grundlagen).

1. Es fehlen formale Grundlagen, mit deren Hilfe effektiv entschieden werden kann, ob und wie gut der Spigot-Algorithmus für eine gegebene Reihe anzuwenden ist. Das Problem des Stellenwertüberlaufs bei der Berechnung von  $\pi$  in [RW95] (vergl. Abschnitt 1.2), das auch in [Sto97, Kapitel 3, Seiten 66, 91, 92] bei der Berechnung von  $\sqrt{2}$  erkannt worden ist, ist z. B. ein Zeichen dafür, dass der auf Intuition basierende Algorithmus für  $e$  nicht ohne weiteres für  $\pi$  und  $\sqrt{2}$  anzuwenden ist. Mit dem Algorithmus in [RW95] wurde das Problema des Stellenwertüberlaufs nur speziell für die Berechnung von  $\pi$  gelöst.
2. Obwohl bei Spigot-Berechnungen die Kontrolle der Genauigkeit wie erwähnt einfach erscheint, fehlen formale Grundlagen, mit deren Hilfe die Genauigkeit einer Berechnung mathematisch exakt begründet werden kann. Nur in [RW95] gibt es einen mathematischen Beweis für die Genauigkeit der Berechnung von  $\pi$ . Der Beweis ist jedoch unvollständig, weil ein wesentlicher Aspekt der Spigot-Berechnung, nämlich der Konvertierungsfehler<sup>4</sup>, nicht berücksichtigt wird.

□

Durch Bemerkung 1.2 und Bemerkung 1.3 ist die Weiterentwicklung der Spigot-Vorgehensweise zu einem möglichst allgemeinen Ansatz für numerische Berechnungen mit mathematisch exakt begründeter Zuverlässigkeit von starkem Interesse. Diese Aufgabe übernimmt die vorliegende Dissertation. In Bemerkung 1.4 wird die Weiterentwicklung im Zusammenhang mit den wichtigsten Begriffen der Spigot-Vorgehensweise näher erklärt.

*Bemerkung 1.4* (Spigot-Algorithmus, Spigot-Berechnung, Spigot-Ansatz und die Weiterentwicklung in der vorliegenden Dissertation). Die drei Begriffe, die in der vorliegenden Dissertation immer wieder vorkommen und den roten Faden bilden, werden im Folgenden näher erklärt:

---

<sup>4</sup>Siehe Definition 2.54



1. Der *Spigot-Algorithmus* ist der Algorithmus, der in verschiedenen Varianten und Notationsformen in [Sal68], [Abd70], [RW95] und [Sto97] vorkommt. In der vorliegenden Dissertation ist er genauer zu spezifizieren, zu vervollkommen und zu formalisieren.
2. *Spigot-Berechnungen* sind numerische Berechnungen, die im Wesentlichen die numerische Berechnung einer Reihe mit dem Spigot-Algorithmus beinhalten.
3. Der *Spigot-Ansatz* versteht sich als die gesamte Vorgehensweise für den effektiven und korrekten Einsatz des Spigot-Algorithmus. In der vorliegenden Dissertation ist er durch das Schaffen der formalen Grundlagen für den Spigot-Algorithmus und die Systematisierung der Techniken der Reihenentwicklung und Reihenbeschleunigung<sup>5</sup> bereitzustellen.

Zum Inhalt der vorliegenden Dissertation gehört auch die Anwendung des bereitgestellten Spigot-Ansatzes auf die speziellen Aufgaben der Wurzel- und Logarithmenberechnung.  $\square$

Im Sinn von Bemerkung 1.4 ist der Spigot-Algorithmus der Kern des Spigot-Ansatzes und wird in ihn eingebettet.

---

<sup>5</sup>Reihentransformationen zur Verbesserung der Konvergenz

# Kapitel 2

## Formale Grundlagen für den Einsatz des Spigot-Algorithmus

In diesem Kapitel werden die formalen Grundlagen für den effektiven und korrekten Einsatz des im vorigen Kapitel intuitiv vorgestellten Spigot-Algorithmus geschaffen. Formale Werkzeuge für die mathematisch exakte Herleitung von Aussagen über die Genauigkeit der Berechnung sowie über die Obergrenze der größten benötigten Zahl werden entwickelt, deren Anwendung dann durch ein komplexes numerisches Beispiel der Berechnung von  $\sqrt{2}$  nach Stoschek illustriert wird. Ein wesentlicher Teil des Inhalts ist aus [Do01, Abschnitt 2] entnommen, wird aber in einer grundlegend neuen Weise dargestellt, dies betrifft insbesondere die Symbolik für spigot-spezifische Funktionen und Operationen, die nun systematisch gemäß den Konventionen im Abschnitt 0.3 aufgebaut ist.

### 2.1 FNEs und der Spigot-Algorithmus

Der grundlegende mathematische Ausdruck für den Spigot-Algorithmus ist die FNE<sup>1</sup>, die wir im vorigen Kapitel bei der Berechnung von  $e$  durch (1.3) intuitiv kennengelernt haben. Im direkten syntaktisch-strukturellen Zusammenhang dazu stehen die Begriffe Faktorsystem und Restausdruck. Der Spigot-Algorithmus besteht aus Berechnungsvorschriften zur Manipulation von FNEs. Für die Formalisierung des Spigot-Algorithmus werden zuerst die Struktur und die Interpretation von FNEs charakterisiert, auf dieser Grundlage wird dann ein System von aufeinander aufbauenden Operationen und Manipulationsalgorithmen definiert, dessen Ziel der Spigot-Algorithmus ist. Die wichtigsten Beziehungen der Spigot-Berechnung lassen sich durch (2.61) und (2.33) formalisieren, wobei (2.61) den Bezugspunkt für die Bewerkstelligung einer gewünschten Genauigkeit der Berechnung und (2.33) den Bezugspunkt für die Abschätzung der

---

<sup>1</sup> Abkürzungen in Bezug auf endlich geschachtelte Summe:

- NE steht für geschachtelte Summe (engl.: NE = Nested Expression).
- FS steht für Faktorsystem (engl.: FS = Factor System).
- FNE steht für endlich geschachtelte Summe (engl.: FNE = Finitely Nested Expression).
- FFS steht für endliches Faktorsystem (engl.: FFS = Finite Factor System).
- RNE steht für geschachtelter Restausdruck (engl.: RNE = Remain Nested Expression).

Diese Liste soll gemäß Bemerkung 0.9 und Bemerkung 0.7 dazu dienen, Bezeichnungen intuitiv anzudeuten, und dadurch die Orientierung und das Denken zu erleichtern. Im Text werden die Abkürzungen gemäß Bemerkung 0.1 behandelt.

größten benötigten Zahl beinhalten. Auf (2.61) und (2.33) lassen sich alle Anwendungsformen in Abschnitt 2.2.2 zurückführen.

### 2.1.1 Struktur, Notation, Manipulation und Wert-Eigenschaften von FNEs

#### 2.1.2 FNEs

**Definition 2.1** (Faktorsystem, Koeffizientenfolge, Konstruktionskette, FNE). Es seien ein **Anfangsindex**  $k \in \mathbb{Z}$ , ein **Grad**  $n \in \mathbb{Z}^+$ , ein **Faktorsystem**

$$\left[ \frac{a_i}{b_i} \right]_{i=k+1}^{k+n} \quad | \quad a_i, b_i \in \mathbb{Z}^+ \quad (2.1)$$

und eine **Koeffizientenfolge**

$$[c_i]_{i=k}^{k+n} \quad | \quad c_i \in \mathbb{Z} \quad (2.2)$$

gegeben. Aus  $\left[ \frac{a_i}{b_i} \right]_{i=k+1}^{k+n}$  und  $[c_i]_{i=k}^{k+n}$  wird eine absteigend indizierte Folge von mathematischen Ausdrücken  $\left[ \overset{n}{NE}_n, \overset{n}{NE}_{n-1}, \dots, \overset{n}{NE}_0 \right]$  induktiv im Sinn von Bemerkung 0.18 wie folgt konstruiert:

1. Induktionsanfang

$$\overset{n}{NE}_n \stackrel{\text{expr}}{=} c_{k+n} . \quad (2.3)$$

2. Induktionsschritt

$$\overset{n}{NE}_{i-1} \stackrel{\text{expr}}{=} c_{k+i-1} + \frac{a_{k+i}}{b_{k+i}} \overset{n}{NE}_i . \quad (2.4)$$

Die **FNE** mit dem Faktorsystem (2.1) und der Koeffizientenfolge (2.2) ist der Ausdruck  $\overset{n}{NE}_0$  dieser Folge. Für die Bezugnahme auf die einzelnen strukturellen Elemente der FNE werden folgende Vereinbarungen eingeführt:

1. Die Zahl  $c_k$  heißt **freier Koeffizient**. Der Teilausdruck, der aus der FNE durch Wegnahme des freien Koeffizienten entsteht, heißt **Restausdruck**. Der Restausdruck kann auch als eine FNE mit dem freien Koeffizienten 0 interpretiert werden, in diesem Kontext wird er **Restsumme** genannt.
2. Die Zahl  $c_{k+pos}$  heißt Koeffizient, der Bruch  $\frac{a_{k+pos}}{b_{k+pos}}$  heißt Faktor,  $a_{k+pos}$  und  $b_{k+pos}$  heißen *Zähler* bzw. *Nenner*, jeweils *an der Position pos* der FNE.
3. Die Folge  $\left[ \overset{n}{NE}_n, \overset{n}{NE}_{n-1}, \dots, \overset{n}{NE}_0 \right]$  heißt **Konstruktionskette** und  $\overset{n}{NE}_{pos}$  heißt (geschachtelte) *Teilsumme an der Position pos*.
4. In Bezug auf die soeben ausgeführten Punkte 2 und 3 heißt *pos* *Positionszahl* und  $k+pos$  *Index* des entsprechenden strukturellen Elementes.

Um alle strukturellen Elemente der FNE übersichtlich darzustellen, wird die **Kompaktnotation**

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) \quad (2.5)$$

eingeführt. In diesem Zusammenhang lautet der Restausdruck

$$\text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) . \quad (2.6)$$

□

*Bemerkung 2.2* (Über Aufbau, Notation, Wert und Interpretation von FNEs).

1. Eine FNE ist durch ihre Kompaktnotation vollständig bestimmt. Der Zweck der Kompaktnotation ist es, in Anlehnung an Bemerkung 0.13 FNEs sowie ihre Manipulationen bequem zu notieren.
2. In der induktiven Konstruktion von  $\left[ \overset{n}{NE}_n, \overset{n}{NE}_{n-1}, \dots, \overset{n}{NE}_0 \right]$  sind Klammern einzusetzen. Die Klammern haben nur die syntaktische Bedeutung, den Anfang und das Ende eines Teilausdruckes kenntlich zu machen.
3. Als ein mathematischer Ausdruck hat eine FNE einen *Wert*, wenn den Koeffizienten, Zählern und Nennern konkrete numerische Werte zugewiesen werden. Dabei wird eine FNE nicht von links nach rechts, sondern von rechts nach links ausgewertet. Diese Reihenfolge der Auswertung ist daran zu erkennen, dass in der induktiven Definition gemäß (2.4) die Variable  $\overset{n}{NE}_i$  für den bereits definierten Ausdruck rechts steht.
4. Wenn es für die Manipulation von Ausdrücken notwendig ist, wird der Restausdruck einer FNE als Restsumme interpretiert, d. h. der freie Koeffizient 0 wird gedanklich hinzugefügt.

□

*Bemerkung 2.3* (Indexverschiebung und Verknüpfung von FNEs). Die Abhängigkeit der Koeffizienten, Zähler und Nenner einer FNE vom Index lässt sich jeweils als eine Funktion des Indexes interpretieren. In diesem Zusammenhang kann der Anfangsindex einer FNE durch eine entsprechende, auf alle diese drei Funktionen angewandte Indexverschiebung (siehe Punkt 2 von Bemerkung 0.17) willkürlich gesetzt werden, ohne die grundlegenden semantischen Eigenschaften der FNE zu ändern. Insbesondere bleibt der Wert einer FNE nach einer solchen Indexverschiebung erhalten. Die Anpassung des Anfangsindex, um eine FNE mit einem anderen Ausdruck zu verknüpfen, ist mit der Anpassung des Offsets eines Objekt-Moduls bei der Verbindung mit den anderen zu einem ausführbarem Programm zu vergleichen.

□

*Beispiel 2.4* (Definition und Kompaktnotation der FNE für  $e_7$ ). Der Ausdruck für  $e_7$  in (1.3) ist äquivalent zu dem folgenden

$$2 + \frac{1}{2} \left( 1 + \frac{1}{3} \left( 1 + \frac{1}{4} \left( 1 + \frac{1}{5} \left( 1 + \frac{1}{6} \left( 1 + \frac{1}{7} (1) \right) \right) \right) \right) \right) .$$

Dies ist eine FNE vom Grad 6. Die einzelnen Schritte der Konstruktion gemäß Definition 2.1 sind in Herleitung 2.1 dargestellt. Dabei ist eine Teilsumme  $\overset{6}{NE}_i$  in Klammern zu setzen, bevor sie in  $\overset{6}{NE}_{i-1}$  eingeht, um den Anfang und das Ende der Teilsumme syntaktisch kenntlich zu machen. Für jeden Anfangsindex haben wir entsprechend eine Kompaktnotation. Wir betrachten nun zwei Beispiele:

1. Mit dem Anfangsindex 0 lautet die Kompaktnotation  $2 + \text{RNE}\left(\left[\left(\frac{1}{i+1}, 1\right)\right]_{i=1}^6\right)$ , die Koeffizientenfolge  $[2]_{i=0}^0 \overset{+}{\text{seq}} [1]_{i=1}^6$  und das Faktorsystem  $\left[\frac{1}{i+1}\right]_{i=1}^6$ .
2. Mit dem Anfangsindex 1 lautet die Kompaktnotation  $2 + \text{RNE}\left(\left[\left(\frac{1}{i}, 1\right)\right]_{i=2}^7\right)$  die Koeffizientenfolge  $[2]_{i=1}^1 \overset{+}{\text{seq}} [1]_{i=2}^7$  und das Faktorsystem  $\left[\frac{1}{i}\right]_{i=2}^7$ .

□

$$\begin{aligned}
& \overset{6}{NE}_6 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 \\
& \overset{6}{NE}_5 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_5 + \frac{a_6}{b_6} \overset{6}{NE}_6 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{7} \overset{6}{NE}_6 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{7}(1) \\
& \overset{6}{NE}_4 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_4 + \frac{a_5}{b_5} \overset{6}{NE}_5 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{6} \overset{6}{NE}_5 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{6}(1 + \frac{1}{7}(1)) \\
& \overset{6}{NE}_3 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_3 + \frac{a_4}{b_4} \overset{6}{NE}_4 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{4} \overset{6}{NE}_4 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{5}(1 + \frac{1}{6}(1 + \frac{1}{7}(1))) \\
& \overset{6}{NE}_2 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_2 + \frac{a_3}{b_3} \overset{6}{NE}_3 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{4} \overset{6}{NE}_3 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{4}(1 + \frac{1}{5}(1 + \frac{1}{6}(1 + \frac{1}{7}(1)))) \\
& \overset{6}{NE}_1 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_1 + \frac{a_2}{b_2} \overset{6}{NE}_2 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{3} \overset{6}{NE}_2 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5}(1 + \frac{1}{6}(1 + \frac{1}{7}(1))))) \\
& \overset{6}{NE}_0 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} c_0 + \frac{a_1}{b_1} \overset{6}{NE}_1 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 2 + \frac{1}{2} \overset{6}{NE}_1 \overset{\bar{\bar{\text{expr}}}}{\text{expr}} 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5}(1 + \frac{1}{6}(1 + \frac{1}{7}(1)))))
\end{aligned}$$

Herleitung 2.1: Einzelne Schritte der Konstruktion der FNE für  $e_7$  laut Definition 2.1

**Definition 2.5** (Notation für die Bezugnahme auf einzelne strukturelle Elemente einer FNE). Es sei  $NE$  eine FNE.

1. Der freie Koeffizient von  $NE$  wird mit  $\overset{\text{ne}}{\text{FREE}}(NE)$  und der Restausdruck von  $NE$  wird mit  $\overset{\text{ne}}{\text{REM}}(NE)$  bezeichnet.
2. Der Koeffizient, der Faktor, der Zähler und der Nenner an der Position  $pos$  von  $NE$  werden entsprechend mit  $\overset{\text{ne}}{\text{COE}}(NE, pos)$ ,  $\overset{\text{ne}}{\text{FAC}}(NE, pos)$ ,  $\overset{\text{ne}}{\text{NUM}}(NE, pos)$  und  $\overset{\text{ne}}{\text{DEN}}(NE, pos)$  bezeichnet. Um den Grad  $n$  der FNE hervorzuheben, werden die Bezeichnungen  $\overset{\text{ne}}{\text{COE}}_n(NE, pos)$ ,  $\overset{\text{ne}}{\text{FAC}}_n(NE, pos)$ ,  $\overset{\text{ne}}{\text{NUM}}_n(NE, pos)$  und  $\overset{\text{ne}}{\text{DEN}}_n(NE, pos)$  entsprechend eingesetzt.
3. Die Teilsumme an der Position  $pos$  wird mit  $\overset{\text{ne}}{\text{SUB}}(NE, pos)$  bezeichnet. Um den Grad  $n$  der FNE hervorzuheben, wird  $\overset{\text{ne}}{\text{SUB}}_n(NE, pos)$  eingesetzt.

□

**Korrolar 2.6** (Einfache strukturelle Eigenschaften einer FNE). *Es sei  $NE$  eine gegebene FNE vom Grad  $n$ . So gelten die folgenden Eigenschaften:*

1.

$$NE \stackrel{val}{=} \text{FREE}^{\text{ne}}(NE) + \text{REM}^{\text{ne}}(NE) . \quad (2.7)$$

2. Die Konstruktionskette von  $NE$  ist identisch mit

$$\left[ \text{SUB}_n^{\text{ne}}(NE, n), \text{SUB}_n^{\text{ne}}(NE, n-1), \dots, \text{SUB}_n^{\text{ne}}(NE, 0) \right] . \quad (2.8)$$

*Beweis.* Offensichtlich aus Definition 2.2 und Definition 2.5. ■

**Lemma 2.7** (Anschauliche Form und expandierte Form von FNE). *Zu einer FNE mit der Notation (2.5) gibt es die **anschauliche Form***

$$c_k + \frac{a_{k+1}}{b_{k+1}}(c_{k+1} + \dots + \frac{a_{k+n-2}}{b_{k+n-2}}(c_{k+n-2} + \frac{a_{k+n-1}}{b_{k+n-1}}(c_{k+n-1} + \frac{a_{k+n}}{b_{k+n}}c_{k+n}))) \dots \quad (2.9)$$

und die **expandierte Form**

$$c_k + \frac{a_{k+1}}{b_{k+1}}c_{k+1} + \left(\frac{a_{k+1}}{b_{k+1}} \frac{a_{k+2}}{b_{k+2}}\right)c_{k+2} + \dots + \left(\frac{a_{k+1}}{b_{k+1}} \frac{a_{k+2}}{b_{k+2}} \dots \frac{a_{k+n}}{b_{k+n}}\right)c_{k+n} \quad (2.10)$$

als äquivalente mathematische Ausdrücke in dem Sinn, dass sie die gleichen Informationen enthalten und den gleichen Wert besitzen.

*Beweis.* Die Ausdrücke (2.9) und (2.10) enthalten die gleichen Informationen in dem Sinn, dass aus jedem von ihnen der Grad, das Faktorsystem und die Koeffizientenfolge der FNE zu entnehmen sind. Die anschauliche Form (2.9) resultiert aus der Definition 2.1, wenn die Konstruktionskette für die FNE analog zur Herleitung 2.1 aufgebaut wird. Der Ausdruck (2.10) ist durch Ausklammerung aus (2.9) entstanden, deshalb besitzt er den gleichen Wert wie (2.9). ■

*Bemerkung 2.8* (Expandierte Form und alternative Definition von FNE). Die expandierte Form (2.10) lautet in  $\sum$ - $\prod$ -Notation:

$$c_k + \sum_{i=k+1}^{k+n} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i . \quad (2.11)$$

Ein stärker formalisierter Beweis dafür, dass (2.11) den gleichen Wert wie die FNE (2.5) hat, kann induktiv im Sinn von Bemerkung 0.18 erfolgen. Dementsprechend wird die allgemeine Eigenschaft

$$\text{NE}_{pos}^n \stackrel{val}{=} c_{k+pos} + \sum_{i=k+pos+1}^{k+n} \left( \prod_{j=k+pos+1}^i \frac{a_j}{b_j} \right) c_i \quad (2.12)$$

der Konstruktionskette durch die Induktion über  $pos$  bewiesen. Für  $pos = 0$  wird die rechte Seite von (2.12) zu (2.11) und die linke Seite zu (2.5).

Mit Lemma 2.7 hätten wir den Begriff FNE durch

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) \stackrel{\text{def}}{=} c_k + \sum_{i=k+1}^{k+n} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i \quad (2.13)$$

definieren können. □

**Definition 2.9** (Wertanteil an der Position  $pos$ ). Es sei  $NE$  eine FNE mit der Notation (2.5). Der Wertanteil an der Position  $pos$  von  $NE$  ist

$$\begin{cases} c_k, & \text{falls } pos = 0; \\ \text{der numerische Wert von } \left( \prod_{j=k+1}^{k+pos} \frac{a_j}{b_j} \right) c_i, & \text{falls } 0 < pos \leq n; \\ 0, & \text{falls } pos > n. \end{cases}$$

□

**Korrolar 2.10.** Der Wert einer FNE ist die Summe der Wertanteile an allen ihrer Positionen.

*Beweis.* Durch Einbeziehung der expandierten Form der FNE ((2.11) bzw. (2.13) für die FNE mit der Notation (2.5)). ■

**Lemma 2.11** (Substitutionslemma für FNEs in Bezug auf deren Wert). Es seien  $NE_1$  eine FNE und  $NE_1^{\text{SUB}}$  eine Teilsumme von  $NE_1$ . Es sei  $NE_2$  eine FNE, für die gilt  $NE_2^{\text{SUB}} \stackrel{\text{SUB}}{=} NE_1^{\text{SUB}}$ . Es sei  $NE_2$  die FNE, die dadurch aus  $NE_1$  entsteht, dass  $NE_1^{\text{SUB}}$  durch  $NE_2^{\text{SUB}}$  ersetzt wird. So gilt  $NE_1 \stackrel{\text{SUB}}{=} NE_2$ .

*Beweis.* Es seien (2.5) die Kompaktnotation von  $NE_1$  (so ist  $n$  der Grad von  $NE_1$ ),  $p$  die Position von  $NE_1^{\text{SUB}}$  in  $NE_1$  (so ist  $p \leq n$ ), und  $n_2$  der Grad von  $NE_2$ . Betrachten wir im Folgenden die Konstruktionsketten für  $NE_1$  und  $NE_2$  ab Position  $p$ :

$$\left[ NE_1^{\text{SUB}} \equiv \text{SUB}_n^{\text{ne}}(NE_1, p), \text{SUB}_n^{\text{ne}}(NE_1, p-1), \dots, \text{SUB}_n^{\text{ne}}(NE_1, 0) \right]$$

und

$$\left[ NE_2^{\text{SUB}} \equiv \text{SUB}_{n_2}^{\text{ne}}(NE_2, p), \text{SUB}_{n_2}^{\text{ne}}(NE_2, p-1), \dots, \text{SUB}_{n_2}^{\text{ne}}(NE_2, 0) \right].$$

Aus

$$\text{SUB}_n^{\text{ne}}(NE_1, p) \equiv NE_1^{\text{SUB}} \stackrel{\text{SUB}}{=} NE_2^{\text{SUB}} \equiv \text{SUB}_{n_2}^{\text{ne}}(NE_2, p)$$

wird

$$c_{k+p-1} + \frac{a_{k+p}}{b_{k+p}} \text{SUB}_n^{\text{ne}}(NE_1, p) \stackrel{\text{SUB}}{=} c_{k+p-1} + \frac{a_{k+p}}{b_{k+p}} \text{SUB}_{n_2}^{\text{ne}}(NE_2, p),$$

d. h.

$$\text{SUB}_n^{\text{ne}}(NE_1, p-1) \stackrel{\text{SUB}}{=} \text{SUB}_{n_2}^{\text{ne}}(NE_2, p-1),$$

impliziert. Im nächsten Schritt wird

$$\text{SUB}_n^{\text{ne}}(NE_1, p-2) \stackrel{\text{SUB}}{=} \text{SUB}_{n_2}^{\text{ne}}(NE_2, p-2)$$

analoge hergeleitet. Der Vorgang wird solange wiederholt, bis

$$NE_1 \equiv \text{SUB}_n^{\text{ne}}(NE_1, 0) \stackrel{\text{SUB}}{=} \text{SUB}_{n_2}^{\text{ne}}(NE_2, 0) \equiv NE_2$$

erreicht ist. ■

Der Sachverhalt in Lemma 2.11 ist mehr oder weniger offensichtlich. Das Lemma wird trotzdem formuliert und bewiesen, um die Struktur und den Aufbau von FNEs zu verdeutlichen.

**Definition 2.12** (Erweiterungen von FNEs). Es sei  $NE$  eine FNE mit der Notation (2.5).

1. Die *linke Erweiterung*  $\overset{+}{head}$  verknüpft ein Paar aus Koeffizienten und Faktor  $\left(c_{k-1}, \frac{a_k}{b_k}\right)$  mit  $NE$  gemäß der Gleichung

$$\left(c_{k-1}, \frac{a_k}{b_k}\right) \overset{+}{head} NE \equiv c_{k-1} + \frac{a_k}{b_k} NE \equiv c_{k-1} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k}^{k+n} \right). \quad (2.14)$$

2. Die *rechte Erweiterung*  $\overset{+}{tail}$  verknüpft  $NE$  mit einem Paar aus Faktor und Koeffizienten  $\left(\frac{a_{k+n+1}}{b_{k+n+1}}, c_{k+n+1}\right)$  gemäß der Gleichung

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) \overset{+}{tail} \left( \frac{a_{k+n+1}}{b_{k+n+1}}, c_{k+n+1} \right) \equiv c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n+1} \right). \quad (2.15)$$

3. Die *Verkettung von FNEs*  $\overset{+}{ne}$  verknüpft  $NE$  mit einer FNE  $d_{k+n} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+n+1}^{k+n+m} \right)$  gemäß der Gleichung

$$\begin{aligned} c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) \overset{+}{ne} d_{k+n} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+n+1}^{k+n+m} \right) &\equiv c_k + \\ \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n-1} \overset{+}{seq} \left[ \left( \frac{a_{k+n}}{b_{k+n}}, c_{k+n} + d_{k+n} \right) \right]_{i=k+n}^{k+n} \overset{+}{seq} \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+n+1}^{k+n+m} \right) &. \end{aligned} \quad (2.16)$$

Durch Einsatz der Indexverschiebung falls notwendig lassen sich die oben definierten Operationen ohne Einschränkung ausführen.  $\square$

**Korollar 2.13** (Werte bei Erweiterungen von FNEs). *Im Anschluss an Definition 2.12 gelten die folgenden Gleichungen:*

$$\left(c_{k-1}, \frac{a_k}{b_k}\right) \overset{+}{head} NE \overset{=}{val} c_{k-1} + \frac{a_k}{b_k} NE, \quad (2.17)$$

$$NE \overset{+}{tail} \left(\frac{a_{k+n+1}}{b_{k+n+1}}, c_{k+n+1}\right) \overset{=}{val} NE + \left(\prod_{i=k+1}^{k+n} \frac{a_i}{b_i}\right) \times \frac{a_{k+n+1}}{b_{k+n+1}} \times c_{k+n+1} \quad (2.18)$$

und

$$NE \overset{+}{ne} NE_1 \overset{=}{val} NE + \left(\prod_{i=k+1}^{k+n} \frac{a_i}{b_i}\right) \times NE_1. \quad (2.19)$$

*Beweis.*

1. Gleichung (2.17) folgt unmittelbar aus (2.14).
2. **Zum Beweis von (2.18).** Das Rückwärtsschließen wird angewandt, d. h. die Zielgleichung wird durch äquivalente Umformungen in eine offensichtliche Gleichung überführt. Demnach wird die linke Seite durch die entsprechende Kompaktnotation gemäß (2.15) ersetzt, dann werden alle FNEs gemäß (2.13) in die expandierte Form überführt, die elementaren Rechenregeln der Arithmetik schließen den Beweis ab.
3. **Zum Beweis von (2.19).** In Analogie zum Beweis von (2.18) wird (2.19) anstelle von (2.15) benutzt.  $\blacksquare$



**Definition 2.14** (Einfache fne-Transformation einer Summe). Es sei  $\sum_{i=k}^n s_i \mid s_i \in \mathbb{Q}^+$  eine Summe. Die *einfache fne-Transformation* überführt  $\sum_{i=k}^n s_i$  in eine FNE wie folgt:

1. Falls der erste Summand  $s_k \in \mathbb{Z}$

$$\begin{aligned} \sum_{i=k}^{k+n} s_i &= s_k + (s_{k+1})1 + \left(\frac{s_{k+1}}{1} \frac{s_{k+2}}{s_{k+1}}\right)1 + \dots + \left(\frac{s_{k+1}}{1} \frac{s_{k+2}}{s_{k+1}} \frac{s_{k+3}}{s_{k+2}} \dots \frac{s_{k+n}}{s_{k+n-1}}\right)1 \\ &= s_k + \text{RNE} \left( \left[ \left( \left( \frac{s_{k+1}}{1} \right), 1 \right) \right]_{i=k+1}^{k+1} \overset{+}{\text{seq}} \left[ \left( \left( \frac{s_i}{s_{i-1}} \right), 1 \right) \right]_{i=k+2}^{k+n} \right). \end{aligned} \quad (2.20a)$$

2. Falls der erste Summand  $s_k \in \mathbb{Q} \setminus \mathbb{Z}$

$$\begin{aligned} \sum_{i=k}^{k+n} s_i &= (s_k)1 + \left(\frac{s_k}{1} \frac{s_{k+1}}{s_k}\right)1 + \dots + \left(\frac{s_k}{1} \frac{s_{k+1}}{s_k} \frac{s_{k+2}}{s_{k+1}} \dots \frac{s_{k+n}}{s_{k+n-1}}\right)1 \\ &= 0 + \text{RNE} \left( \left[ \left( \left( \frac{s_k}{1} \right), 1 \right) \right]_{i=k}^k \overset{+}{\text{seq}} \left[ \left( \left( \frac{s_i}{s_{i-1}} \right), 1 \right) \right]_{i=k+1}^{k+n} \right). \end{aligned} \quad (2.20b)$$

Die Schreibweise  $\left(\frac{s_i}{s_{i-1}}\right)$  hat dabei die Bedeutung, dass nicht  $s_i$  und  $s_{i-1}$  als Zähler und Nenner, sondern der Quotient  $\frac{s_i}{s_{i-1}}$  als Faktor gemeint ist: Für eine konkrete Anwendung werden die Funktionen für Zähler und Nenner so gewählt, dass deren Berechnung aus dem Index oder der Positionszahl möglichst einfach wird. Das Ergebnis der Transformation heißt *einfache geschachtelte* oder *einfache fne-Form* der Summe.

Aus (2.20a) und (2.20b) ist offensichtlich, dass die einfache fne-Form *den gleichen Wert* wie die Summe hat.  $\square$

*Bemerkung 2.15* (FNE als allgemeinere Form der Zahlendarstellung und die Grundidee der Spigot-Berechnung). Es seien eine Zahl  $r \in \mathbb{R}$  und eine Basis  $g \mid g \in \mathbb{N}, g \geq 2$  gegeben. Einerseits hat die Darstellung von  $r$  im Positionssystem zur Basis  $g$  die Semantik

$$r = c_0 \cdot c_1 c_2 \dots c_n = c_0 + \sum_{i=1}^n \frac{1}{g^i} c_i, \quad (2.21)$$

wobei der Ausdruck am rechten Ende von (2.21) die Form eines *g-adischen Ausdrucks* (vergl. z. B. [Wal92]) hat, und die *g-adische Darstellung* oder *g-adische Entwicklung* von  $r$  genannt wird. Andererseits kann der Ausdruck gemäß der Umformung

$$c_0 + \sum_{i=1}^n \frac{1}{g^i} c_i = c_0 + \sum_{i=1}^n \left( \prod_{j=1}^i \frac{1}{g} \right) c_i = c_0 + \text{RNE} \left( \left[ \left( \frac{1}{g}, c_i \right) \right]_{i=1}^n \right)$$

als eine FNE interpretiert werden; nämlich die FNE, deren sämtliche Faktoren gleich  $\frac{1}{g}$  sind und deren Koeffizientenfolge mit der Ziffernfolge der Darstellung von  $r$  zur Basis  $g$  zu identifizieren ist. Eine FNE ist demnach eine allgemeinere Form der Darstellung einer Zahl in einem Positionssystem. Diese Erkenntnis ist der Schlüssel zur Grundidee der Spigot-Berechnung: durch Verallgemeinerung des bekannten Verfahrens der Basis-Konvertierung (siehe z. B. [Knu97, Abschnitt 4.4.A]) ist ein allgemeineres Verfahren für die „Konvertierung“ einer FNE in eine *g-adische Darstellung*, insbesondere die gewöhnliche dezimale Darstellung, zu erwarten. Dieses allgemeinere Verfahren ist der Spigot-Algorithmus.  $\square$

**Definition 2.16** (Homogenes Faktorsystem). Ein Faktorsystem  $\left[ \frac{a_i}{b_i} \right]_{i=k+1}^{k+n}$  heißt *homogen*, wenn  $\frac{a_i}{b_i} = \frac{a}{b}$  für  $i \in \mathbb{Z} \mid k+1 \leq i \leq k+n$  gilt. Das Faktorsystem wird in diesem Fall vereinfachend mit  $\left[ \frac{a}{b} \right]$  bezeichnet.  $\square$

### 2.1.3 Normierte Form von FNEs

Laut der Semantik der Zahlendarstellung muss in (2.21) jeder Koeffizient  $c_i \mid 1 \leq i \leq n$  die Ungleichung  $c_k < g$  erfüllen, damit die Signifikanz der Anfangsziffern gewährleistet wird. Dies wird in der Welt von FNEs durch *normierte* FNEs imitiert. Die Überführung einer FNE in eine normierte FNE mit gleichem Wert durch die Modifizierung der Koeffizienten bei gleich bleibendem Faktorsystem ist der Kern des Spigot-Algorithmus. Die mathematischen Aspekte der Überführung werden in diesem Abschnitt untersucht.

**Definition 2.17** (Normierte Position und Position mit Übertrag). Es seien  $NE$  eine FNE mit der Notation (2.5) und  $pos$  eine Positionszahl für  $NE$ .

1.  $NE$  ist *normiert an der Position*  $pos$ , falls  $c_{k+pos} < b_{k+pos}$  gilt.
2.  $NE$  hat einen *Übertrag an der Position*  $pos$ , falls  $c_{k+pos} \geq b_{k+pos}$  gilt.

$\square$

**Definition 2.18** (Begriffe im Zusammenhang mit normierten FNEs).

1. Eine FNE heißt *normiert*, falls sie an allen Positionen normiert ist.
2. Ein RNE heißt *normiert*, falls er, als geschachtelte Restsumme gesehen, normiert ist.
3. In Bezug auf ein gegebenes Faktorsystem  $FS \equiv \left[ \frac{a_i}{b_i} \right]_{i=m}^n$  werden folgende Mengen definiert:

(a) Die *Menge der normierten FNEs von  $FS$* , bezeichnet mit  $\overset{\text{reg-set}}{\text{FNE}}(FS)$ , ist definiert durch

$$\overset{\text{reg-set}}{\text{FNE}}(FS) \stackrel{\text{def}}{=} \left\{ c_0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, x_i \right) \right]_{i=m}^n \right) \mid c_0 \in \mathbb{N}, x_i \in \mathbb{N}, x_i < b_i \right\} . \quad (2.22)$$

(b) Die *Menge der normierten Restausdrücke von  $FS$* , bezeichnet mit  $\overset{\text{reg-set}}{\text{RNE}}(FS)$ , ist definiert durch

$$\overset{\text{reg-set}}{\text{RNE}}(FS) \stackrel{\text{def}}{=} \left\{ \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, x_i \right) \right]_{i=m}^n \right) \mid x_i \in \mathbb{N}, x_i < b_i \right\} . \quad (2.23)$$

$\square$

**Definition 2.19** (Versetzung des Übertrags). Es sei  $NE$  eine FNE mit der Notation (2.5). Die *Versetzung des Übertrags an der Position*  $pos$  modifiziert  $NE$  wie folgt:

1. Der Koeffizient  $c_{k+pos}$  wird zu  $(c_{k+pos} \bmod b_{k+pos})$ .
2. Der Koeffizient  $c_{k+pos-1}$  wird zu  $c_{k+pos-1} + a_{k+pos}(c_{k+pos} \div b_{k+pos})$ .
3. Alle anderen Koeffizienten und das Faktorsystem bleiben unverändert.

Die Versetzung des Übertrags an der Position  $pos$  wird auch die *Versetzung des Übertrags von der Position  $pos$  an die Position  $pos - 1$*  genannt. In diesem Zusammenhang sind folgende Begriffe/Notationen relevant:

1. Das Ergebnis der Versetzung wird mit  $\overset{\text{ne}}{\text{CARRY}}(NE, pos)$  bezeichnet.
2. Der Zuwachs  $a_{k+pos}(c_{k+pos} \div b_{k+pos})$  der Position  $pos-1$  heißt *Übertragszahl* der Versetzung □

**Lemma 2.20** (Grundeigenschaft der Versetzung des Übertrags). *Es gilt*

$$\overset{\text{ne}}{\text{CARRY}}(NE, pos) \stackrel{\text{val}}{=} NE . \quad (2.24)$$

Außerdem ist  $\overset{\text{ne}}{\text{CARRY}}(NE, pos)$  normiert an der Position  $pos$  und hat das gleiche Faktorsystem wie  $NE$ .

*Beweis.* Die resultierende FNE ist normiert wegen der Eigenschaft der Operation "mod". Das Erhalten des Faktorsystems ist offensichtlich. Im Folgenden werden zwei Beweise für das Erhalten des Wertes ausgeführt. ■

*Erster Beweis für das Erhalten des Wertes.* Die Versetzung des Übertrags modifiziert die Teilsumme an der Position  $pos - 1$  wie in Herleitung 2.2, wodurch ihr Wert erhalten bleibt. Anschließend wird Lemma 2.11 angewandt und der Beweis ist damit abgeschlossen. ■

$$\begin{aligned} & \underbrace{c_{k+pos-1} + \frac{a_{k+pos}}{b_{k+pos}} \left( \underbrace{c_{k+pos}}_{((c_{k+pos} \div b_{k+pos})b_{k+pos} + c_{k+pos} \bmod b_{k+pos})} + \overset{\text{ne}}{\text{SUB}}(NE, pos + 1) \right)}_{\text{Die Teilsumme an der Position } pos - 1 \text{ vor der Manipulation (von } NE)} \\ & \stackrel{\text{val}}{=} c_{k+pos-1} + \frac{a_{k+pos}}{b_{k+pos}} \left( \underbrace{(c_{k+pos} \div b_{k+pos})b_{k+pos}}_{\text{ausklammern}} + c_{k+pos} \bmod b_{k+pos} + \overset{\text{ne}}{\text{SUB}}(NE, pos + 1) \right) \\ & \stackrel{\text{val}}{=} c_{k+pos-1} + \frac{a_{k+pos}}{b_{k+pos}} \underbrace{(c_{k+pos} \div b_{k+pos})b_{k+pos}}_{a_{k+pos}(c_{k+pos} \div b_{k+pos})} + \frac{a_{k+pos}}{b_{k+pos}} \left( c_{k+pos} \bmod b_{k+pos} + \overset{\text{ne}}{\text{SUB}}(NE, pos + 1) \right) \\ & \stackrel{\text{val}}{=} \underbrace{(c_{k+pos-1} + a_{k+pos}(c_{k+pos} \div b_{k+pos}))}_{\text{Die Teilsumme an der Position } pos - 1 \text{ nach der Manipulation (von } \overset{\text{ne}}{\text{CARRY}}(NE, pos))} + \frac{a_{k+pos}}{b_{k+pos}} \left( (c_{k+pos} \bmod b_{k+pos}) + \overset{\text{ne}}{\text{SUB}}(NE, pos + 1) \right) \end{aligned}$$

Herleitung 2.2: Das Erhalten des Wertes der Teilsumme an der Position  $pos - 1$  nach der Versetzung des Übertrags an der Position  $pos$

*Zweiter Beweis für das Erhalten des Wertes.* Beziehen wir Definition 2.9 und Korollar 2.10 ein. Ein Betrag von

$$\left( \prod_{i=k+1}^{k+pos} \frac{a_i}{b_i} \right) b_{k+pos}(c_{k+pos} \div b_{k+pos}) = \left( \prod_{i=k+1}^{k+pos-1} \frac{a_i}{b_i} \right) a_{k+pos}(c_{k+pos} \div b_{k+pos})$$

wird von der Position  $pos$  an die Position  $pos - 1$  transferiert, um  $\overset{\text{ne}}{\text{CARRY}}(NE, pos)$  aus  $NE$  zu bilden. Daher gilt (2.24). ■

Erinnern wir uns an die verbal-intuitive Einführung in Kapitel 1. Eine Versetzung des Übertrags ist mit einer Anwendung der Regel (1.9) gleichzusetzen. Dementsprechend beinhaltet Herleitung 2.2 den Beweis dafür, dass dabei der Wert der betreffenden Teilsumme der NE erhalten bleibt.

**Definition 2.21** (Normierungsfolge und normierte Form). Es sei  $NE$  eine FNE mit der Notation (2.5).

1. Die *Normierungsfolge* von  $NE$  ist die folgende absteigend indizierte Folge

$$\left[ \overset{\text{norm}}{NE}_n, \overset{\text{norm}}{NE}_{n-1}, \dots, \overset{\text{norm}}{NE}_0 \right]$$

von FNEs, die induktiv im Sinn von Bemerkung 0.18 durch die zwei Gleichungen

$$\overset{\text{norm}}{NE}_n \equiv NE \quad (2.25)$$

und

$$\overset{\text{norm}}{NE}_{i-1} \equiv \overset{\text{ne}}{\text{CARRY}}\left(\overset{\text{norm}}{NE}_i, i\right) \quad (2.26)$$

konstruiert ist.

2. Die *normierte Form* von  $NE$ , bezeichnet mit  $\overset{\text{ne}}{\text{NORM}}(NE)$ , ist das letzte Element (d. h. das Element mit dem Index 0) der Normierungsfolge. Es gilt also

$$\overset{\text{ne}}{\text{NORM}}(NE) \equiv \overset{\text{norm}}{NE}_0 \quad (2.27)$$

3. Der Koeffizient an der Position  $i$  von  $\overset{\text{norm}}{NE}_{i+1}$  heißt der *Normierungskoeffizient* der Position  $i$  von  $NE$  und wird mit  $\overset{\text{norm}}{\text{COE}}(NE, i)$  bezeichnet.

4. Die Übertragszahl der Versetzung des Übertrags an der Position  $i$  von  $\overset{\text{norm}}{NE}_i$ , um  $\overset{\text{norm}}{NE}_{i-1}$  zu bilden, heißt *Normierungsübertragszahl* der Position  $i$  von  $NE$  und wird mit  $\overset{\text{norm}}{\text{Carry}}(NE, i)$  bezeichnet.

□

Erinnern wir uns wieder an die verbal-intuitive Einführung in Kapitel 1. Die Normierungsfolge der FNE in (1.5) ist z. B. in Herleitung 1.2 ab dritter Zeile dargestellt. Auch aus Herleitung 1.3 und Herleitung 1.4 sind Normierungsfolgen von FNEs zu entnehmen.

**Lemma 2.22** (Grundeigenschaft der normierten Form). *Es gilt*

$$\overset{\text{ne}}{\text{NORM}}(NE) \stackrel{\text{val}}{=} NE \quad (2.28)$$

Außerdem ist  $\overset{\text{ne}}{\text{NORM}}(NE)$  normiert und hat das gleiche Faktorsystem wie  $NE$ .

*Beweis.* Wenden wir Lemma 2.20 an.  $\overset{\text{norm}}{NE}_{n-1}$  ist normiert an der Position  $n$ ,  $\overset{\text{norm}}{NE}_{n-2}$  ist zusätzlich normiert an der Position  $n - 1$ , usw. Der Vorgang wird solange wiederholt, bis die NE  $\overset{\text{norm}}{NE}_0$  erreicht wird, die an allen Positionen von  $n$  bis 1 normiert ist. Dabei bleiben der Wert und das Faktorsystem während des ganzen Prozesses erhalten. ■

**Korrolar 2.23** (Grundeigenschaft des Normierungskoeffizienten). *Es gilt*

$$\overset{\text{norm}}{\text{COE}}(NE, pos) = \overset{\text{ne}}{\text{COE}}(NE, pos) + \overset{\text{norm}}{\text{Carry}}(NE, pos + 1) . \quad (2.29)$$

*Beweis.* Offensichtlich aus Definition 2.5 und Definition 2.21. ■

**Lemma 2.24** (Grundeigenschaft der Normierungsfolge). *Im Anschluss an Definition 2.21 gilt die Gleichung*

$$\overset{\text{norm}}{NE}_{pos-1} = c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+pos-1} \right) \overset{\text{ne}}{+} \overset{\text{ne}}{\text{NORM}} \left( 0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right) . \quad (2.30)$$

*Beweis.* Aus Definition 2.21 ist durch die Induktion<sup>2</sup> über  $pos$  mit dem Induktionsanfang  $pos = n$  ersichtlich, dass die Teilsumme an der Position  $pos - 1$  von  $\overset{\text{norm}}{NE}_{pos-1}$  die normierte Form

$$\overset{\text{ne}}{\text{NORM}} \left( c_{k+pos-1} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right) \quad (2.31)$$

ist. Wird mit  $d_{k+pos-1}$  der freie Koeffizient von

$$\overset{\text{ne}}{\text{NORM}} \left( 0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right)$$

bezeichnet, so ist  $c_{k+pos-1} + d_{k+pos-1}$  der freie Koeffizient von (2.31). Gleichung (2.16) der Definition 2.12 schließt den Beweis ab. ■

Die Grundeigenschaft der Normierungsfolge hat für die Abschätzung der größten benötigten Zahl große Bedeutung.

*Bemerkung 2.25* (Äquivalente Formen von Lemma 2.24). Der Beweis von Lemma 2.24 impliziert gleich zwei äquivalente Formen von (2.30), nämlich

$$\overset{\text{norm}}{\text{Carry}}(NE, pos) = \overset{\text{ne}}{\text{FREE}} \left( \overset{\text{ne}}{\text{NORM}} \left( 0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right) \right) \quad (2.32)$$

und

$$\overset{\text{norm}}{\text{COE}}(NE, pos - 1) = \overset{\text{ne}}{\text{COE}}(NE, pos - 1) + \overset{\text{ne}}{\text{FREE}} \left( \overset{\text{ne}}{\text{NORM}} \left( 0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right) \right) . \quad (2.33)$$

□

### 2.1.4 Der Normierungsalgorithmus und seine Korrektheit

Die im vorigen Abschnitt untersuchte Überführung einer FNE in die normierte Form wird nun zu einem Algorithmus zusammengefasst, dessen Korrektheit dann in Übereinstimmung mit Bemerkung 0.19 nachvollzogen wird.

**Definition 2.26** (Normierungsalgorithmus). Der Algorithmus zur Berechnung der normierten Form einer FNE gemäß den Vorschriften in Definition 2.21 heißt der *Normierungsalgorithmus* und ist in Algorithmus 2.1 als Pseudocode notiert. □

<sup>2</sup>im Sinn von Bemerkung 0.18.

*Bemerkung 2.27* (Allgemeine Konventionen für die Spezifikation des Spigot-Algorithmus und seiner Teilprozeduren). Die Berechnungsprozeduren werden mit folgenden Konventionen spezifiziert:

1. **Spezifikationseinheit.** Die kleinste in sich abgeschlossene Spezifikationseinheit ist der *Algorithmus*. Mit einem Algorithmus werden die Teilprozeduren des Spigot-Algorithmus sowie der Spigot-Algorithmus selbst spezifiziert. Als eine Spezifikationseinheit setzt sich ein Algorithmus aus den *Vorbedingungen*, dem *lokalen Verarbeitungscode* und den *Nachbedingungen* zusammen. Die Vor- und Nachbedingungen bilden den *Spezifikationsteil* des Algorithmus. Der Spezifikationsteil beinhaltet den **Zweck** und der lokale Verarbeitungscode die **Realisierung** des Algorithmus.
  
2. **Parameter, freie Spezifikationsvariable und lokale Variable.** Ein Algorithmus soll bestimmte gewünschte Änderungen bewirken, die durch einen Eingangs- und einen Ausgangszustand charakterisiert sind. Die Zustandsindikatoren lassen sich in einer Spezifikation syntaktisch als **Parameter** auffassen. Der Eingangszustand ist mit den Vorbedingungen und der Ausgangszustand mit den Nachbedingungen zu beschreiben. Um Vor- und Nachbedingungen zu formulieren, werden **freie Spezifikationsvariable** zu Hilfe genommen. Sie enthalten Bezugspunkte für die Feststellung der Änderung der Parameter und sind *nur im Spezifikationsteil sichtbar*. Zur Realisierung des Algorithmus werden in der Regel **lokale Variable** zu Hilfe genommen, die *nur im lokalen Verarbeitungscode sichtbar* sind. In diesem Zusammenhang gelten folgende Regeln bzw. Bemerkungen:
  - (a) Ein Parameter ist sowohl im Spezifikationsteil als auch im lokalen Verarbeitungscode gültig und bezieht sich dabei auf das gleiche semantische Objekt. Um sich von lokalen Variablen und freien Spezifikationsvariablen zu unterscheiden, sind alle Parameter in einer *Parameterliste* aufgelistet. Ferner wird mit dem Schlüsselwort **Out** ein Parameter als Eingangs- und Ausgangsparameter markiert, d. h., er dient als Indikator sowohl für den Eingangs- als auch für den Ausgangszustand (vergl. "call by reference"). Ein Parameter ohne das Schlüsselwort **Out** dient nur als Eingangsparameter, d. h., er dient als Indikator nur für den Eingangs- und nicht für den Ausgangszustand (vergl. "call by value").
  - (b) Eine freie Spezifikationsvariable und eine lokale Variable mit dem gleichen Namen beziehen sich auf unterschiedliche semantische Objekte.
  - (c) Eine freie Spezifikationsvariable ist dadurch zu erkennen, dass sie im Spezifikationsteil, jedoch nicht in der Parameterliste vorkommt.
  - (d) Eine lokale Variable ist dadurch zu erkennen, dass sie im lokalen Verarbeitungscode, jedoch nicht in der Parameterliste vorkommt.
  
3. **Pseudosprache für den lokalen Verarbeitungscode.** Die Berechnungsprozedur im lokalen Verarbeitungscode wird als Pseudocode notiert. Die *Pseudosprache* soll möglichst einfach und leicht verständlich sein. Im Folgenden wird sie näher charakterisiert:
  - (a) Die elementaren Anweisungen (mit Ausnahme von Deklarationen) sind etwa durch die Nachahmung von Pascal entstanden.
  - (b) Im Hinblick auf Deklarations- und Interpretationssemantik wird etwa wie in der Sprachen von Maple oder Macsyma verfahren. Dementsprechend muss eine lokale Variable nicht deklariert werden, deren Typ ist durch den Kontext zu erschließen ist. Ihr Gültigkeitsbereich ist der ganze lokale Verarbeitungscode, in dem sie auftaucht.

- (c) Die Pseudosprache ist speziell auf die Charakteristika der Spigot-Berechnung zugeschnitten. Dementsprechend setzt sie sich aus einer möglichst kleinen Menge von Ausdrucksmitteln für *ganzzahlige Arithmetik*, *Array*, *Schleife* und *Aufruf von bekannten Algorithmen* zusammen.
- (d) Durch *Aufrufen von bekannten Algorithmen*, gegebenenfalls mit Parameterübergabe, lassen sich komplexere Algorithmen definieren. In diesem Sinn gilt ein Algorithmus in der vorliegenden Dissertation als bekannt, wenn er vor dem Aufruf in einem anderen Algorithmus definiert wurde.

Wir wollen nun die oben ausgeführten Konventionen anhand eines Beispiels erläutern.

- Die Berechnungsprozedur für die Normierung von FNEs wird durch Algorithmus 2.1 spezifiziert.
- Mit Hilfe von "**Require:**", "**DECLARE AS:**" und "**Ensure:**" wird die Spezifikation in die entsprechenden Blöcke für die Vorbedingungen, den lokalen Verarbeitungscode und die Nachbedingungen gegliedert.
- Der lokale Verarbeitungscode wird mit dem Schlüsselwort **DECLARE AS:** eingeleitet. Mit der durch 1 nummerierten Anweisung (der Anweisung 1) werden der Name des Algorithmus und die Parameter spezifiziert. Der Name ist *fneNormalize*, die Parameter sind *Num*, *Den*, *fne-len* und *Reg*. Das Schlüsselwort **Out** vor der Variablen *Reg* kennzeichnet *Reg* als Eingangs- und Ausgangsparameter. Alle anderen Parameter dienen nur als Eingangsparameter.
- Der eigentliche Verarbeitungscode des Algorithmus umfasst die Anweisungen 2 bis 8 und führt die gewünschten Änderungen an den Parametern mit Hilfe der lokalen Variablen *help*, *pos* und *carry* aus.
- Die Vorbedingung (2.34) wird mit dem Schlüsselwort **Require:** eingeleitet und spezifiziert die Eigenschaften der Parameter vor der Ausführung des lokalen Verarbeitungscode. Die Nachbedingungen (2.35) und (2.36) sind mit dem Schlüsselwort **Ensure:** nach der letzten Anweisung des lokalen Verarbeitungscode eingeleitet und spezifiziert die Eigenschaften der Parameter nach der Ausführung des lokalen Verarbeitungscode. Die Vor- und Nachbedingungen sind Aussagen über die Parameter und werden mit Hilfe der freien Spezifikationsvariablen  $a_j$ ,  $b_j$ ,  $c_j$  und *NE* formuliert.
- Nach der Art des Zugriffs sind *Num*, *Den* und *Reg* Felder (englisch: arrays) von ganzen Zahlen; *fne-len*, *help*, *pos* und *carry* ganze Zahlen. Die freie Spezifikationsvariable *NE* dient nur als Abkürzung für die durch  $a_j$ ,  $b_j$  und  $c_j$  definierte FNE.
- Algorithmus 2.1 gilt bei der Definition von Algorithmus 2.4 als bekannt und kann deshalb zur Spezifikation von Algorithmus 2.4 anhand des Namens *fneNormalize* aufgerufen werden. Algorithmus 2.4 ist dabei der Spigot-Algorithmus.

□

*Kommentar 2.28* (Arbeitsweise des Algorithmus 2.1). Die Arbeitsweise bzw. die Korrektheit des Algorithmus lässt sich konkret und anschaulich durch die Berechnungen von Normierungsfolgen in den Herleitungen 1.2, 1.3 und 1.4 nachvollziehen.

Im Folgenden begehen wir etwa den allgemeinen Weg, indem wir die grundlegenden Beziehungen des Algorithmus in Übereinstimmung mit seiner Spezifikation aufzeigen. Für die Bezugnahme

**Algorithmus 2.1** Normierungsalgorithmus**Require:**

$$\text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{\text{Num}[j]}{\text{Den}[j]}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv c_0 + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, c_j \right) \right]_{j=1}^n \right) \equiv NE \quad (2.34)$$

- 1: **DECLARE AS:** *fneNormalize* (Num, Den, fne-len, Out Reg)
- 2: *carry* := 0
- 3: **for** *pos* = fne-len **to** 1 **step** -1 **do**
- 4:   *help* = Reg[*pos*] + *carry*
- 5:   Reg[*pos*] := *help* mod Den[*pos*]
- 6:   *carry* := Num[*pos*] × (*help* ÷ Den[*pos*])
- 7: **end for**
- 8: Reg[0] := Reg[0] + *carry*

**Ensure:**

$$\text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \stackrel{\text{val}}{=} NE \quad (2.35)$$

**and**

$$\text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \in \overset{\text{reg-set}}{\text{FNE}} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^{\text{fne-len}} \right) \quad (2.36)$$

auf die Werte der Datenelemente während der Abarbeitung des Algorithmus führen wir die folgenden Vereinbarungen ein:

1.  $X^{\text{fne-len}+1}$  bezeichnet den Wert der Variable  $X$  vor der Ausführung der For-Schleife.
2.  $X^i$  bezeichnet den Wert der Variable  $X$  nach der Ausführung des Iterationsschrittes der For-Schleife, in dem der Variable  $pos$  der Wert  $i$  zugewiesen wird.
3.  $X^0$  bezeichnet den Wert der Variable  $X$  nach der Ausführung des Algorithmus.

Die Beziehung der Datenwerte des Algorithmus zur Definition 2.21 lautet:

1. Die Normierungsübertragszahl der Position  $i$  ist  $carry^i$ .
2. Der Normierungskoeffizient an der Position  $i$  ist  $help^i = \text{Reg}[i] + carry^{i+1}$ .
3. Die FNE  $NE_i^{\text{norm}}$  der Normierungskette lautet

$$\overset{\text{fne-len}+1}{\text{Reg}}[0] + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \overset{\text{fne-len}+1}{\text{Reg}}[j] \right) \right]_{j=1}^{\text{fne-len}-1} \overset{+}{\text{seq}} \underbrace{\left[ \left( \frac{a_j}{b_j}, \overset{\text{fne-len}+1}{\text{Reg}}[j] + \overset{\text{fne-len}+1}{carry} \right) \right]_{j=\text{fne-len}}^{\text{fne-len}}}_{\text{aktuelle Position}} \right) \quad (2.37)$$



für  $i = \mathbf{fne-len}$ ,

$$\mathbf{Reg}[0] + \mathbf{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] \right) \right]_{j=1}^{i-1} \overset{+seq}{\underbrace{\left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] + \mathit{carry} \right) \right]_{j=i}^i}_{\text{aktuelle Position}}} \overset{+seq}{\left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] \right) \right]_{j=i+1}^{\mathbf{fne-len}}} \right) \quad (2.38)$$

für  $1 < i < \mathbf{fne-len}$ ,

$$\mathbf{Reg}[0] + \mathbf{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] + \mathit{carry} \right) \right]_{j=1}^1 \overset{+seq}{\underbrace{\left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] \right) \right]_{j=1+1}^{\mathbf{fne-len}}}_{\text{aktuelle Position}}} \right) \quad (2.39)$$

für  $i = 1$ , und

$$\underbrace{\mathbf{Reg}[0] + \mathit{carry}}_{\text{aktuelle Position}} + \mathbf{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \mathbf{Reg}[j] \right) \right]_{j=1}^{\mathbf{fne-len}} \right) \quad (2.40)$$

für  $i = 0$ ; wobei (2.37) und (2.39) als spezielle Fälle von (2.38) betrachtet werden können, wenn  $[X]_{j=y+1}^y$  für alle  $X$  und  $y$  naheliegenderweise als leere Folge verstanden wird.

Die grundlegende Invarianz des Algorithmus ist der Wert der FNE mit dem originalen Faktorsystem, deren Koeffizienten wie folgt bestimmt sind:

1. der Koeffizient an der aktuellen Position  $pos$  ist  $\mathbf{Reg}[pos] + \mathit{carry}$ ,
2. der Koeffizient an einer beliebigen, nicht aktuellen Position  $pos$  ist  $\mathbf{Reg}[pos]$ ;

d. h. die FNE, die in Abhängigkeit von der aktuellen Position durch (2.37), (2.38), (2.39) bzw. (2.40) dargestellt wird. Durch die Anweisung 8 des Algorithmus ist (2.40) äquivalent zu der linken Seite von (2.35). Es ist nun offensichtlich, dass das Erhalten der grundlegenden Invarianz des Algorithmus durch Lemma 2.20 und die Eigenschaften (2.35), (2.36) des Endergebnisses durch Lemma 2.22 begründet sind.  $\square$

### 2.1.5 Spigot-Approximationen und Spigot-Algorithmus

**Definition 2.29** (Multiplikation mit einer Konstanten). Die Operation  $\overset{ne}{\times}_{coe}$  verknüpft eine Konstante  $K \in \mathbb{Z}$  mit einer FNE zu einer neuen FNE gemäß der Gleichung

$$K \overset{ne}{\times}_{coe} \left( c_0 + \mathbf{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=1}^n \right) \right) \equiv K \times c_0 + \mathbf{RNE} \left( \left[ \left( \frac{a_i}{b_i}, K \times c_i \right) \right]_{i=1}^n \right) . \quad (2.41)$$

$\square$

**Korrolar 2.30** (Wert bei der Multiplikation mit einer Konstanten). *Es seien NE eine FNE und  $K \in \mathbb{Z}$  eine Konstante. So gilt die Gleichung*

$$K \overset{ne}{\times}_{coe} NE \overset{val}{=} K \times NE . \quad (2.42)$$

*Beweis.* Durch Anwendung des Distributivgesetzes.  $\blacksquare$

### 2.1.6 Spigot-Algorithmus

**Definition 2.31** (Approximationssumme und Restausdruck). Es seien  $NE$  eine FNE,  $g \mid g \in \mathbb{N}$ ,  $g \geq 2$  eine Basis und  $i \in \mathbb{N}$  eine Iterationszahl. Die Spigot-Approximationssumme der Iterationszahl  $i$  zur Basis  $g$  von  $NE$ , bezeichnet mit  $\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i)$ ,<sup>3</sup> und der Spigot-Restausruck der Iterationszahl  $i$  zur Basis  $g$  von  $NE$ , bezeichnet mit  $\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i)$ , sind in ihrer wechselseitigen Abhängigkeit induktiv wie folgt definiert:

1. **Induktionsanfang.** Mit Hilfe des indizierten Hilfsparameters

$$d_0 \stackrel{\text{def}}{=} \overset{\text{ne}}{\text{FREE}} \left( \overset{\text{ne}}{\text{NORM}}(NE) \right) \quad (2.43)$$

definieren wir

$$\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, 0) \equiv d_0 \quad (2.44)$$

und

$$\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, 0) \equiv \overset{\text{ne}}{\text{REM}} \left( \overset{\text{ne}}{\text{NORM}}(NE) \right) . \quad (2.45)$$

2. **Induktionsschritt.** Mit Hilfe des indizierten Hilfsparameters

$$d_{i+1} \stackrel{\text{def}}{=} \overset{\text{ne}}{\text{FREE}} \left( \overset{\text{ne}}{\text{NORM}} \left( g \underset{\text{coe}}{\times} \overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i) \right) \right) \quad (2.46)$$

definieren wir

$$\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i+1) \equiv \overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i) + \frac{d_{i+1}}{g^{i+1}} \quad (2.47)$$

und

$$\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i+1) \equiv \overset{\text{ne}}{\text{REM}} \left( \overset{\text{ne}}{\text{NORM}} \left( g \underset{\text{coe}}{\times} \overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i) \right) \right) . \quad (2.48)$$

□

**Korollar 2.32** (Struktur der Approximationssumme). Mit den Vereinbarungen in Definition 2.31 gilt

$$\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i) = d_0 + \overset{\text{ne}}{\text{RNE}} \left( \left[ \left( \frac{1}{g}, d_j \right) \right]_{j=1}^i \right) , \quad (2.49)$$

d. h., eine Spigot-Approximationssumme ist eine FNE mit einem homogenen Faktorsystem.

*Beweis.* Durch die Induktion mit dem Induktionsanfang gemäß (2.44) und dem Induktionsschritt gemäß (2.47) ist die linke Seite von (2.49) sofort mit der expandierten Form der rechten Seite zu identifizieren. ■

**Satz 2.33** (Grundlegende Invarianz des Spigot-Algorithmus). Im Anschluss an Definition 2.31 gilt die Gleichung

$$NE \stackrel{\text{val}}{=} \overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i) + \frac{\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i)}{g^i} . \quad (2.50)$$

<sup>3</sup> Der Hauptteil "APR" der Bezeichnung deutet: "approximation" an.

$$\begin{aligned}
& \text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \underbrace{\frac{\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)}{g^i}} \\
&= \text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \frac{\underbrace{g \times \text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)}_{(2.42)}}{g^{i+1}} \\
&= \text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \frac{\underbrace{g \times_{\text{coe}} \text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)}_{(2.51)}}{g^{i+1}} \\
&= \text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \frac{\underbrace{d_{i+1} + \text{spigot}_{\text{fne}} \text{RNE}(NE, g, i + 1)}}{g^{i+1}} \\
&= \underbrace{\text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \frac{d_{i+1}}{g^{i+1}}}_{(2.47)} + \frac{\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i + 1)}{g^{i+1}} \\
&= \text{spigot}_{\text{fne}} \text{APR}(NE, g, i + 1) + \frac{\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i + 1)}{g^{i+1}}
\end{aligned}$$

Herleitung 2.3: Das Erhalten der Beziehung zwischen Approximationsumme und Restausdruck durch Induktion

*Beweis.* Aus (2.46), (2.48), (2.7) und (2.28) lässt sich

$$g \times_{\text{coe}} \text{spigot}_{\text{fne}} \text{RNE}(NE, g, i) \stackrel{=}{\text{val}} d_{i+1} + \text{spigot}_{\text{fne}} \text{RNE}(NE, g, i + 1) \quad (2.51)$$

herleiten. Die induktive Konstruktion von  $\text{spigot}_{\text{fne}} \text{APR}(NE, g, i + 1)$  und  $\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i + 1)$  aus  $\text{spigot}_{\text{fne}} \text{APR}(NE, g, i)$  und  $\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)$  offenbart sich in Herleitung 2.3, wodurch die Invarianz der Summe

$$\text{spigot}_{\text{fne}} \text{APR}(NE, g, i) + \frac{\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)}{g^i}$$

in Bezug auf  $i$  hergeleitet wird. Herleitung 2.4 schließt den Beweis ab. ■

**Definition 2.34** (Extraktionsalgorithmus). Im Anschluss an Definition 2.31 heißt der Algorithmus zur Berechnung der Koeffizienten von  $\text{spigot}_{\text{fne}} \text{APR}(NE, g, i)$  und  $\text{spigot}_{\text{fne}} \text{RNE}(NE, g, i)$  aus  $NE, g$  und  $i$  *Extraktionsalgorithmus*. Er ist in Algorithmus 2.2 als Pseudocode notiert. □

$$\begin{aligned}
& \overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i) + \frac{\overset{\text{spigot}}{\text{RNE}}(NE, g, i)}{g^i} \stackrel{=}{=} \underbrace{\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, 0)}_{(2.44)} + \frac{\overset{\text{spigot}}{\text{RNE}}(NE, g, 0)}{g^0} \\
& \stackrel{=}{=} \underbrace{\overset{\text{val}}{d_0}}_{(2.43)} + \underbrace{\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, 0)}_{(2.45)} \stackrel{=}{=} \overset{\text{val}}{\text{FREE}}\left(\overset{\text{ne}}{\text{NORM}}(NE)\right) + \overset{\text{val}}{\text{REM}}\left(\overset{\text{ne}}{\text{NORM}}(NE)\right) \\
& \qquad \qquad \qquad \stackrel{=}{=} \overset{\text{val}}{\text{NORM}}\left(\overset{\text{ne}}{NE}\right) \stackrel{=}{=} \overset{\text{val}}{NE} \text{ ,} \quad (2.7) \quad (2.28)
\end{aligned}$$

Herleitung 2.4: Zum Abschluss des Beweises der grundlegenden Invarianz des Spigot-Algorithmus

*Kommentar 2.35* (Arbeitsweise des Algorithmus 2.2). Die Arbeitsweise bzw. die Korrektheit des Algorithmus lässt sich konkret und anschaulich durch die semantische Kombination von Herleitung 1.2 und Herleitung 1.3, die der Extraktion der Dezimalziffern aus  $e_7$  entspricht, anschaulich nachvollziehen.

Im Folgenden begehen wir den allgemeinen Weg, indem wir die grundlegenden Beziehungen des Algorithmus in Übereinstimmung mit seiner Spezifikation aufzeigen. Für die Bezugnahme auf die Werte der Datenelemente während der Abarbeitung des Algorithmus führen wir die folgenden Vereinbarungen ein:

1.  $\overset{0}{X}$  bezeichnet den Wert der Variable  $X$  vor der Ausführung der äußeren For-Schleife.
2.  $\overset{idx}{X}$  bezeichnet den Wert der Variable  $X$  nach der Ausführung des Iterationsschrittes der äußeren For-Schleife, bei dem der Variable  $ite$  den Wert  $idx$  zugewiesen wird.

Es gilt die Gleichung

$$\mathbf{g} \times \text{RNE} \left( \left[ \left[ \left( \frac{a_j}{b_j}, \overset{i}{\text{Reg}}[j] \right) \right]_{j=1}^{\text{fne-len}} \right] \right) \stackrel{=}{=} \overset{ite}{\text{carry}} + \text{RNE} \left( \left[ \left[ \left( \frac{a_j}{b_j}, \overset{i+1}{\text{Reg}}[j] \right) \right]_{j=1}^{\text{fne-len}} \right] \right) \quad (2.52)$$

aufgrund der Eigenschaft des aus den Anweisungen 4 bis 9 bestehenden Blockes; der Block ist nämlich analog zu Algorithmus 2.1, und anhand dieser Analogie lässt sich (2.52) aus (2.35) ableiten. Gleichung (2.52) entspricht der Induktion in Definition 2.31. Zu dieser Entsprechung gehören noch folgende Beziehungen:

1. 
$$\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, \mathbf{g}, ite) \equiv \overset{ite}{\text{Res}}[0] + \text{RNE} \left( \left[ \left[ \left( \frac{1}{\mathbf{g}}, \overset{ite}{\text{Res}}[j] \right) \right]_{j=1}^{ite} \right] \right) \quad (2.53)$$

und

2. 
$$\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, \mathbf{g}, ite) \equiv \text{RNE} \left( \left[ \left[ \left( \frac{a_j}{b_j}, \overset{ite}{\text{Reg}}[j] \right) \right]_{j=1}^{\text{fne-len}} \right] \right) \text{ .} \quad (2.54)$$

Bemerkungen über die Vor- und Nachbedingungen:

1. Die Eigenschaft (2.57) stellt die grundlegende Invarianz der Spigot-Berechnung dar und ist unter der Berücksichtigung von (2.53) und (2.54) äquivalent zur Gleichung (2.50), die in Satz 2.33 bewiesen ist.
2. Die Eigenschaften (2.57) und (2.58) sind von (2.55) unabhängig.
3. Die Ableitung von (2.59) aus (2.55) entspricht dem Beweis von Lemma 2.43.

□

---

**Algorithmus 2.2** Extraktionsalgorithmus
 

---

**Require:**

$$c_0 + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, c_j \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv NE \in \text{FNE}^{\text{reg-set}} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^n \right) \quad (2.55)$$

**and**

$$\text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{\text{Num}[j]}{\text{Den}[j]}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv NE \quad (2.56)$$

- 1: **DECLARE AS:** *extractInBase* (Num, Den, fne-len, ite-num, g, **Out** Reg, **Out** Res)
- 2: Res[0] := Reg[0]
- 3: **for** ite = 1 **to** ite-num **step** 1 **do**
- 4:   carry := 0
- 5:   **for** pos = fne-len **to** 1 **step** -1 **do**
- 6:     help := g × Reg[pos] + carry
- 7:     Reg[pos] := help mod Den[pos]
- 8:     carry := Num[pos] × (help ÷ Den[pos])
- 9:   **end for**
- 10: Res[ite] := carry
- 11: **end for**

**Ensure:**

$$\text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{ite-num}} \right) + \frac{\text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right)}{g^{\text{ite-num}}} \stackrel{=}{\text{val}} NE \quad (2.57)$$

**and**

$$\text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \in \text{RNE}^{\text{reg-set}} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^{\text{fne-len}} \right) \quad (2.58)$$

**and**

$$\forall RNE \in \text{RNE}^{\text{reg-set}} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^n \right). RNE \leq K \quad \Rightarrow \quad \left( \text{Res}[j] < K \times g \text{ for } 1 \leq j \leq \text{fne-len} \right) \quad (2.59)$$


---

**Definition 2.36** (Spigot-Approximation und Spigot-Algorithmus). Der Gedankengang in Definition 2.31 wird unter Benutzung gleicher Notationen fortgesetzt.

1. Die *Spigot-Approximation der Iterationszahl  $i$  zur Basis  $g$*  von  $NE$ , bezeichnet mit

**Algorithmus 2.3** Ergebnismormierungsalgorithmus**Require:**

$$\text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{res}-\text{len}} \right) \equiv NE \equiv d_0 + \text{RNE} \left( \left[ \left( \frac{1}{g}, d_j \right) \right]_{j=1}^{\text{res}-\text{len}} \right)$$

- 1: **DECLARE AS:** *resNormalize* (*g*, *res - len*, **Out** *Res*)
- 2: *carry* := 0
- 3: **for** *pos* = *res - len* **to** 1 **step** -1 **do**
- 4:   *help* := *Res[pos]* + *carry*
- 5:   *Res[pos]* := *help* mod *g*
- 6:   *carry* := *help* ÷ *g*
- 7: **end for**
- 8: *Res[0]* := *Res[0]* + *carry*

$$\text{Ensure: } \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{res}-\text{len}} \right) \stackrel{=}{val} NE$$

$$\text{and } \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{res}-\text{len}} \right) \in \text{FNE} \left( \left[ \frac{1}{g} \right]_{i=1}^{\text{res}-\text{len}} \right)$$

$\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, i)$ ,<sup>4</sup> ist definiert durch

$$\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, i) \equiv \text{NORM}^{\text{ne}} \left( \overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i) \right). \quad (2.60)$$

2. Der Algorithmus zur Berechnung der Koeffizienten von  $\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, i)$  aus  $\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, i)$  heißt der *Ergebnismormierungsalgorithmus* und ist in Algorithmus 2.3 als Pseudocode notiert.
3. Der Algorithmus zur Berechnung der Koeffizienten von  $\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, i)$  aus *NE*, *g* und *i* heißt der *Spigot-Algorithmus* und ist in Algorithmus 2.4 bzw. Algorithmus 2.5 als Pseudocode notiert.

□

**Satz 2.37** (Grundgleichung des Spigot-Algorithmus). *Im Anschluss an Definition 2.36 gilt die Gleichung*

$$NE = \overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, i) + \frac{\overset{\text{spigot}}{\text{RNE}}_{\text{fne}}(NE, g, i)}{g^i}. \quad (2.61)$$

*Beweis.* Offensichtlich aus Satz 2.33 und Lemma 2.22 ■

*Bemerkung 2.38* (Über Spigot-Algorithmus und seine Teilprozeduren).

1. Algorithmus 2.3 ist ein spezieller Fall von Algorithmus 2.1. Diese Tatsache ist dadurch bedingt, dass es bei Algorithmus 2.3 um die Normierung von *homogenen* und bei Algorithmus 2.1 um die Normierung von *beliebigen* FNEs geht. Somit kann man Algorithmus 2.3 aus Algorithmus 2.1 durch Spezialisierung ableiten.

<sup>4</sup> Der Hauptteil "APN" der Bezeichnung deutet "normalised approximation" (= „normierte Approximation“) an.

**Algorithmus 2.4** Spigot-Algorithmus mit Einbeziehung der Unterprogramme

**Require:**  $\text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{\text{Num}[j]}{\text{Den}[j]}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv$

$$NE \equiv c_0 + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, c_j \right) \right]_{j=1}^{\text{fne-len}} \right)$$

- 1: **DECLARE AS:** *SpigotWithSubRoutines* (Num, Den, fne-len, ite-num, g, **Out** Reg, **Out** Res)
- 2: **execute** *fneNormalize* (Num, Den, fne-len, Reg)
- 3: **execute** *extractInBase* (Num, Den, fne-len, ite-num, g, Reg, Res)
- 4: **execute** *resNormalize* (g, ite-num, Res)

**Ensure:**

$$\begin{aligned} & \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{ite-num}} \right) + \frac{\text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right)}{g^{\text{ite-num}}} \stackrel{=}{\text{val}} NE \\ & \text{and} \\ & \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{ite-num}} \right) \in \text{reg-set FNE} \left( \left[ \frac{1}{g} \right]_{i=1}^{\text{ite-num}} \right) \\ & \text{and} \\ & \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \in \text{reg-set RNE} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^{\text{fne-len}} \right) \end{aligned}$$

2. Der Spigot-Algorithmus ist eine Kombination aus Normierungsalgorithmus, Extraktionsalgorithmus und Ergebnismormierungsalgorithmus. In allen drei Teilprozeduren steht die Normierung im Mittelpunkt. In Algorithmus 2.5 erfolgt die Kombination durch direkte Komposition und in Algorithmus 2.4 durch Aufruf von Unterprogrammen.
3. Die Bedeutung der Inputnormierung für den Spigot-Algorithmus ist geringfügig, sie dient nur zur Sicherstellung der Eigenschaft (2.59) für die erste Runde der Extraktion.

□

**2.1.7 Obere Schranke der Zahlen bei der Ausführung des Spigot-Algorithmus**

**Definition 2.39** (Restschranke eines FFS). Es sei  $FS \stackrel{\text{def}}{=} \left[ \frac{a_i}{b_i} \right]_{i=m}^n$  ein Faktorsystem.

1. Eine reelle Zahl  $\overset{\square}{\Delta}_{\text{r-ffs}}(FS)$  heißt eine (ffs-)Restschranke von  $FS$ , falls die Eigenschaft

$$\forall NE \in \text{reg-set RNE}(FS) . NE \leq \overset{\square}{\Delta}_{\text{r-ffs}}(FS) \quad (2.62)$$

gilt.

2. Der *größte normierte Restausdruck* ist definiert durch

$$\text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, b_i - 1 \right) \right]_{i=m}^n \right), \quad (2.63)$$

dessen Wert *minimale (ffs-)Restschranke* von  $FS$  heißt und mit  $\overset{\min}{\Delta}_{\text{r-ffs}}(FS)$  bezeichnet wird.

□

**Algorithmus 2.5** Spigot-Algorithmus

$$\text{Require: } \text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{\text{Num}[j]}{\text{Den}[j]}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv \\ NE \equiv c_0 + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, c_j \right) \right]_{j=1}^{\text{fne-len}} \right)$$

1: **DECLARE AS:** *Spigot* (Num, Den, fne-len, ite-num, g, **Out** Reg, **Out** Res)  
2: *carry* := 0 {Beginn der Inputnormierung}  
3: **for** *pos* = fne-len **to** 1 **step** -1 **do**  
4:   *help* = Reg[*pos*] + *carry*  
5:   Reg[*pos*] := *help* mod Den[*pos*]  
6:   *carry* := Num[*pos*] × (*help* ÷ Den[*pos*])  
7: **end for**  
8: Reg[0] := Reg[0] + *carry* {Ende der Inputnormierung}  
9: Res[0] := Reg[0] {Beginn der Extraktion}  
10: **for** *ite* = 1 **to** ite-num **step** 1 **do**  
11:   *carry* := 0  
12:   **for** *pos* = fne-len **to** 1 **step** -1 **do**  
13:     *help* := g × Reg[*pos*] + *carry*  
14:     Reg[*pos*] := *help* mod Den[*pos*]  
15:     *carry* := Num[*pos*] × (*help* ÷ Den[*pos*])  
16:   **end for**  
17:   Res[*ite*] := *carry*  
18: **end for**{Ende der Extraktion}  
19: *carry* := 0 {Beginn der Ergebnismormierung}  
20: **for** *pos* = ite-num **to** 1 **step** -1 **do**  
21:   *help* := Res[*pos*] + *carry*  
22:   Res[*pos*] := *help* mod g  
23:   *carry* := *help* ÷ g  
24: **end for**  
25: Res[0] := Res[0] + *carry* {Ende der Ergebnismormierung}

**Ensure:**

$$\text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{ite-num}} \right) + \frac{\text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right)}{g^{\text{ite-num}}} \stackrel{=}{\text{val}} NE \\ \text{and} \\ \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{\text{ite-num}} \right) \in \text{FNE}^{\text{reg-set}} \left( \left[ \frac{1}{g} \right]_{i=1}^{\text{ite-num}} \right) \\ \text{and} \\ \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \in \text{RNE}^{\text{reg-set}} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^{\text{fne-len}} \right)$$



**Korrolar 2.40** (Eigenschaften der minimalen Restschranke). *Es gelten im Anschluss an Definition 2.39 für  $\Delta_{r\text{-ffs}}^{\min}(FS)$  die folgenden Eigenschaften:*

1. Die minimale Restschranke  $\Delta_{r\text{-ffs}}^{\min}(FS)$  ist eine Restschranke von FS.
2. Eine reelle Zahl  $\Delta_{r\text{-ffs}}^{\square}(FS)$  ist genau dann eine Restschranke von FS, wenn  $\Delta_{r\text{-ffs}}^{\min}(FS) \leq \Delta_{r\text{-ffs}}^{\square}(FS)$  gilt.

*Beweis.* Offensichtlich aus (2.23) und Definition 2.39. ■

**Lemma 2.41** (Grundlemma für die Abschätzung der Normierungsübertragszahl). *Sei NE eine FNE mit der Notation (2.5), für die alle  $c_i \geq 0$  sind. So gilt die Ungleichung*

$$\text{norm}_{\text{Carry}}(NE, pos) \leq \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right). \quad (2.64)$$

*Beweis.* Offensichtlich aus (2.32). ■

**Lemma 2.42** (Obere Schranke der Normierungskoeffizienten bei der Extraktion). *Es seien NE eine normierte FNE mit der Notation (2.5) und K eine ffs-Restschranke von  $\left[ \frac{a_i}{b_i} \right]_{i=k+pos}^{k+n}$ . Dann gilt die Ungleichung*

$$\text{norm}_{\text{COE}} \left( g \times_{\text{coe}}^{\text{ne}} NE, pos - 1 \right) \leq g \times (b_{k+pos-1} - 1 + K). \quad (2.65)$$

*Beweis.* Wenden wir Lemma 2.41 an, wobei NE in (2.64) mit  $g \times_{\text{coe}}^{\text{ne}} NE$  zu intanzieren ist. Die folgende Herleitung kann aufgebaut werden:

$$\begin{aligned} \text{norm}_{\text{Carry}} \left( g \times_{\text{coe}}^{\text{ne}} NE, pos \right) &\stackrel{(2.64), (2.41)}{\leq} \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, g \times c_i \right) \right]_{i=k+pos}^{k+n} \right) \\ &\stackrel{(2.41), (2.42)}{=} g \times \left( \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+pos}^{k+n} \right) \right) \leq g \times K. \end{aligned} \quad (2.66)$$

Wenden wir nun (2.29) an, wobei NE in (2.29) mit  $g \times_{\text{coe}}^{\text{ne}} NE$  und pos in (2.29) mit pos - 1 zu intanzieren ist. Die folgende Herleitung kann aufgebaut werden:

$$\begin{aligned} \text{norm}_{\text{COE}} \left( g \times_{\text{coe}}^{\text{ne}} NE, pos - 1 \right) &= \underbrace{\text{COE} \left( g \times_{\text{coe}}^{\text{ne}} NE, pos - 1 \right)}_{\leq g \times (b_{pos-1} - 1), \text{ weil NE normiert ist}} + \underbrace{\text{norm}_{\text{Carry}} \left( g \times_{\text{coe}}^{\text{ne}} NE, pos \right)}_{\leq g \times K \text{ wegen (2.66)}} \\ &\leq g \times (b_{pos-1} - 1) + g \times K = g \times (b_{k+pos-1} - 1 + K), \end{aligned}$$

die anschließend zu (2.65) abgekürzt wird. ■

**Lemma 2.43** (Obere Schranke der Zahlen bei der Ergebnism normierung). *Es seien NE eine NE mit der Notation (2.5) und K eine Restschranke von (2.1). Dann gelten für die Ergebnism normierung, d. h. die Berechnung von  $\text{APN}_{\text{fne}}^{\text{spigot}}(NE, g, \text{ite-num})$  aus  $\text{APR}_{\text{fne}}^{\text{spigot}}(NE, g, \text{ite-num})$ , die folgenden Eigenschaften:*

1. Die Normierungsübertragszahlen sind durch  $\frac{g \times K}{g-1}$  begrenzt.
2. Die Normierungskoeffizienten sind durch  $g \times K(1 + \frac{1}{g-1})$  begrenzt.

*Beweis.* Die Ergebnismormierung ist laut Definition 2.21 auszuführen, wobei  $NE$  in Definition 2.21 mit  $\overset{\text{spigot}}{\text{APR}}(NE, g, \text{ite-num})$  zu instanziiieren ist. Unter Beachtung von (2.46) aus Definition 2.31 lässt sich analog zu (2.66) die Ungleichung

$$d_i \leq g \times K \quad (2.67)$$

für alle  $i \geq 1$  herleiten.

Wenden wir nun Lemma 2.41 für  $\overset{\text{spigot}}{\text{APR}}_{\text{fne}}(NE, g, \text{ite-num}) = d_0 + \text{RNE}\left(\left[\left(\frac{1}{g}, d_j\right)\right]_{j=1}^{\text{ite-num}}\right)$  an, wobei  $INE$  mit  $d_0 + \text{RNE}\left(\left[\left(\frac{1}{g}, d_j\right)\right]_{j=1}^{\text{ite-num}}\right)$  zu instanziiieren ist. Es gilt die folgende Herleitung

$$\begin{aligned} & \overset{\text{norm}}{\text{Carry}}\left(d_0 + \text{RNE}\left(\left[\left(\frac{1}{g}, d_j\right)\right]_{j=1}^{\text{ite-num}}\right), \text{pos}\right) \stackrel{(2.64)}{\leq} \text{RNE}\left(\left[\left(\frac{1}{g}, d_j\right)\right]_{j=\text{pos}}^{\text{ite-num}}\right) \\ & \leq \text{RNE}\left(\left[\left(\frac{1}{g}, g \times K\right)\right]_{j=\text{pos}}^{\text{ite-num}}\right) \stackrel{\text{expandierte Form}}{=} \sum_{j=\text{pos}}^{\text{ite-num}} \frac{g \times K}{g^{j+1-\text{pos}}} < \frac{g \times K}{g-1}, \end{aligned} \quad (2.68)$$

die den Punkt 1 des Lemmas impliziert. Durch (2.29), (2.67) und (2.68) lässt sich der Punkt 2 des Lemmas herleiten. ■

## 2.2 INEs und Spigot-Berechnung reeller Zahlen

Die Anwendung des Spigot-Algorithmus zur Berechnung der Ziffern einer reellen Zahl  $r$  geschieht im Wesentlichen folgendermaßen: zuerst wird für  $r$  eine Reihe hergeleitet; dann wird in Bezug auf eine gewünschte Genauigkeit eine Teilsumme ausgewählt; danach wird die Teilsumme in eine FNE umgeformt; und anschließend wird auf die FNE der Spigot-Algorithmus angewandt.

Um die Formulierung von spigot-spezifischen Eigenschaften zu ermöglichen, war in [RW95] der Begriff  $INE^5$  auf intuitiver Basis als ein Ausdruck der Form

$$c_k + \frac{a_{k+1}}{b_{k+1}}(c_{k+1} + \dots + \frac{a_{k+n-2}}{b_{k+n-2}}(c_{k+n-2} + \frac{a_{k+n-1}}{b_{k+n-1}}(c_{k+n-1} + \frac{a_{k+n}}{b_{k+n}}(c_{k+n} + \dots \quad (2.69)$$

eingeführt worden.

In diesem Abschnitt wird der Begriff formalisiert und darauf aufbauend werden formale Werkzeuge für die Bewerkstelligung gewünschter Eigenschaften von Spigot-Berechnung entwickelt.

Einerseits ist der Zusammenhang zwischen  $INE$  und Reihe analog zu dem zwischen  $FNE$  und Summe; andererseits ist der Zusammenhang zwischen  $INE$  und  $FNE$  analog zu dem zwischen Reihe und Summe. Diese Analogien liegen der Formalisierung des Begriffes  $INE$  inhaltlich zugrunde.

<sup>5</sup> Von diesem Abschnitt an werden die folgenden Abkürzungen benutzt:

- $INE$  steht für unendlich geschachtelte Summe (engl.:  $INE = \text{Infinitely Nested Expression}$ ).
- $IFS$  steht für unendliches Faktorsystem (engl.:  $IFS = \text{Infinite Factor System}$ ).

### 2.2.1 Struktur, Konvergenzverhalten und Manipulation von INEs

**Definition 2.44** (INE und ihre strukturellen Elemente). Es seien ein *Anfangsindex*  $k \in \mathbb{Z}$ , ein *Faktorsystem*

$$\left[ \frac{a_i}{b_i} \right]_{i=k+1}^{\infty} \mid a_i, b_i \in \mathbb{N} - \{0\} \quad (2.70)$$

und eine *Koeffizientenfolge*

$$[c_i]_{i=k}^{\infty} \mid c_i \in \mathbb{Z} \quad (2.71)$$

gegeben. Die **INE** mit dem Faktorsystem (2.70) und der Koeffizientenfolge (2.71) lässt sich formal zunächst als das Tupel aus (2.70) und (2.71) definieren, in dem Sinn, dass damit alle ihre strukturellen Elemente gegeben sind. Um die Struktur und den Aufbau einer INE zu charakterisieren, sind folgende Begriffe und Notationen relevant:

1. Der formale Ausdruck

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{\infty} \right) \quad (2.72)$$

heißt die *Kompaktnotation* der INE. Sie soll in Verbindung mit Bemerkung 0.13 dazu dienen, eine INE vollständig, kurz, übersichtlich und bequem zu notieren.

2. Die FNE  $c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^n \right)$  heißt die *fne-Approximation vom Grad  $n$* . Für eine mit *INE* bezeichnete INE wird die fne-Approximation vom Grad  $n$  mit  $\overset{\text{fne}}{\text{APR}}_{\text{ine}}(\text{INE}, n)$  bezeichnet, es gilt also

$$\overset{\text{fne}}{\text{APR}}_{\text{ine}} \left( \underbrace{(2.72)}_{\text{EqE}}, n \right) \equiv c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right). \quad (2.73)$$

3. Die Reihe

$$c_k + \sum_{i=k+1}^{\infty} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i \quad (2.74)$$

heißt die *expandierte Reihe* der INE.

4. Die Begriffe *Koeffizient*, *Faktor*, *Zähler*, *Nenner* und deren Anbindung an einer Position der INE, die Begriffe *Positionszahl* und *Index*, der Begriff *freier Koeffizient*, sowie der Begriff *Wertanteile an Position  $j$*  (in Bezug auf eine Positionszahl  $j$ ) haben ähnliche Bedeutungen wie bei FNEs.

INEs lassen sich hauptsächlich anhand ihrer Kompaktnotation handhaben. Wir nennen (2.69) die *anschauliche Form* der INE. Diese Form ermöglicht zwar einen schnellen Einblick in die Semantik von INEs, ist aber für die Handhabung von INEs in der vorliegenden Dissertation nicht notwendig.  $\square$

**Definition 2.45** (ine-Darstellung einer Zahl). Es seien  $r$  eine reelle Zahl und *INE* eine INE mit der Notation (2.72). Dann heißt  $r$  der Grenzwert von *INE* bzw. *INE* eine *ine-Darstellung* von  $r$ , falls die Folge

$$\left[ c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) \right]_{n=0}^{\infty} \quad (2.75)$$

der fne-Approximationen von *INE* gegen  $r$  konvergiert.  $\square$

**Lemma 2.46** (Konvergenzverhalten von INEs). *Es sei INE eine INE mit der Notation (2.72).*

1. *INE hat genau dann einen Grenzwert  $r$ , wenn die expandierte Reihe von INE gegen  $r$  konvergiert.*
2. *Hat INE einen Grenzwert  $r$ , so gilt die Gleichung*

$$r \stackrel{\text{val}}{=} \underset{\text{ine}}{\overset{\text{fne}}{\text{APR}}}(INE, n) + \sum_{i=k+n+1}^{\infty} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i . \quad (2.76)$$

*Beweis.* Bemerken wir zunächst, dass die expandierte Form der Approximation vom Grad  $n$  genau der  $n$ -ten Teilsumme der expandierten Reihe entspricht. Dann wird der Grenzübergang ausgeführt. ■

**Definition 2.47** (Einfache ine-Transformation einer Reihe). Es sei  $\sum_{i=k}^{\infty} s_i \mid s_i \in \mathbb{Q}^+$  eine Reihe.

Analog zu Definition 2.14 überführt die *einfache ine-Transformation*  $\sum_{i=k}^{\infty} s_i$  in eine INE wie folgt:

1. Falls der erste Summand  $s_k \in \mathbb{Z}$

$$\sum_{i=k}^{\infty} s_i \mapsto s_k + \text{RNE} \left( \left[ \left( \left( \frac{s_{k+1}}{1} \right), 1 \right) \right]_{i=k+1}^{k+1} \overset{+}{\text{seq}} \left[ \left( \left( \frac{s_i}{s_{i-1}} \right), 1 \right) \right]_{i=k+2}^{\infty} \right) \quad (2.77a)$$

2. Falls der erste Summand  $s_k \in \mathbb{Q} \setminus \mathbb{Z}$

$$\sum_{i=k}^{\infty} s_i \mapsto 0 + \text{RNE} \left( \left[ \left( \left( \frac{s_k}{1} \right), 1 \right) \right]_{i=k}^k \overset{+}{\text{seq}} \left[ \left( \left( \frac{s_i}{s_{i-1}} \right), 1 \right) \right]_{i=k+1}^{\infty} \right) . \quad (2.77b)$$

Das Ergebnis der Transformation heißt *einfache ine-Form* der Reihe. □

**Lemma 2.48** (Konvergenz und Grenzwert der einfachen ine-Form einer Reihe). *Eine Reihe konvergiert genau dann gegen einen Grenzwert  $r$ , wenn deren einfache ine-Form den Grenzwert  $r$  hat.*

*Beweis.* Eine Reihe ist wegen Punkt 3 von Definition 2.44 und Definition 2.47 genau die expandierte Reihe ihrer ine-Form. Lemma 2.46 schließt den Beweis ab. ■

**Definition 2.49** (Erweiterungen von INEs). Es seien *INE* eine INE mit der Notation (2.72) und *NE* eine FNE mit der folgenden Notation

$$c_{k-m} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k-m+1}^{k-1} \overset{+}{\text{seq}} \left[ \left( \frac{a_k}{b_k}, d_k \right) \right]_{i=k}^k \right) . \quad (2.78)$$

1. Die *linke Erweiterung*  $\overset{+}{\text{head}}$  verknüpft ein Paar aus Koeffizienten und Faktor  $\left( c_{k-1}, \frac{a_k}{b_k} \right)$  mit *INE* gemäß der Gleichung

$$\left( c_{k-1}, \frac{a_k}{b_k} \right) \overset{+}{\text{head}} c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{\infty} \right) \equiv c_{k-1} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k}^{\infty} \right) . \quad (2.79)$$

2. Die Verkettung  $\overset{+}{ine}$  verknüpft  $NE$  mit  $INE$  gemäß der Gleichung

$$NE \overset{+}{ine} INE \equiv c_{k-m} + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k-m+1}^{k-1} \overset{+}{seq} \left[ \left( \frac{a_k}{b_k}, d_k + c_k \right) \right]_{i=k}^k \overset{+}{seq} \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{\infty} \right). \quad (2.80)$$

Wie bei Definition 2.12 lassen sich die oben definierten Operationen durch Einsatz der Indexverschiebung im notwendigen Fall ohne Einschränkung ausführen.  $\square$

**Lemma 2.50** (Werte bei Erweiterungen von INEs). *Im Anschluss an Definition 2.49 gelten folgenden Gleichungen:*

$$\left( c_{k-1}, \frac{a_k}{b_k} \right) \overset{+}{head} INE \overset{=}{val} c_{k-1} + \frac{a_k}{b_k} INE \quad (2.81)$$

und

$$NE \overset{+}{ine} INE \overset{=}{val} NE + \left( \prod_{i=k-m+1}^k \frac{a_i}{b_i} \right) \times INE. \quad (2.82)$$

*Beweis.* Wenden wir (2.17) und (2.19) für die fne-Approximationen von  $INE$  an und führen den Grenzübergang aus.  $\blacksquare$

### 2.2.2 Formale Werkzeuge für die Untersuchung und Bewerkstelligung der Eigenschaften von Spigot-Berechnungen durch INE

Stellen wir zuerst die grundlegenden Beziehungen der Spigot-Berechnung durch eine INE in Definition 2.51, Definition 2.52, Satz 2.53 und Definition 2.54 zusammen.

**Definition 2.51** (Parameterisierte Kennwerte der Spigot-Berechnung durch eine INE). Es sei  $INE$  eine INE.

1. Die *Spigot-Approximationssumme der Iterationszahl  $i$  vom Grad  $n$  zur Basis  $g$  von  $INE$* , bezeichnet mit  $\overset{\text{spigot}}{\text{APR}}_{\text{ine}}(INE, g, n, i)$ , ist definiert durch

$$\overset{\text{spigot}}{\text{APR}}_{\text{ine}}(INE, g, n, i) \equiv \overset{\text{spigot}}{\text{APR}}_{\text{fne}} \left( \overset{\text{fne}}{\text{APR}}_{\text{ine}}(INE, n), g, i \right). \quad (2.83)$$

2. Der *Spigot-Resta Ausdruck der Iterationszahl  $i$  vom Grad  $n$  zur Basis  $g$  von  $INE$* , bezeichnet mit  $\overset{\text{spigot}}{\text{RNE}}_{\text{ine}}(INE, g, n, i)$ , ist definiert durch

$$\overset{\text{spigot}}{\text{RNE}}_{\text{ine}}(INE, g, n, i) \equiv \overset{\text{spigot}}{\text{RNE}}_{\text{fne}} \left( \overset{\text{fne}}{\text{APR}}_{\text{ine}}(INE, n), g, i \right). \quad (2.84)$$

3. Die *Spigot-Approximation der Iterationszahl  $i$  vom Grad  $n$  zur Basis  $g$  von  $INE$* , bezeichnet mit  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}}(INE, g, n, i)$ , ist definiert durch

$$\overset{\text{spigot}}{\text{APN}}_{\text{ine}}(INE, g, n, i) \equiv \overset{\text{spigot}}{\text{APN}}_{\text{fne}} \left( \overset{\text{fne}}{\text{APR}}_{\text{ine}}(INE, n), g, i \right). \quad (2.85)$$

$\square$

**Definition 2.52** (Spigot-Berechnung von reellen Zahlen). Es seien  $r$  eine reelle Zahl und  $INE$  eine ine-Darstellung von  $r$ . Die *Spigot-Berechnung* von  $r$  durch  $INE$  mit dem Approximationsgrad  $n$  und der Iterationszahl  $i$  zur Basis  $g$  ist die Berechnung von  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}}(INE, g, n, i)$  mit dem Spigot-Algorithmus. Der absolute Fehler der Berechnung wird mit  $\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i)$  bezeichnet.  $\square$

**Satz 2.53** (Grundgleichung für den absoluten Fehler der Spigot-Berechnung). Es sei  $INE$  eine ine mit der Notation (2.72). Es gilt für die Berechnung von  $\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i)$  die Gleichung

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) = \frac{\overset{\text{spigot}}{\text{RNE}}_{\text{ine}}(INE, g, n, i)}{g^i} + \sum_{i=k+n+1}^{\infty} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i . \quad (2.86)$$

*Beweis.* Durch Anwendung von (2.76), (2.85), (2.84) und (2.61), wobei die Variable  $NE$  in (2.61) mit  $\overset{\text{fine}}{\text{APR}}_{\text{ine}}(INE, n)$  zu instanzieren ist.  $\blacksquare$

**Definition 2.54** (Konvertierungsfehler und Approximationsfehler). Der erste Summand der rechten Seite von (2.86) heißt *Konvertierungsfehler* und wird mit  $\overset{\text{cnv}}{\text{ERR}}_{\text{ine}}(INE, g, n, i)$  bezeichnet, es gilt also

$$\overset{\text{cnv}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) \equiv \frac{\overset{\text{spigot}}{\text{RNE}}_{\text{ine}}(INE, g, n, i)}{g^i} . \quad (2.87)$$

Der zweite Summand der rechten Seite von (2.86) heißt *Approximationsfehler* (vom Grad  $n$ ) und wird mit  $\overset{\text{appr}}{\text{ERR}}_{\text{ine}}(INE, n)$  bezeichnet, es gilt also

$$\overset{\text{appr}}{\text{ERR}}_{\text{ine}}(INE, n) \equiv \sum_{i=k+n+1}^{\infty} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i . \quad (2.88)$$

Mit den eingeführten Bezeichnungen lässt sich (2.86) in der Form

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) = \overset{\text{cnv}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) + \overset{\text{appr}}{\text{ERR}}_{\text{ine}}(INE, n) \quad (2.89)$$

darstellen.  $\square$

Als nächstes werden die Begriffe Spigot-Schranke, ifs-Restschranke und ine-Restschranke als Maßstäbe der Eignung von IFs und INEs für Spigot-Berechnungen und zugleich als Hilfsmittel für die Bewerkstelligung ihrer Eigenschaften eingeführt.

**Definition 2.55** (Spigot-Schranke eines IFS). Es sei  $IFS$  ein IFS mit der Notation (2.70).

<sup>6</sup> Die Variable  $i$  innerhalb der Summe kommt als eine *gebundene* Variable vor, die nur dort sichtbar ist und das freie  $i$  schattiert.

1. Eine Funktion von zwei Variablen  $m$  und  $n$ , deren Bezeichnung, Definitionsbereich und Wertebereich durch

$$\overset{\square}{\Delta}_{\text{s-ifs}}(IFS)(m, n) : \{m \mid m \in \mathbb{N}, m \geq k + 1\} \times \{n \mid n \in \mathbb{N}, n \geq k + 1\} \longrightarrow \mathbb{R}$$

festgelegt wird, heißt eine (*funktionale*) *Spigot-Schranke* von  $IFS$ , wenn die Eigenschaft

$$\forall NE \in \overset{\text{reg-set}}{\text{RNE}} \left( \left[ \frac{a_i}{b_i} \right]_{i=m}^n \right) . NE \leq \overset{\square}{\Delta}_{\text{s-ifs}}(IFS)(m, n) \quad (2.90)$$

für  $n \geq m \geq k + 1$  gilt.

2. Die *minimale (funktionale) Spigot-Schranke* von  $IFS$ , bezeichnet mit  $\overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n)$ , hat den gleichen Definitionsbereich und Wertebereich wie eine Spigot-Schranke und ist durch die Gleichung

$$\overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n) \stackrel{=}{\text{val}} 0 + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, b_i - 1 \right) \right]_{i=m}^n \right) \mid n \geq m \geq k + 1 \quad (2.91)$$

definiert. □

**Lemma 2.56** (Eigenschaften der minimalen Spigot-Schranke). *Es gelten im Anschluss an Definition 2.55 für die minimale Spigot-Schranke  $\overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n)$  die folgenden Eigenschaften:*

1. Die *minimale Spigot-Schranke* ist eine *Spigot-Schranke*, d. h.

$$NE \in \overset{\text{reg-set}}{\text{RNE}} \left( \left[ \frac{a_i}{b_i} \right]_{i=m}^n \right) \Rightarrow NE \leq \overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n) . \quad (2.92)$$

2. Eine Funktion zweier Variablen  $m$  und  $n$   $\overset{\square}{\Delta}_{\text{s-ifs}}(IFS)(m, n)$  ist genau dann eine *Spigot-Schranke*, wenn

$$\forall n \geq m \geq k + 1 . \overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n) \leq \overset{\square}{\Delta}_{\text{s-ifs}}(IFS)(m, n) \quad (2.93)$$

*gilt.*

- 3.

$$n_1 < n_2 \Rightarrow \overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n_1) < \overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n_2) . \quad (2.94)$$

*Beweis.*

1. Punkt 1 und Punkt 2 des Lemmas lassen sich aus Korollar 2.40 herleiten.
2. Die Beziehung (2.94) ist ersichtlich, wenn die Kompaktnotation laut (2.91) für  $\overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n_1)$  und  $\overset{\min}{\Delta}_{\text{s-ifs}}(IFS)(m, n_2)$  jeweils in die expandierte Form gebracht wird. ■

**Definition 2.57** (Funktionale Restschranken für Fehlerabschätzung). Es seien  $INE$  eine INE mit der Notation (2.72) und  $IFS$  das Faktorsystem von  $INE$  mit der Notation (2.70).

1. Eine Funktion der Variablen  $n$ , deren Bezeichnung, Definitionsbereich und Wertebereich durch

$$\overset{\square}{\underset{\text{r-ifs}}{\Delta}}(IFS)(n) : \{n \mid n \in \mathbb{N}, n \geq k+1\} \longrightarrow \mathbb{R}$$

festgelegt wird, heißt eine (*funktionale*) (*ifs-*)*Restschranke des Faktorsystems IFS*, wenn die Ungleichung

$$\overset{\min}{\underset{\text{s-ifs}}{\Delta}}(IFS)(k+1, n) \leq \overset{\square}{\underset{\text{r-ifs}}{\Delta}}(IFS)(n) \quad (2.95)$$

gilt.

2. Eine Funktion der Variablen  $n$ , deren Bezeichnung, Definitionsbereich und Wertebereich durch

$$\overset{\square}{\underset{\text{r-ine}}{\Delta}}(INE)(n) : \{n \mid n \in \mathbb{N}, n \geq k+1\} \longrightarrow \mathbb{R}$$

festgelegt wird, heißt eine (*funktionale*) (*ine-*)*Restschranke von INE*, wenn die Ungleichung

$$\overset{\text{appr}}{\underset{\text{ine}}{\text{ERR}}}(INE, n) \leq \overset{\square}{\underset{\text{r-ine}}{\Delta}}(INE)(n) \quad (2.96)$$

gilt.

□

*Bemerkung 2.58* (Beziehung zwischen Spigot-Schranke, ffs-Restschranke, ifs-Restschranke und größtem normiertem Restausdruck). Stellen wir Definition 2.55 und Definition 2.39 gegenüber:

$\overset{\square}{\underset{\text{s-ifs}}{\Delta}}(IFS)(k+1, n)$  ist eine ffs-Restschranke und  $\overset{\min}{\underset{\text{s-ifs}}{\Delta}}(IFS)(k+1, n)$  ist die minimale ffs-Restschranke

des endlichen Faktorsystems  $\left[ \frac{a_i}{b_i} \right]_{i=k+1}^n$ . Außerdem ist die Beziehung zwischen den Begriffen ifs-Spigot-Schranke und ifs-Restschranke durch (2.95) deutlich zu erkennen. Damit sind die Begriffe ifs-Restschranke und ffs-Restschranke — folglich auch größter normierte Restausdruck — auf den Begriff Spigot-Schranke zurückzuführen. Der Begriff Spigot-Schranke hat dadurch die zentrale Bedeutung für die Charakterisierung der Eignung eines (unendlichen sowie endlichen) Faktorsystems für Spigot-Berechnungen. □

Nun werden durch Definition 2.59, Lemma 2.60 und Lemma 2.61 einige allgemeine Werkzeuge für die Konstruktion von Spigot-Schranke und Restschranken bereitgestellt.

**Definition 2.59** (Quotientenkriterium für INEs). Es seien  $INE$  eine INE mit der Notation (2.72) und dem Anfangsindex  $k$ ,  $q$  eine reelle Zahl mit  $0 < q < 1$  und  $m$  eine ganze Zahl mit  $m \geq k$ . Das *Quotientenkriterium mit dem Quotienten  $q$  und dem Anfangsindex  $m$*  für  $INE$  verlangt, dass die Ungleichung

$$\frac{a_{i+1}}{b_{i+1}} \frac{c_{i+1}}{c_i} \leq q \quad (2.97)$$

für alle  $i \geq m$  erfüllt werden muss. □

**Lemma 2.60** (*ine-Restschranke durch Quotientenkriterium*). *Setzen wir den Gedankengang in Definition 2.59 unter Benutzung gleicher Notationen fort. Erfüllt INE das Quotientenkriterium, so ist*

$$\overset{\text{quot: } q, m}{\underset{\text{r-ine}}{\Delta}}(INE)(n) \stackrel{\text{def}}{=} \left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \frac{q^{n+k-m}}{1-q} \quad (2.98)$$

eine funktionale Restschranke von  $INE$ .



*Beweis.* Mit Einbeziehung der Gleichung (2.96) in Definition 2.57 ist es hinreichend zu zeigen

$$\text{ERR}_{\text{ine}}^{\text{appr}}(INE, n) \leq \left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \frac{q^{n+k-m}}{1-q} . \quad (2.99)$$

Die Ungleichung (2.99) wird im Wesentlichen durch die Aufwertung der einzelnen Terme von  $\text{ERR}_{\text{ine}}^{\text{appr}}(INE, n)$  und Summierung der Ergebnisse gewonnen. Ein einzelner Term von  $\text{ERR}_{\text{ine}}^{\text{appr}}(INE, n)$

hat laut (2.88) die Form  $\left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i$ , dessen Aufwertung durch

$$\begin{aligned} \left( \prod_{j=k+1}^i \frac{a_j}{b_j} \right) c_i &= \left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \times \left( \prod_{j=m+1}^i \frac{a_j}{b_j} \frac{c_j}{c_{j-1}} \right) \\ &\leq \left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \times \left( \prod_{j=m+1}^i q \right) = \left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \times q^{i-m} . \end{aligned}$$

erfolgt. Die Ergebnisse der Aufwertung haben demnach die Form  $\left( \prod_{j=k+1}^m \frac{a_j}{b_j} \right) c_m \times q^{i-m}$ , bilden eine geometrische Reihe in Bezug auf die Variable  $i$  und lassen sich leicht zur rechten Seite von (2.99) summieren. ■

**Lemma 2.61** (Rundungslemma für Restschranke). *Das Faktorsystem  $\left[ \frac{a_i}{b_i} \right]_{i=m}^n$  hat die Restschranke*

$$a_m + \sum_{i=m+1}^n a_i \prod_{j=m}^{i-1} \frac{a_j}{b_j} .$$

*Beweis.* Durch die folgende Herleitung

$$\begin{aligned} \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, b_i - 1 \right) \right]_{i=m}^n \right) &= \sum_{i=m}^n \prod_{j=m}^i \frac{a_j}{b_j} (b_i - 1) = \sum_{i=m}^n \underbrace{(b_i - 1)}_{< b_i} \prod_{j=m}^i \frac{a_j}{b_j} < \sum_{i=m}^n b_i \prod_{j=m}^i \frac{a_j}{b_j} \\ &= b_m \frac{a_m}{b_m} + \sum_{i=m+1}^n b_i \prod_{j=m}^i \frac{a_j}{b_j} = a_m + \sum_{i=m+1}^n b_i \frac{a_i}{b_i} \prod_{j=m}^{i-1} \frac{a_j}{b_j} = a_m + \sum_{i=m+1}^n a_i \prod_{j=m}^{i-1} \frac{a_j}{b_j} \end{aligned}$$

in Verbindung mit Korrolar 2.40 und (2.13). ■

Lemma 2.61 wird oft benutzt, um zuerst eine Vergleichsbasis für Spigot-Schranken und ifs-Restschranken herzustellen; anschließend wird gegebenenfalls die Vorgehensweise von Lemma 2.60 herangezogen, um ifs-Restschranken bzw. Spigot-Schranken in analytisch einfache Form zu bringen. Dagegen kann Lemma 2.60 direkt zur Konstruktion von ine-Restschranken angewandt werden.

Kommen wir nun zu den anwendungsorientierten Werkzeugen für die Bewerkstellung der Eigenschaften von Spigot-Berechnungen.

**Satz 2.62** (Abschätzung des absoluten Fehlers mit ine- und ifs-Restschranke). *Der Gedankengang in Definition 2.57 wird unter Benutzung gleicher Notationen fortgesetzt. Ist  $\overset{\square}{\Delta}_{r\text{-ine}}(INE)(n)$  eine ine-Restschranke von INE und  $\overset{\square}{\Delta}_{r\text{-ifs}}(IFS)(n)$  eine ifs-Restschranke von IFS, so erfüllt die Spigot-Berechnung durch INE die Ungleichung*

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) \leq \frac{\overset{\square}{\Delta}_{r\text{-ifs}}(IFS)(n)}{g^i} + \overset{\square}{\Delta}_{r\text{-ine}}(INE)(n) . \quad (2.100)$$

*Beweis.* In Verbindung mit dem Sachverhalt, der durch Definition 2.54 und Satz 2.53 zum Ausdruck kommt, ist (2.100) im Wesentlichen dadurch hergeleitet, dass der Konvertierungsfehler und der Approximationsfehler jeweils aufgewertet werden. Im Folgenden wird die Herleitung näher betrachtet. Bemerken wir zuerst, dass

$$\overset{\text{spigot}}{\text{RNE}}_{\text{ine}}(INE, g, n, i) \in \overset{\text{reg-set}}{\text{RNE}} \left( \left[ \begin{array}{c} a_i \\ b_i \end{array} \right]_{i=k+1}^n \right)$$

gilt. Im Anschluss daran kann der Konvertierungsfehler gemäß (2.87) durch (2.92) und (2.95) mit  $\frac{\overset{\square}{\Delta}_{r\text{-ifs}}(IFS)(n)}{g^i}$  aufgewertet werden. Der Approximationsfehler wird durch (2.96) mit  $\overset{\square}{\Delta}_{r\text{-ine}}(INE)(n)$  aufgewertet. Die zwei Ungleichungen werden addiert und (2.89) wird anschließend benutzt, um (2.100) zu bekommen. ■

**Definition 2.63** (Charakteristische Funktionen der Spigot-Berechnung). Es seien *INE* eine INE mit dem Faktorsystem *IFS* und *g* eine natürliche Zahl mit  $g \geq 2$ .

1. Eine Funktion der Variablen *l*, deren Bezeichnung, Definitionsbereich und Wertebereich durch  $\overset{\square}{\chi}_{\text{spigot}}(INE, g)(l) : \mathbb{N} \longrightarrow \mathbb{N} \times \mathbb{N}$  festgelegt wird, heißt eine *charakteristische Funktion* der Spigot-Berechnung durch *INE* zur Basis *g*, wenn es gilt

$$\left( \overset{\square}{\chi}_{\text{spigot}}(INE, g)(l) = (n, i) \right) \Rightarrow \left( \overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, g, n, i) \leq \frac{1}{g^l} \right) . \quad (2.101)$$

2. Eine Funktion der Variablen *n*, deren Bezeichnung, Definitionsbereich und Wertebereich durch  $\overset{\square}{\Delta}_{\text{spigot}}(IFS, g)(n) : \mathbb{N} \longrightarrow \mathbb{N}$  festgelegt wird, heißt eine *charakteristische Schrankenfunktion* der Spigot-Berechnung durch *IFS* zur Basis *g*, wenn für alle *i* die für die Berechnung von  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}}(INE, g, n, i)$  benötigten Zahlen durch  $\overset{\square}{\Delta}_{\text{spigot}}(IFS, g)(n)$  begrenzt sind. □

**Lemma 2.64** (Genauigkeit und gültige Ziffern). *Es sei*

$$c_0 \cdot c_1 \dots c_l c_{l+1} \dots c_{l+k}$$

*die Approximation zur Basis g einer reeller Zahl r. Erfüllt die Approximation die folgenden Eigenschaften*

1. *der absolute Fehler liegt zwischen 0 und  $\frac{1}{g^{l+m}}$ ,*
2. *k < m oder mindestens eine der Ziffern  $c_{l+1}, \dots, c_{l+m}$  unterscheidet sich von g - 1,*

so ist  $c_0.c_1 \dots c_l$  die Darstellung von  $r$  zur Basis  $g$  mit  $l$  gültigen Nachkommaziffern.

*Beweis.*

1. Ist  $k < m$ , so liegt  $r$  zwischen  $\underbrace{c_0 \cdot c_1 \dots c_l}_{c_0 \cdot c_1 \dots c_l} c_{l+1} \dots c_{l+k}$  und  $\underbrace{c_0 \cdot c_1 \dots c_l}_{c_0 \cdot c_1 \dots c_l} c_{l+1} \dots c_{l+k} 0 \dots 1$ , d. h.,  $c_0.c_1 \dots c_l$  ist die Darstellung von  $r$  zur Basis  $g$  mit  $l$  Nachkommaziffern.

2. Ist  $c_{l+i} < g - 1$  mit  $1 \leq i \leq k$ , so liegt  $r$  zwischen

$$\underbrace{c_0 \cdot c_1 \dots c_l}_{c_0 \cdot c_1 \dots c_l} c_{l+1} \dots c_{l+k}$$

und

$$\underbrace{c_0 \cdot c_1 \dots c_l}_{c_0 \cdot c_1 \dots c_l} \dots \underbrace{(d_{l+i})}_{c_{l+i} \text{ oder } c_{l+i} + 1 \text{ als eine Ziffer}} \dots ,$$

d. h.,  $c_0.c_1 \dots c_l$  ist die Darstellung von  $r$  zur Basis  $g$  mit  $l$  Nachkommaziffern. ■

Ein Problem bei der Anwendung von Lemma 2.64 kann auftreten, wenn die Anzahl der aufeinander folgenden, am rechten Ende stehenden und mit  $g - 1$  übereinstimmenden Ziffern der Approximation sehr groß ist. Die Wahrscheinlichkeit, dass  $m$  aufeinander folgende Ziffern einer Zahl mit  $g - 1$  übereinstimmen, ist jedoch klein.

## 2.3 Ressourcenverbrauch der Spigot-Berechnung

In diesem Abschnitt wird der Ressourcenverbrauch der Spigot-Berechnung charakterisiert. Atomare Rechengänge und atomare Speichereinheiten des Spigot-Algorithmus werden gekennzeichnet. Das Maß des normierten Aufwandes wird jeweils für atomare Rechengänge und atomare Speichereinheiten eingeführt. Der Ressourcenverbrauch einer Spigot-Berechnung kann dann in Abhängigkeit von dem Approximationsgrad und der Iterationszahl anhand einfacher Gleichungen abgeschätzt werden. In diesem Kontext wird „normiert“ im Sinn von „durchschnittlich“ gebraucht.

**Definition 2.65** (Atomare Rechengänge und normierter Rechenaufwand).

1. Ein *atomarer Schritt der Spigot-Versetzung* versteht sich als eine einmalige Ausführung der Anweisungen 6 bis 8 in Algorithmus 2.2 (bzw. der Anweisungen 13 bis 15 in Algorithmus 2.5). Der normierte (im Sinne von durchschnittliche) Rechenaufwand dafür wird mit  $\epsilon_t^{\text{norm}}_{\text{carry}}(NE, g)$  bezeichnet.
2. Der Rechengang, der sich aus den Anweisungen 4 bis 6 in Algorithmus 2.1 zusammensetzt, kann als ein Spezialfall der Spigot-Versetzung betrachtet werden, nämlich wenn  $g = 1$  gilt. Der normierte Rechenaufwand dafür wird deshalb mit  $\epsilon_t^{\text{norm}}_{\text{carry}}(NE, 1)$  bezeichnet.
3. Ein *atomarer Schritt der Ergebnisnormierung* versteht sich als eine einmalige Ausführung der Anweisungen 4 bis 6 in Algorithmus 2.3. Der normierte Rechenaufwand dafür wird mit  $\epsilon_t^{\text{norm}}_{g\text{-carry}}(g)$  bezeichnet. □

**Definition 2.66** (Normierter Speicherbedarf atomarer Speichereinheiten). In Bezug auf Algorithmus 2.5 wird mit

1.  $\epsilon_m^{\text{norm}}(\text{NE})$  der normierte Speicherbedarf für ein Element des Arrays **Num**,
2.  $\epsilon_m^{\text{norm}}(\text{NE})$  der normierte Speicherbedarf für ein Element des Arrays **Den**,
3.  $\epsilon_m^{\text{norm}}(\text{NE})$  der normierte Speicherbedarf für ein Element des Arrays **Reg**

und

4.  $\epsilon_m^{\text{norm}}(\text{NE}, \mathbf{g})$  der normierte Speicherbedarf für ein Element des Arrays **Res**

bezeichnet. □

**Lemma 2.67** (Charakterisierung des Ressourcenverbrauchs der Spigot-Berechnung). *Der Ressourcenverbrauch von Algorithmus 2.5 wird betrachtet.*

1. *Der gesamte Rechenaufwand bei Approximationsgrad  $n$  und Iterationszahl  $i$ , bezeichnet mit  $\epsilon_t^{\text{total}}(\text{NE}, \mathbf{g}, n, i)$ , lässt sich durch*

$$\epsilon_t^{\text{total}}(\text{NE}, \mathbf{g}, n, i) = n \times \epsilon_t^{\text{norm}}(\text{NE}, 1) + n \times i \times \epsilon_t^{\text{norm}}(\text{NE}, \mathbf{g}) + i \times \epsilon_t^{\text{norm}}(\mathbf{g}) \quad (2.102)$$

oder

$$\epsilon_t^{\text{total}}(\text{NE}, \mathbf{g}, n, i) \approx n \times i \times \epsilon_t^{\text{norm}}(\text{NE}, \mathbf{g}) \quad (2.103)$$

abschätzen.

2. *Der gesamte Speicherbedarf bei Approximationsgrad  $n$  und der Iterationszahl  $i$ , bezeichnet mit  $\epsilon_m^{\text{total}}(\text{NE}, \mathbf{g}, n, i)$ , lässt sich durch*

$$\epsilon_m^{\text{total}}(\text{NE}, \mathbf{g}, n, i) = n \times \left( \epsilon_m^{\text{norm}}(\text{NE}) + \epsilon_m^{\text{norm}}(\text{NE}) + \epsilon_m^{\text{norm}}(\text{NE}) \right) + i \times \epsilon_m^{\text{norm}}(\text{NE}, \mathbf{g}) \quad (2.104)$$

abschätzen.

Lemma 2.67 lässt sich anschaulich aus Definition 2.65, Definition 2.66 und Algorithmus 2.5 einfach ableiten. Ein formaler Beweis wäre sehr aufwendig und für den Gegenstand der vorliegenden Dissertation nicht notwendig.

In der praktischen Anwendung erhöht sich der Aufwand durch die Ein- und Ausgabe sowie durch die einmalige Berechnung der Elemente von **Num** und **Den**. Außerdem lassen sich der gesamte Aufwand sowie seine Verteilung auf die einzelnen Anteile durch die konkrete Implementierung beeinflussen. Der Speicheraufwand kann z. B. auf Kosten des Rechenaufwandes dadurch erniedrigt werden, dass die Elemente von **Num** und **Den** nicht gespeichert, sondern jedes Mal direkt aus dem entsprechenden Index berechnet werden, wie es bei Algorithmus 1.1 und Algorithmus 1.2 der Fall ist.

## 2.4 Variationen des Spigot-Algorithmus

Die Ergebnismormierung kann gegebenenfalls — ohne die Korrektheit der Berechnung zu beeinträchtigen — gespart werden. Dafür ist Satz 2.68 heranzuziehen.

**Satz 2.68** (Spezialfall ohne Ergebnismormierung). *Existiert eine Restschranke  $K < 1$  für das Faktorsystem von  $NE$ , so ist  $\text{APR}^{\text{spigot}}(NE, g, i)$  für alle  $i$  normiert. Infolgedessen kann bei der Spigot-Berechnung durch  $NE$  die Ergebnismormierung gespart werden.*

*Beweis.* Für  $i \geq 1$  gilt  $d_i < g \times \frac{\text{ne}}{\text{coe}} \times \text{RNE}^{\text{spigot}}(NE, g, i - 1) < g \times K < g$ . ■

Bei der Spigot-Berechnung von  $e$  anhand von (1.1) ist z.B. die Ergebnismormierung nicht notwendig.

---

### Algorithmus 2.6 Spigot-Algorithmus mit Potenzüberbrückung

---

$$\text{Require: } \text{Reg}[0] + \text{RNE} \left( \left[ \left( \frac{\text{Num}[j]}{\text{Den}[j]}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \equiv \\ NE \equiv c_0 + \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, c_j \right) \right]_{j=1}^{\text{fne-len}} \right)$$

- 1: **DECLARE AS:** *SpigotWithPowerBridging* (Num, Den, fne-len, ite-num, g, m, **Out** Reg, **Out** Res)
- 2: **execute** *SpigotWithSubRoutines* (Num, Den, fne-len, ite-num,  $g^m$ , Reg, Rm)
- 3: **execute** *spigResPowBack* (Rm, g, m, ite-num, Res)

**Ensure:**

$$\text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{m \times \text{ite-num}} \right) + \frac{\text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right)}{g^{m \times \text{ite-num}}} \stackrel{=}{\text{val}} NE \\ \text{and} \\ \text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{g}, \text{Res}[j] \right) \right]_{j=1}^{m \times \text{ite-num}} \right) \in \text{reg-set FNE} \left( \left[ \frac{1}{g} \right]_{i=1}^{m \times \text{ite-num}} \right) \\ \text{and} \\ \text{RNE} \left( \left[ \left( \frac{a_j}{b_j}, \text{Reg}[j] \right) \right]_{j=1}^{\text{fne-len}} \right) \in \text{reg-set RNE} \left( \left[ \frac{a_i}{b_i} \right]_{i=1}^{\text{fne-len}} \right)$$


---

Für die Verbesserung der Effizienz des Spigot-Algorithmus stellt die *Potenzüberbrückung* eine interessante Möglichkeit dar. Potenzüberbrückung bedeutet, dass für die Berechnung zur Basis  $g$  zuerst das Ergebnis zur Basis  $g^m$  erzeugt und dann durch *Potenzrückführung* in die Basis  $g$  konvertiert wird. Die Potenzüberbrückung wird durch Definition 2.69 und Definition 2.70 formal spezifiziert und deren Einwirkung auf die Verbesserung der Effizienz wird in Bemerkung 2.71 diskutiert. Die Potenzüberbrückung geht auf Sale ([Sal68]) zurück.

**Definition 2.69** (Potenzrückführungsalgorithmus). Der *Potenzrückführungsalgorithmus* konvertiert die Darstellung einer Zahl zur Basis  $g^m$  in die Darstellung zur Basis  $g$  dadurch, dass jede Ziffer der Darstellung zur Basis  $g^m$  in einen Block von  $m$  Ziffern<sup>7</sup> zur Basis  $g$  konvertiert wird und die Blöcke in unveränderter Reihenfolge aneinander gereiht werden. Er ist in Algorithmus 2.7 als Pseudocode notiert. □

<sup>7</sup> Die Nullen sind gegebenenfalls am Anfang des Blocks hinzuzufügen.

**Algorithmus 2.7** Potenzrückführungsalgorithmus**Require:**

$$\text{Rm}[0] + \text{RNE} \left( \left[ \left( \frac{1}{\mathbf{g}^m}, \text{Rm}[j] \right) \right]_{j=1}^{\text{len}} \right) \equiv \text{NE}$$

**and**  $\left( \text{Rm}[j] < \mathbf{g}^m \text{ for } 1 \leq j \leq \text{len} \right)$

- 1: **DECLARE AS:** *spigResPowBack* (Rm, g, m, len, **Out Res**)
- 2:  $\text{Res}[0] := \text{Rm}[0]$
- 3: **for**  $\text{pos} = 1$  **to** len **do**
- 4:   {**Begin**  $\text{Rm}[\text{pos}] \mapsto \text{Res}[(\text{pos} - 1) \times m + 1] \dots \text{Res}[\text{pos} \times m]$   
    $\text{help} := \text{Rm}[\text{pos}]$ }
- 5:   **for**  $\text{count} = 0$  **to**  $m - 2$  **do**
- 6:      $\text{digit} := \text{help} \bmod \mathbf{g}$
- 7:      $\text{Res}[\text{pos} \times m - \text{count}] := \text{digit}$
- 8:      $\text{help} := \text{help} \div \mathbf{g}$
- 9:   **end for**
- 10:    $\text{Res}[(\text{pos} - 1) \times m + 1] := \text{help}$   
   {**Ende**  $\text{Rm}[\text{pos}] \mapsto \text{Res}[(\text{pos} - 1) \times m + 1] \dots \text{Res}[\text{pos} \times m]$ }
- 11: **end for**

**Ensure:**  $\text{Res}[0] + \text{RNE} \left( \left[ \left( \frac{1}{\mathbf{g}}, \text{Res}[j] \right) \right]_{j=1}^{\text{len} \times m} \right) \stackrel{\text{val}}{=} \text{NE}$

**and**  $\left( \text{Res}[j] < \mathbf{g} \text{ for } 1 \leq j \leq \text{len} \times m \right)$

Wenn  $g$  eine Potenz der Basis der internen Zahlendarstellung ist, ist die Potenzrückführung einfach ohne Berechnung durch Kopieren von Speicherzellen auszuführen.

**Definition 2.70** (Spigot-Algorithmus mit Potenzüberbrückung). Es seien eine geschachtelte Summe  $NE$ , eine Basis  $g \mid g \in \mathbb{N}, g \geq 2$ , ein *Exponent*  $m$  und eine *Iterationszahl*  $i \in \mathbb{N}$  gegeben. Der *Spigot-Algorithmus mit Potenzüberbrückung und dem Exponent  $m$*  berechnet eine mit  $\overset{\text{spigot}}{\text{APN}}_{\text{p-brid}}(NE, g, m, i)$  bezeichnete  $g$ -adische Approximation von  $NE$  wie folgt:

1. Auf  $NE$  wird der Spigot-Algorithmus angewandt, um  $\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g^m, i)$  zu berechnen.
2. Der  $g^m$ -adische Ausdruck  $\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g^m, i)$  wird zu dem gleichwertigen  $g$ -adischen Ausdruck  $\overset{\text{spigot}}{\text{APN}}_{\text{p-brid}}(NE, g, m, i)$  mit dem Potenzrückführungsalgorithmus konvertiert.

Der Algorithmus ist in Algorithmus 2.6 als Pseudocode notiert. □

*Bemerkung 2.71* (Potenzüberbrückung und Verbesserung der Effizienz). Der *normierte Aufwand* für einen *atomaren Schritt der Potenzrückführung*, der durch die Anweisungen 4 bis 10 in Algorithmus 2.7 zu realisieren ist, wird von nun an mit  $\overset{\text{norm}}{\epsilon_t}_{\text{p-back}}(g, m)$  bezeichnet. Der Sinn der

Potenzüberbrückung liegt darin, dass  $\overset{\text{spigot}}{\text{APN}}_{\text{p-brid}}(NE, g, m, i)$  und  $\overset{\text{spigot}}{\text{APN}}_{\text{fne}}(NE, g, m \times i)$  Approximationen für  $NE$  mit etwa gleicher Genauigkeit sind, während die wesentliche Teile des jeweiligen

Rechenaufwandes im Verhältnis

$$n \times i \times \underset{\text{carry}}{\text{norm}}_{\epsilon_t}(NE, \mathbf{g}^m) + i \times \underset{\text{p-back}}{\text{norm}}_{\epsilon_t}(g, m) \quad \text{zu} \quad n \times m \times i \times \underset{\text{carry}}{\text{norm}}_{\epsilon_t}(NE, \mathbf{g}) \quad (2.105)$$

stehen. Dieses Verhältnis ist bedeutend, wenn  $\mathbf{g}^m$  und  $\mathbf{g}$  in einem geeigneten Intervall liegen, so dass

$$\underset{\text{carry}}{\text{norm}}_{\epsilon_t}(NE, \mathbf{g}) \approx \underset{\text{carry}}{\text{norm}}_{\epsilon_t}(NE, \mathbf{g}^m)$$

gilt. □

## 2.5 Formalisierung der Spigot-Berechnung von $\sqrt{2}$ nach Stoschek

Die ursprüngliche Spigot-Berechnung von  $\sqrt{2}$  nach Stoschek findet man in [Sto97, Kap. 3, S. 66, 91, 92]. Die dabei zugrunde liegenden Ideen und Methoden haben einen entscheidenden Einfluss auf die Forschungsergebnisse in [Do01] sowie auf die Entwicklung und den Inhalt der vorliegenden Dissertation.

In [Sto97] wird die Spigot-Berechnung von  $\sqrt{2}$  nach Stoschek zum Zweck der Anregung von Ideen nur experimentell ausgeführt. In diesem Abschnitt soll darauf bezugnehmend die Mächtigkeit der in diesem Kapitel entwickelten formalen Werkzeuge gezeigt werden. Von der originalen INE nach Stoschek ausgehend werden die Bewerkstelligung einer gewünschten Genauigkeit, die Abschätzung der größten benötigten Zahl und die Bestimmung der optimalen Potenzüberbrückung auf mathematisch exakten Grundlagen ausgeführt.

Das komplexe Beispiel in diesem Abschnitt soll als Muster für die Anwendung des formalisierten Spigot-Algorithmus mit Ausnutzung all seiner positiven Eigenschaften dienen.

### 2.5.1 Die zugrunde liegende INE, ihre Restschranke und Spigot-Schranke

**Lemma 2.72** (Die ine-Darstellung  $\underset{\sqrt{2}}{\text{sim}}\text{INE}$  für  $\sqrt{2}$ ). *Für  $\sqrt{2}$  existiert eine ine-Darstellung, definiert bzw. bezeichnet durch*

$$\sqrt{2} = 1 + \text{RNE} \left( \left[ \left( \frac{2i-1}{4i}, 1 \right) \right]_{i=1}^{\infty} \right) \stackrel{\text{def}}{=} \underset{\sqrt{2}}{\text{sim}}\text{INE} . \quad (2.106)$$

*Beweis.* Siehe Herleitung 2.5. ■

**Lemma 2.73** (Die ifs-Restschranke  $\underset{\text{r-ifs}}{\text{sim}}\underset{\sqrt{2}}{\text{sim}}\Delta(\text{IFS})$ ). *Das Faktorsystem*

$$\underset{\sqrt{2}}{\text{sim}}\text{IFS} \stackrel{\text{def}}{=} \left[ \frac{2i-1}{4i} \right]_{i=1}^{\infty} \quad (2.107)$$

von  $\underset{\sqrt{2}}{\text{sim}}\text{INE}$  hat eine konstante ifs-Restschranke  $\underset{\text{r-ifs}}{\text{sim}}\underset{\sqrt{2}}{\text{sim}}\Delta(\text{IFS})$ , spezifiziert bzw. definiert durch

$$\forall n. \underset{\text{r-ifs}}{\text{sim}}\underset{\sqrt{2}}{\text{sim}}\Delta(\text{IFS})(n) = 4 . \quad (2.108)$$

$$\begin{aligned}
\sqrt{2} &= 2^{\frac{1}{2}} = (2^{-1})^{-\frac{1}{2}} = \left(\frac{1}{2}\right)^{-\frac{1}{2}} = \underbrace{\left(1 - \frac{1}{2}\right)^{-\frac{1}{2}}}_{\text{Binomialreihe, vergl. (3.36)}} = \sum_{i=0}^{\infty} \underbrace{\binom{-\frac{1}{2}}{i}}_{\left(-\frac{1}{2}\right)^i} \\
&= 1 + \sum_{i=1}^{\infty} \left( \prod_{j=0}^{i-1} \frac{-\frac{1}{2} - j}{j+1} \right) \left(-\frac{1}{2}\right)^i = 1 + \sum_{i=1}^{\infty} \left( \prod_{j=0}^{i-1} \frac{(-\frac{1}{2} - j)(-\frac{1}{2})}{j+1} \right) \\
&= 1 + \sum_{i=1}^{\infty} \left( \prod_{j=0}^{i-1} \frac{2j+1}{4(j+1)} \right) = 1 + \sum_{i=1}^{\infty} \left( \prod_{j=1}^i \frac{2j-1}{4j} \right) \\
&\stackrel{(2.77a)}{=} 1 + \text{RNE} \left( \left[ \left( \frac{2i-1}{4i}, 1 \right) \right]_{i=1}^{\infty} \right) \stackrel{\text{def}}{=} \frac{\text{sim}}{\sqrt{2}} \text{INE} .
\end{aligned}$$

Herleitung 2.5: Die ine-Darstellung  $\frac{\text{sim}}{\sqrt{2}} \text{INE}$  für  $\sqrt{2}$

*Beweis.* Unter Beachtung von (2.95) aus Definition 2.57 ist es hinreichend zu zeigen, dass

$$\min_{\text{s-ifs}} \left( \frac{\text{sim}}{\sqrt{2}} \text{IFS} \right) (1, n) < 4$$

gilt. Dazu wird Herleitung 2.6 herangezogen. ■

**Lemma 2.74** (Die ine-Restschranke  $\frac{\text{sim}}{\text{r-ine}} \frac{\text{sim}}{\sqrt{2}} \text{INE}$ ).  $\frac{\text{sim}}{\sqrt{2}} \text{INE}$  hat eine ine-Restschranke, bezeichnet bzw. definiert durch

$$\frac{\text{sim}}{\text{r-ine}} \frac{\text{sim}}{\sqrt{2}} \text{INE}(n) \stackrel{\text{def}}{=} \left(\frac{1}{2}\right)^{n-1} . \quad (2.109)$$

*Beweis.* Durch Anwendung des Quotientenkriteriums gemäß Lemma 2.60. In (2.98) ist  $\text{INE}$  mit  $\frac{\text{sim}}{\sqrt{2}} \text{INE}$ ,  $g$  mit  $\frac{1}{2}$  und  $m$  mit 1 zu instanzieren. Dadurch wird aus  $\frac{\text{quot: } \frac{1}{2}, 1}{\text{r-ine}} \frac{\text{sim}}{\sqrt{2}} \text{INE}(n)$  die rechte Seite von (2.109), die dann als  $\frac{\text{sim}}{\text{r-ine}} \frac{\text{sim}}{\sqrt{2}} \text{INE}(n)$  definiert wird. ■

**Lemma 2.75** (Die Spigot-Schranke  $\frac{\text{sim}}{\text{s-ifs}} \left( \frac{\text{sim}}{\sqrt{2}} \text{IFS} \right) (m, n)$ ). Das Faktorsystem  $\frac{\text{sim}}{\sqrt{2}} \text{IFS}$  mit der Definition (2.107) hat eine Spigot-Schranke, bezeichnet bzw. definiert durch

$$\frac{\text{sim}}{\text{s-ifs}} \left( \frac{\text{sim}}{\sqrt{2}} \text{IFS} \right) (m, n) \stackrel{\text{def}}{=} 4m . \quad (2.110)$$

*Beweis.* Analog zum Beweis von Lemma 2.73. ■



$$\begin{aligned}
\min_{s\text{-ifs}} \left( \overset{\text{sim}}{\Delta}(\text{IFS})_{\sqrt{2}} \right) (1, n) &= 0 + \text{RNE} \left( \left[ \left( \frac{2i-1}{4i}, 4i-1 \right) \right]_{i=1}^n \right) < 0 + \text{RNE} \left( \left[ \left( \frac{2i-1}{4i}, 4i-1 \right) \right]_{i=1}^{\infty} \right) \\
&< 0 + \text{RNE} \left( \left[ \left( \frac{2i-1}{4i}, 4i \right) \right]_{i=1}^{\infty} \right) = \sum_{i=1}^{\infty} \left( \prod_{j=1}^i \frac{2j-1}{4j} \right) 4i \\
&\qquad\qquad\qquad =_{\frac{1}{2} \frac{1}{2} \text{ f\"ur } i=1 \text{ und } < \frac{1}{2} \text{ f\"ur } i>1} \\
\sum_{i=1}^{\infty} \left( \frac{1}{2} \right)^{i+1} 4i &= 2 \underbrace{\sum_{i=1}^{\infty} i \left( \frac{1}{2} \right)^i}_{\frac{x}{(1-x)^2} = \sum_{i=1}^{\infty} ix^i} = 2 \frac{\frac{1}{2}}{\left(1 - \frac{1}{2}\right)^2} = 4 \stackrel{\text{def}}{=} \overset{\text{sim}}{\Delta}_{\text{r-ifs}}(\overset{\text{sim}}{\Delta}(\text{IFS})_{\sqrt{2}})(n) .
\end{aligned}$$

Herleitung 2.6: Die ifs-Restschranke  $\overset{\text{sim}}{\Delta}_{\text{r-ifs}}(\overset{\text{sim}}{\Delta}(\text{IFS})_{\sqrt{2}})(n)$

### 2.5.2 Einfache numerische Berechnung zur Basis 10

**Lemma 2.76** (Allgemeine Fehlerabschätzung der Berechnung durch  $\overset{\text{sim}}{\text{INE}}$  zur Basis 10). *Der absolute Fehler der Spigot-Berechnung von  $\sqrt{2}$  durch  $\overset{\text{sim}}{\text{INE}}$  erfüllt die Ungleichung*

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}} \left( \overset{\text{sim}}{\text{INE}}_{\sqrt{2}}, 10, n, i \right) \leq \frac{4}{10^i} + \left( \frac{1}{2} \right)^{n-1} . \quad (2.111)$$

*Beweis.* Satz 2.62 wird angewandt. In (2.100) ist  $\text{INE}$  mit  $\overset{\text{sim}}{\text{INE}}_{\sqrt{2}}$ ,  $\overset{\square}{\Delta}_{\text{r-ifs}}(\text{IFS})(n)$  laut Lemma 2.75 und  $\overset{\square}{\Delta}_{\text{r-ine}}(\text{INE})(n)$  laut Lemma 2.74 zu instanzieren. ■

**Satz 2.77** (Charakterisierung der Berechnung durch  $\overset{\text{sim}}{\text{INE}}$  zur Basis 10). *Bei der Spigot-Berechnung von  $\sqrt{2}$  durch  $\overset{\text{sim}}{\text{INE}}$  ist die obere Schranke  $\frac{1}{10^l}$  für den absoluten Fehler mit dem Approximationsgrad  $n = \lceil 2 + l \times \log_2 10 \rceil$  und der Iterationszahl  $i = l + 1$  sichergestellt.*

Im Ausdruck von Definition 2.63 ist die obere Schranke  $\frac{1}{10^l}$  für den absoluten Fehler durch die charakteristische Funktion

$$\overset{\text{sim}}{\chi}_{\text{spigot}} \left( \overset{\text{sim}}{\text{INE}}_{\sqrt{2}}, 10 \right) (l) \stackrel{\text{def}}{=} (\lceil 2 + l \times \log_2 10 \rceil, l + 1) \quad (2.112)$$

sichergestellt. Für die praktische Fehlerabschätzung wird  $\log_2 10$  durch eine größere rationale Zahl (z. B. 4) ersetzt.

*Beweis.* Lemma 2.76 wird angewandt. Die obere Schranke  $\frac{1}{10^l}$  wird mit Hilfe der Gleichung

$$\frac{1}{10^l} = \frac{1}{2 \times 10^l} + \frac{1}{2 \times 10^l}$$

zerlegt. Dann wird die Ungleichung  $\left(\frac{1}{2}\right)^{n-1} \leq \frac{1}{2 \times 10^i}$  nach  $n$  und die Ungleichung  $\frac{4}{10^i} \leq \frac{1}{2 \times 10^i}$  nach  $i$  aufgelöst. ■

**Satz 2.78** (Abschätzung der größten benötigten Zahl in der Berechnung mit IFS $_{\sqrt{2}}^{\text{sim}}$  zur Basis 10).

Die größte benötigte Zahl der Spigot-Berechnung von  $\sqrt{2}$  durch  $\text{INE}_{\sqrt{2}}^{\text{sim}}$  mit dem Approximationsgrad  $n > 2$  ist durch  $80 \times n - 40$  begrenzt.

Im Ausdruck von Definition 2.63 ist die Abschätzung der größten Zahl beim Approximationsgrad  $n$  durch die Schrankenfunktion

$$\Delta_{\text{spigot}}^{\text{sim}} \left( \text{IFS}_{\sqrt{2}}^{\text{sim}}, 10 \right) (n) \stackrel{\text{def}}{=} 80 \times n - 40 \quad (2.113)$$

charakterisiert.

*Beweis.* Die Teilprozesse der Berechnung sind bezüglich der größten benötigten Zahlen zu betrachten.

1. Durch Anwendung von Lemma 2.42 mit der Restschranke aus Lemma 2.75 auf das Faktorsystem (2.107) sind die Normierungskoeffizienten der Extraktion durch  $80 \times n - 40$  begrenzt.
2. Durch Lemma 2.43 und Lemma 2.73 werden die Normierungskoeffizienten der Ergebnismormierung durch 50 begrenzt.
3. Die Inputnormierung erzeugt keine neuen Normierungskoeffizienten, weil  $\text{INE}_{\sqrt{2}}^{\text{sim}}$  schon normiert ist.

Schließlich sind die benötigten Zahlen der Berechnung für  $n > 2$  durch  $80 \times n - 40$  begrenzt. ■

Wir betrachten nun ein numerisches Beispiel.

*Beispiel 2.79* (Berechnung von  $\sqrt{2}$  mit 62 dezimalen Nachkommaziffern mit der einfachen Spigot-Berechnung). Für die Sicherstellung der Fehlerschranke  $\frac{1}{10^{63}}$  sind nach Satz 2.77 der Approximationsgrad  $n = 214$  und die Iterationszahl  $i = 64$  hinreichend. Die Spigot-Approximation

$\text{APN}_{\text{ine}}^{\text{spigot}} \left( \text{INE}_{\sqrt{2}}^{\text{sim}}, 10, 214, 64 \right)$  lautet

1.4142135623730950488016887242096980785696718753769480731766797378 .

Nach Lemma 2.64 für  $l = 62$  ist

$$1. \underbrace{41421356237309504880168872420969807856967187537694807317667973}_{62 \text{ Nachkommaziffern}} 78 \quad (2.114)$$

die Darstellung von  $\sqrt{2}$  zur Basis 10 mit 62 Ziffern nach dem Punkt, weil  $c_{63} = 7$  mit  $10 - 1 = 9$  nicht übereinstimmt. Nach Satz 2.78 sind die Zahlenwerte der Berechnung durch  $17080 = 80 \times 214 - 40$  begrenzt. Für die Ausführung der Berechnung wird eine Implementation des Spigot-Algorithmus in C benutzt, wobei die ganzen Zahlen standardmäßig mit 32 Bits repräsentiert sind. Weil  $17080 < 2^{32} = 4294967296$  ist, ist das Ergebnis der maschinellen Ausführung zuverlässig.

Bei der Überwachung im C-Programm tritt die tatsächlich größte Zahl der Berechnung 15025 als ein Normierungskoeffizient der Position 207 auf. Dieser Wert liegt sehr nahe bei dem theoretisch maximalen  $80 \times (207 + 1) - 40 = 16600$ . □

## 2.5.3 Berechnung mit Potenzüberbrückung zur Basis 10

**Lemma 2.80** (Charakterisierung der Berechnung durch  $\overset{\text{sim}}{\text{INE}}$  zur Basis  $10^m$ ). *Im Ausdruck der Definition 2.63 ist die Sicherstellung der oberen Schranke  $\frac{1}{10^l}$  durch die Funktion*

$$\overset{\text{sim}}{\chi}_{\text{spigot}} \left( \overset{\text{sim}}{\text{INE}}, 10^m \right) (l) \stackrel{\text{def}}{=} \left( \lceil 2 + l \times \log_2 10 \rceil, \left\lceil \frac{l+1}{m} \right\rceil \right) \quad (2.115)$$

charakterisiert.

*Beweis.* Analog zum Beweis von Satz 2.77. ■

**Lemma 2.81** (Abschätzung der größten benötigten Zahl in der Berechnung mit  $\overset{\text{sim}}{\text{IFS}}$  zur Basis  $10^m$ ). *Im Ausdruck von Definition 2.63 ist die Abschätzung der größten benötigten Zahl beim Approximationsgrad  $n$  durch die Schrankenfunktion*

$$\overset{\text{sim}}{\Delta}_{\text{spigot}} \overset{\text{sim}}{(\text{IFS}, 10^m)}(n) \stackrel{\text{def}}{=} (8n - 4) 10^m \quad (2.116)$$

charakterisiert.

*Beweis.* Analog zum Beweis von Satz 2.78. ■

Wir betrachten nun ein numerisches Beispiel der Herleitung von optimaler Potenzüberbrückung.

*Beispiel 2.82* (Berechnung von  $\sqrt{2}$  mit 62 dezimalen Nachkommaziffern mit der Potenzüberbrückung). Die Sicherstellung der Fehlerschranke  $\frac{1}{10^{63}}$  ist nach Lemma 2.80 mit dem Approximationsgrad  $n = 214$  zu erreichen, wobei  $m$  noch zu variieren ist. Lemma 2.81 wird nun angewandt. Bei einer zugrunde liegenden Zahlendarstellung von 32 Bits ist der optimale Wert von  $m$  der größte Wert, der die Ungleichung

$$(8 \times 214 - 4) \times 10^m < 2^{32}$$

noch erfüllt. Daraus wird  $m = 6$  berechnet. Nach Lemma 2.80 ist  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}} \left( \overset{\text{sim}}{\text{INE}}, 10^6, 214, 11 \right)$  eine Approximation von  $\sqrt{2}$  mit einer Abweichung kleiner als  $\frac{1}{10^{63}}$ .  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}} \left( \overset{\text{sim}}{\text{INE}}, 10^6, 214, 11 \right)$  lässt sich durch den Spigot-Algorithmus berechnen und hat zur Basis  $10^6$  die folgende Notation:

$$1.[414213][562373][\underbrace{95048}][801688][724209][698078][569671][875376][948073][176679][737988] . \quad (2.117)$$

Die Potenzrückführung laut Algorithmus 2.7 erzeugt das folgende Ergebnis:

$$1.[414213][562373][\underbrace{095048}][801688][724209][698078][569671][875376][948073][176679][737988] , \quad (2.118)$$

wobei  $\underbrace{\quad}$  die von (2.117) abweichende Stelle hervorhebt. Für die Sicherstellung von 62 dezimalen Nachkommaziffern führt (2.118) laut Lemma 2.64 zum gleichen numerischen Ergebnis wie (2.114). Man siehe Bemerkung 2.71 für die durch die Überbrückung erzielte Effizienzverbesserung □

## 2.6 Zusammenfassung

In diesem Kapitel sind die formalen Grundlagen für den korrekten Einsatz des Spigot-Algorithmus geschaffen worden, deren Anwendung durch die detaillierte Ausführung des komplexen Beispiels in Abschnitt 2.5 illustriert ist. Mit ihnen lassen sich demnach die Bewerkstelligung einer gewünschten Genauigkeit, die Abschätzung der größten benötigten Zahl und die Bestimmung der optimalen Potenzüberbrückung effektiv durchführen.

Das Höchstmaß an Zuverlässigkeit von maschinell ausgeführten Berechnungen mit dem Spigot-Algorithmus ist insbesondere dadurch begründet, dass zur Berechnung eines numerischen Ergebnisses  $\overset{\text{spigot}}{\text{APN}}_{\text{ine}}(INE, g, n, i)$  nur elementare arithmetische Operationen benötigt werden, die von jedem Computer exakt ohne Rundung ausführbar sind.

Fassen wir nun die zuverlässige numerische Berechnung einer Reihe mit dem Spigot-Algorithmus in einem Schema zusammen.

**Schema 2.83** (Zuverlässige numerische Berechnung einer Reihe mit dem Spigot-Algorithmus). Es sei eine reelle Zahl  $r$  mit einer ine-Darstellung  $INE$  gegeben. Das Faktorsystem von  $INE$  ist  $IFS$ . Die Spigot-Berechnung von  $r$  setzt sich aus den folgenden Schritten zusammen:

1. Konstruiere eine möglichst einfache und passende ine-Restschranke  $f(n)$  für  $INE$ , z. B. mit Lemma 2.60.
2. Konstruiere eine möglichst einfache und passende Spigot-Schranke  $g(m, n)$  für  $IFS$  durch geeignete Aufwertung von  $\underset{s\text{-ifs}}{\Delta}^{\min}(IFS)(m, n)$ .
3. Wähle eine Zielbasis  $g$ .
4. Setze  $f(n)$  und  $g(m, n)$  in (2.100) ein und untersuche die funktionale Beziehung zwischen den Parametern des Spigot-Algorithmus und der entsprechenden erreichten Genauigkeit. Leite eine charakteristische Funktion  $c(l) \mid l \mapsto (n, i)$  (siehe Bemerkung 2.63) gegebenenfalls her.
5. Bestimme die Parameter für die gewünschte Genauigkeit anhand des Ergebnisses im vorigen Schritt.
6. Führe den Spigot-Algorithmus mit den berechneten Parametern aus.

Vor der Ausführung des Spigot Algorithmus sind optional auszuführen:

- Setze  $g(m, n)$  in Lemma 2.43 ein, um die größte benötigte Zahl zu untersuchen.
- Untersuche den Vorteil einer Potenzüberbrückung anhand der größten benötigten Zahl und des verfügbaren maschinellen Zahlensystem.

□

Eine INE kann durch (2.77a) bzw. (2.77b) aus einer Reihe gewonnen werden. Für die Herleitung von Reihen werden die Techniken der Reihenentwicklung und Reihenbeschleunigung im nächsten Teil der Dissertation systematisiert.

## Teil II

# Techniken der Herleitung und Beschleunigung von Reihen

## Kapitel 3

# Reihenentwicklung durch charakteristische Funktionalgleichung

Die Darstellung des Sachverhalts in diesem Kapitel mit mathematischer Strenge bezieht Begriffe und Gesetzmäßigkeiten aus Anhang A ein, der eine systematische Zusammenfassung von gut bekannten Kenntnissen aus der reellen Analysis ist und vor allem als eine Basis zum Nachschlagen dienen soll. Ein schnelles Durchschauen von Anhang A vor der Lektüre dieses Kapitels kann das Verständnis fördern.

Charakterisieren wir zunächst die Vorgehensweise, eine Reihe für die Berechnung eines numerischen Wertes herzuleiten. Ein zu berechnender reeller numerischer Wert wird im Allgemeinen als der Wert einer speziellen Funktion — in diesem Kontext als *Trägerfunktion* bezeichnet — bei einem rationalen Argument aufgefasst. Eine Reihe für diesen numerischen Wert wird dadurch hergeleitet, dass die Trägerfunktion in eine (Potenz)Reihe entwickelt wird. Die Kunst liegt dabei in der Wahl der Trägerfunktion — z. B. durch Einsatz geeigneter Substitution — sowie in der Technik der Reihenentwicklung.

Zur Entwicklung der Funktionen, deren mehrfache Ableitungen durch die Regeln der Differenziation effektiv berechenbar sind, liefert die *Taylor'sche Reihe* im Satz A.36 einen effektiven und bequemen Ansatz.

Wir wollen in diesem Abschnitt aber eine andere Vorgehensweise motivieren, die auf die klassische Idee der *Methode der unbestimmten Koeffizienten* zurückgeht. Dementsprechend werden aus der Trägerfunktion *charakteristische Funktionalgleichungen*<sup>1</sup> abgeleitet, die dann als Grundlage dazu dienen, die Koeffizienten der Potenzreihe für die Funktion effektiv zu berechnen, wobei die Regeln im Abschnitt A.5 anzuwenden sind. Der Korrektheit der Vorgehensweise liegt Satz A.28 zugrunde. In der vorliegenden Dissertation wird sie als *Methode des Koeffizientenvergleiches in charakteristischen Funktionalgleichungen* bezeichnet. Die Methode ist allgemeiner als der Ansatz der Taylorschen Reihe und deshalb auch für Funktionen anwendbar, deren Ableitungen nicht offensichtlich zu berechnen sind. Ihre Anwendung ist im Allgemeinen ein schöpferischer Prozess und soll in diesem Kapitel durch Beispiele motiviert werden. Die Kunst liegt vor allem bei der „Entdeckung“ von treffenden charakteristischen Funktionalgleichungen, dann eventuell bei der „Überbrückung“ durch passende Substitutionen. Die Einbeziehung von speziellen Punkten der Funktion kann auch eine Rolle spielen.

---

<sup>1</sup>Hier auch als Oberbegriff für Differenzial- und Integralgleichungen gemeint

Eine charakteristische Funktionalgleichung ist nicht nur für die Entwicklung der Trägerfunktion interessant. Wir werden auch noch sehen, dass mit deren Hilfe die Funktion in eine günstigere Berechnungsform überführt werden kann, die für die numerische Berechnung der Funktion entscheidend ist. Dies betrifft insbesondere die Wurzelberechnung in Kapitel 7 und die Logarithmenberechnung in Kapitel 8.

Mit diesem Kapitel werden keine neuen Reihenentwicklungen erzeugt. Aber die Ideen und die Methoden, die durch die Beispiele motiviert werden, können Denkanstöße für die Entwicklung weiterer Funktionen geben.

### 3.1 Exponentialreihe

Die charakteristische Funktionalgleichung für alle Exponentialfunktionen lautet

$$E(x_1 + x_2) = E(x_1) \cdot E(x_2) , \quad (3.1)$$

wobei  $E(x)$  als Platzhalter für eine Funktion steht. Die Funktion  $\exp(x) \stackrel{\text{def}}{=} e^x$  erfüllt zusätzlich die Gleichung

$$\exp'(0) = 1 , \quad (3.2)$$

die  $\exp(x)$  gegenüber allen anderen Exponentialfunktionen auszeichnet. Man sieht auch, dass Gleichung (3.2) *der Schlüssel zur Zahl e* ist.

In diesem Abschnitt sehen wir uns eine Herleitung der Potenzreihe für  $\exp(x)$  an, die von (3.1) und (3.2) ausgeht.

**Satz 3.1** (Die Exponentialreihe). [Vergl. Wal92, S. 145]. *Es gilt die Reihenentwicklung*

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \dots \quad (3.3)$$

für alle  $x \in \mathbb{R}$ .

*Beweis.* Um die Problematik der Reihenentwicklung mit mathematischer Strenge zu demonstrieren, gehen wir im Folgenden sowohl auf die Herleitung als auch auf den Beweis der Gültigkeit ein. Die Identität von  $\exp(x)$  durch (3.1) und (3.2) wird dabei vorausgesetzt.

**Herleitung der Reihenentwicklung aus der charakteristischen Funktionalgleichung.**

Setzen wir zunächst in (3.1)  $x_2 = 0$ , um

$$\forall x . E(x) = E(x) \cdot E(0) \quad (3.4)$$

zu erhalten. Unter der Annahme, dass  $E(x)$  nicht überall gleich 0 ist, wird

$$E(0) = 1 \quad (3.5)$$

impliziert.

Wir versuchen nun, eine Differenzialgleichung für  $\exp(x)$  zu finden, um eine bequeme Anwendung der Methode des Koeffizientenvergleiches zu ermöglichen. Dazu wird die Transformation

$$\frac{\underbrace{E(x + \Delta x) - E(x)}_{(3.1)}}{\Delta x} = \frac{E(x) \cdot E(\Delta x) - E(x)}{\Delta x} = E(x) \cdot \frac{\underbrace{E(\Delta x) - 1}_{(3.5)}}{\Delta x} = E(x) \cdot \frac{E(\Delta x) - E(0)}{\Delta x}$$

durchgeführt, woraus durch den Grenzübergang  $\Delta x \rightarrow 0$  die Differenzialgleichung

$$E'(x) = E(x) \cdot E'(0) \quad (3.6)$$

hergeleitet wird. Als ein spezieller Fall von Exponentialfunktionen erfüllt auch  $\exp(x)$  (3.6).

Gleichung (3.6) wird nun als Grundgleichung für den Koeffizientenvergleich benutzt. Als Ansatz für  $\exp(x)$  wird dabei die allgemeine Form (A.27a) herangezogen. Zugleich wird für die Berechnung der Koeffizienten von  $\exp'(x)$  Satz A.32 ( $x_0$  ist mit 0 zu instanzieren) angewandt. Durch diesen Koeffizientenvergleich erhalten wir als Ergebnis die Rekursionsformel

$$a_{n+1} = \frac{a_n \cdot \exp'(0)}{(n+1)} \quad n \geq 0, \quad (3.7)$$

außerdem wird aus (3.5) die Gleichung

$$a_0 = \exp(0) = 1 \quad (3.8)$$

impliziert. Aus (3.8) und (3.7) ergibt sich die Gleichung

$$a_n = \frac{(\exp'(0))^n}{n!}, \quad (3.9)$$

die unter Beachtung von (3.2) zu

$$a_n = \frac{1}{n!} \quad (3.10)$$

wird. In Kombination mit (A.27a) entspricht (3.10) der Entwicklung (3.3).

**Beweis der Gültigkeit.** Dass die Reihe (3.3) für alle  $x \in \mathbb{R}$  konvergiert, kann durch Satz A.27 und Satz A.21 bewiesen werden. Dass die Reihe (3.1) erfüllt, wird durch das Cauchy-Produkt von Reihen (siehe (A.23)) und die Binomialformel

$$(x+y)^n = \sum_{i=0}^n \binom{n}{i} x^i y^{n-i} \quad (3.11)$$

verifiziert. Dass die Reihe (3.2) erfüllt, ist durch Anwendung von Satz A.32 ( $x_0$  ist mit 0 zu instanzieren) offensichtlich. ■

## 3.2 Logarithmische Reihe

Die charakteristische Funktionalgleichung für alle Logarithmusfunktionen lautet

$$L(x_1 \cdot x_2) = L(x_1) + L(x_2), \quad (3.12)$$

wobei  $L(x)$  als Platzhalter für eine Funktion steht. Der natürliche Logarithmus  $\ln(x)$  ist dabei durch die Gleichung

$$\ln'(1) = 1 \quad (3.13)$$

ausgezeichnet.

In diesem Abschnitt leiten wir von (3.12) und (3.13) ausgehend die Potenzreihe für  $\ln(1+x)$  her und darauf aufbauend auch einige spezielle Reihen.



**Satz 3.2** (Mercatorsche Reihe). *Es gilt die Reihenentwicklung*

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - + \dots = \sum_{i=1}^{\infty} \frac{x^i}{i} (-1)^{i+1} \quad |x| < 1 . \quad (3.14)$$

*Herleitung.* Setzen wir zunächst in (3.12)  $x_2 = 1$ , um

$$\forall x . L(x) = L(x) + L(1) \quad (3.15)$$

zu erhalten. Aus (3.15) folgt unmittelbar

$$L(1) = 0 . \quad (3.16)$$

Jetzt wird versucht nun, eine Differenzialgleichung für  $\ln(x)$  zu finden, um eine bequeme Anwendung der Methode des Koeffizientenvergleiches zu ermöglichen. Dazu führen wir die Transformation

$$\begin{aligned} \frac{L(\underbrace{x + \Delta x}) - L(x)}{\Delta x} &= \frac{L\left(x \cdot \left(1 + \frac{\Delta x}{x}\right)\right) - L(x)}{\Delta x} = \frac{\underbrace{L\left(x \cdot \left(1 + \frac{\Delta x}{x}\right)\right)}_{(3.12)} - L(x)}{\Delta x} \\ &= \frac{(L(x) + L(1 + \frac{\Delta x}{x})) - L(x)}{\Delta x} = \frac{L(1 + \frac{\Delta x}{x})}{\Delta x} \stackrel{(3.16)}{=} \frac{1}{x} \cdot \frac{L(1 + \frac{\Delta x}{x})}{\frac{\Delta x}{x}} \stackrel{(3.16)}{=} \frac{1}{x} \cdot \frac{L(1 + \frac{\Delta x}{x}) - L(1)}{\frac{\Delta x}{x}} \end{aligned}$$

durch, woraus durch den Grenzübergang  $\Delta x \rightarrow 0$  die Differenzialgleichung

$$L'(x) = L'(1) \cdot \frac{1}{x} \quad (3.17)$$

hergeleitet wird. Unter Beachtung von (3.13) ergibt sich aus (3.17) die Gleichung

$$\ln'(x) = \frac{1}{x} . \quad (3.18)$$

Der Koeffizientenvergleich, bei dem  $\ln(x)$  die Form (A.27a) einnimmt und (3.18) als Grundlage benutzt wird, liefert keine Lösung. Um über dieses Problem hinwegzukommen, wird die Substitution  $x \xrightarrow{\text{sub}} (x+1)$  zu Hilfe genommen, wodurch sich

$$\ln'(1+x) = \frac{1}{1+x} \quad (3.19)$$

aus (3.18) ergibt. Die rechte Seite von (3.19) als eine Funktion  $q(x)$  von  $x$  wird durch den Koeffizientenvergleich, bei dem  $q(x)$  die Form (A.27a) einnimmt und die „charakteristische Gleichung“  $q(x)(1+x) = 1$  als Grundgleichung benutzt sowie  $q(0) = 1$  als ein spezieller Punkt einbezogen wird, zu der Reihe

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + x^4 - + \dots = \sum_{n=0}^{\infty} (-1)^n x^n . \quad (3.20)$$

entwickelt. Aus (3.19) und (3.20) wird

$$\ln'(1+x) = \sum_{n=0}^{\infty} (-1)^n x^n . \quad (3.21)$$

impliziert. Durch den Koeffizientenvergleich, bei dem  $\ln(1+x)$  die Form (A.27a) einnimmt und (3.21) als Grundgleichung benutzt sowie  $\ln(1+0) = 0$  (vergl.(3.16)) als ein spezieller Punkt einbezogen wird, wird die Reihe(3.14) hergeleitet<sup>2</sup>.

Der Konvergenzradius der Reihe ist durch Anwendung von Satz A.27 leicht zu berechnen, es gilt nämlich

$$\limsup_{i \rightarrow \infty} \sqrt[i]{|a_i|} = \limsup_{i \rightarrow \infty} \sqrt[i]{1} = 1$$

für (3.20).

Für die Gültigkeit der Entwicklung ist im Wesentlichen zu zeigen, dass (3.12) aus (3.16) und (3.17) folgt. Wir gehen aber nicht auf Details ein. ■

**Lemma 3.3** (Harmonische Reihe mit alternierenden Vorzeichen). *Es gilt die Gleichung*

$$\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} . \quad (3.22)$$

*Beweis.* Für  $x = 1$  entspricht die Reihe auf der rechten Reihe von (3.22) der Reihe auf der rechten Reihe von (3.14). Die Konvergenz wird durch Satz A.18 festgestellt und der Grenzwert wird nach Satz A.29 berechnet. ■

**Lemma 3.4** (Gregory-Halley-Reihe). [Vergl. Wal92, S. 148]. *Es gilt die Reihenentwicklung*

$$\ln \frac{1+x}{1-x} = 2 \left[ x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots \right] = 2 \sum_{k=0}^{\infty} \frac{x^{2k+1}}{2k+1} \quad |x| < 1 . \quad (3.23)$$

Diese Entwicklung war schon seit Gregory (1668) und Halley (1695) bekannt.

*Beweis.* Es gilt  $\ln \frac{1+x}{1-x} = \ln(1+x) - \ln(1-x)$ . Sowohl für  $\ln(1+x)$  als auch für  $\ln(1-x)$  wird die Entwicklung (3.14) angewandt. Der Rest ist offensichtlich. ■

**Lemma 3.5** (Grundreihe für Berechnung natürlicher Logarithmen). *Es gilt die Gleichung*

$$\ln 2 = \sum_{k=0}^{\infty} \frac{2}{2k+1} \frac{1}{3^{2k+1}} . \quad (3.24)$$

*Beweis.* In (3.23) wird  $x = \frac{1}{3}$  gesetzt. ■

Weitere Anregungen für die Transformation der logarithmischen Reihe findet man in [Sto97, S. 96,97].

<sup>2</sup> Eine leichte Abänderung: Auf (3.21) wird die gliedweise Integration in Bezug auf die Variable  $x$  mit der Untergrenze 0 und Obergrenze  $x$  nach Satz A.34 angewandt.

### 3.3 Die Funktion arctan und die Gregory-Leibniz-Reihe für $\pi$

Die Potenzreihe für  $\arctan x$  wird anhand der charakteristischen Gleichung ihrer Umkehrfunktion  $\tan x$  hergeleitet.

**Lemma 3.6** (Charakteristische Funktionalgleichung und Stützpunkte für die Funktion  $\tan x$ ).  
Die Funktion  $\tan x$  erfüllt die charakteristische Funktionalgleichung

$$\tan(x + y) = \frac{\tan x + \tan y}{1 - \tan x \tan y} , \quad (3.25)$$

und außerdem noch zwei weitere Gleichungen

$$\tan 0 = 0 \quad (3.26)$$

und

$$\tan' 0 = 1 . \quad (3.27)$$

Die Gültigkeit von Lemma 3.6 wird hier angenommen.

**Lemma 3.7** (Differenzialeigenschaft von  $\tan x$ ). Es gilt die Gleichung

$$\tan' x = 1 + \tan^2 x . \quad (3.28)$$

*Beweis.* Bilden wir den Differenzialquotienten für  $\tan x$  mit Hilfe der charakteristischen Gleichung (3.25) wie folgt

$$\begin{aligned} \frac{\tan(x + \Delta x) - \tan x}{\Delta x} &= \frac{\frac{\tan x + \tan \Delta x}{1 - \tan x \tan \Delta x} - \tan x}{\Delta x} = \frac{1 + \tan^2 x}{1 - \tan x \tan \Delta x} \cdot \frac{\tan \Delta x}{\Delta x} \\ &\stackrel{(3.26)}{=} \frac{1 + \tan^2 x}{1 - \tan x \tan \Delta x} \cdot \frac{\tan \Delta x - \tan 0}{\Delta x} , \end{aligned}$$

woraus sich durch den Grenzübergang  $\Delta x \rightarrow 0$

$$\tan' x = 1 + \tan^2 x \cdot \underbrace{\tan' 0}_{(3.27)} = 1 + \tan^2 x$$

ergibt. Nach Abtrennung der „Brücke“ erhalten wir (3.28). ■

**Satz 3.8** (Leibniz-Reihe für  $\arctan x$ ). Es gilt die Reihenentwicklung

$$\arctan x = \sum_{i=0}^{\infty} (-1)^i \cdot \frac{x^{2i+1}}{2i+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11} + \dots \quad \text{für } |x| < 1 . \quad (3.29)$$

*Zur Herleitung.* Die Differenziationsregel der Umkehrfunktion wird für das Paar

$$y = \tan x \quad \text{und} \quad x = \arctan y$$

angewandt, es gilt

$$\arctan' y = \frac{1}{1 + \tan^2 x} = \frac{1}{1 + y^2}$$

oder nach Abtrennung der „Brücke“

$$\arctan' y = \frac{1}{1+y^2} . \quad (3.30a)$$

Durch Variablenumbenennung ergibt sich

$$\arctan' x = \frac{1}{1+x^2} \quad (3.30b)$$

aus (3.30a), außerdem lässt sich

$$\frac{1}{1+x^2} = 1 - x^2 + x^4 - x^6 + x^8 - + \dots \quad \text{für } |x| < 1 \quad (3.31)$$

aus (3.20) herleiten. Der Schritt von (3.30b), (3.31) und (3.26) zu (3.29) erfolgt ähnlich wie bei der Herleitung von (3.14) aus (3.16) und (3.21). ■

**Satz 3.9** (Gregory-Leibniz-Reihe für  $\pi$ ). *Es gilt die Reihenentwicklung*

$$\frac{\pi}{4} = \sum_{i=0}^{\infty} \frac{(-1)^i}{2i+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + - \dots \quad (3.32)$$

*Beweis.* Analog zum Beweis von (3.22). Die Gleichung

$$\frac{\pi}{4} = \arctan 1 \quad (3.33)$$

wird hier vorausgesetzt. Auf (3.29) wird Satz A.29 angewandt. ■

### 3.4 Binomialreihe

Die charakteristische Funktionalgleichung für alle Funktionen der Form  $x^\alpha$  ( $x$  ist die Variable und  $\alpha$  ist ein Parameter) lautet

$$P(x_1 \cdot x_2) = P(x_1) \cdot P(x_2) , \quad (3.34)$$

wobei  $P(x)$  als Platzhalter für eine Funktion steht. Der Exponent  $\alpha$  ist dabei durch

$$\alpha = P'(1) \quad (3.35)$$

ausgezeichnet.

Dieser Abschnitt beschäftigt sich mit der Herleitung der Potenzreihe für  $(1+x)^\alpha$  von (3.34) und (3.35) ausgehend. Diese Reihe bildet die Grundlage für die Wurzelberechnung in Kapitel 7.

**Satz 3.10** (Binomialreihe). *[Vergl. Wal92, S. 103]. Es gilt die Reihenentwicklung*

$$B(x, \alpha) \stackrel{\text{def}}{=} \sum_{n=0}^{\infty} \binom{\alpha}{n} x^n = 1 + \alpha x + \frac{\alpha(\alpha-1)}{1 \cdot 2} x^2 + \frac{\alpha(\alpha-1)(\alpha-2)}{1 \cdot 2 \cdot 3} x^3 + \dots = (1+x)^\alpha \quad |x| < 1 . \quad (3.36)$$

*Herleitung.* In (3.34) wird  $x_2 = 1$  gesetzt, woraus sich

$$\forall x . P(x) = P(x) \cdot P(1) \quad (3.37)$$

ergibt. Unter der Annahme, dass  $P(x)$  nicht überall gleich 0 ist, wird

$$P(1) = 1 \quad (3.38)$$

impliziert.

Versuchen wir nun eine Differenzialgleichung für  $P(x)$  zu finden, um eine bequeme Anwendung des Koeffizientenvergleiches zu ermöglichen. Dazu wird die Transformation

$$\begin{aligned} \frac{P(\underbrace{x + \Delta x}) - P(x)}{\Delta x} &= \frac{\underbrace{P\left(x \cdot \left(1 + \frac{\Delta x}{x}\right)\right) - P(x)}_{(3.34)}}{\Delta x} = \frac{P(x) \cdot P\left(1 + \frac{\Delta x}{x}\right) - P(x)}{\Delta x} \\ &= \frac{P(x) \cdot (P\left(1 + \frac{\Delta x}{x}\right) - 1)}{\Delta x} = \frac{P(x)}{x} \cdot \frac{(P\left(1 + \frac{\Delta x}{x}\right) - \underbrace{1}_{(3.38)})}{\frac{\Delta x}{x}} = \frac{P(x)}{x} \cdot \frac{(P\left(1 + \frac{\Delta x}{x}\right) - P(1))}{\frac{\Delta x}{x}} \end{aligned}$$

durchgeführt, woraus durch den Grenzübergang  $\Delta x \rightarrow 0$  die Differenzialgleichung

$$P'(x) = \frac{P(x)}{x} \cdot P'(1) \quad (3.39)$$

hergeleitet wird.

Der Koeffizientenvergleich, bei dem  $P(x)$  die Form (A.27a) einnimmt und (3.39) als Grundgleichung benutzt wird, führt dazu, dass sämtliche Koeffizienten gleich 0 sind. Um über dieses Problem hinwegzukommen, wird ähnlich wie bei der Herleitung der Mercatorschen Reihe verfahren. In (3.39) wird die Substitution  $x \xrightarrow{\text{sub}} x + 1$  durchgeführt, woraus sich die Gleichung

$$P'(1 + x) = \frac{P(1 + x)}{1 + x} \cdot P'(1)$$

ergibt, die unter Beachtung von (3.35) in

$$P'(1 + x) \cdot (1 + x) = \alpha \cdot P(1 + x) \quad (3.40)$$

transformiert werden kann. Durch den Koeffizientenvergleich, bei dem  $P(1 + x)$  die Form (A.27a) einnimmt und (3.40) als Grundgleichung benutzt sowie (3.38) als ein spezieller Punkt einbezogen wird, kommen wir zu (3.36). Der Koeffizientenvergleich wird im Folgenden näher betrachtet. Aus (3.38) folgt

$$a_0 = 1 . \quad (3.41)$$

Der Koeffizientenvergleich mit (3.40) als Grundgleichung und unter Anwendung von Satz A.32 ( $x_0$  ist mit 0 zu instanzieren) liefert die Gleichungen

$$\alpha \cdot a_0 = a_1 \quad \Leftrightarrow \quad a_1 = \alpha \cdot a_0 \quad (3.42)$$

und

$$\alpha \cdot a_i = (i + 1) \cdot a_{i+1} + i \cdot a_i \quad \Leftrightarrow \quad a_{i+1} = \frac{(\alpha - i) \cdot a_i}{i + 1} \quad \text{für } i \geq 1 . \quad (3.43)$$

Aus (3.41), (3.42) und (3.43) ergeben sich die Gleichungen

$$a_0 = 1, \quad a_1 = \alpha \quad \text{und} \quad a_i = \frac{\alpha \cdot (\alpha - 1) \dots (\alpha - i + 1)}{1 \cdot 2 \dots i} \quad \text{für } i \geq 2, \quad (3.44)$$

die in Kombination mit (A.27a) der Reihe (3.36) entsprechen.

Der Konvergenzradius der Reihe (3.36) lässt sich durch Anwendung von Satz A.21 bestimmen. Der Gültigkeitsbeweis geht auf

$$\lim_{x \rightarrow 0} \frac{(1+x)^\alpha - 1}{x} = \alpha \quad (3.45)$$

und (A.23) zurück. Wir gehen aber nicht auf Details ein. ■

### 3.5 Reihen mit durch Bernoullische Zahlen dargestellten Koeffizienten

Bisher haben wir nur Funktionen betrachtet, bei deren Entwicklung sich die Koeffizienten durch gebrochen rationale Funktionen vom Index bequem darstellen lassen. Im Allgemeinen betrifft dies nur die günstigsten Fälle. In diesem Abschnitt sehen wir Potenzreihenentwicklungen an, bei denen die Koeffizienten zwar aus dem Index effektiv rekursiv berechenbar, jedoch nicht durch gebrochen rationale Funktionen des Indexes darstellbar sind.

**Lemma 3.11** (Die Reihe für  $\frac{x}{\exp(x)-1}$  nach  $x$ ). [Vergl. Kno96, S. 185,207]. Es gilt die Entwicklung

$$\frac{x}{\exp(x) - 1} = \sum_{i=0}^{\infty} \frac{B_i x^i}{i!} \quad x \in \mathbb{R}^3, \quad (3.46)$$

wobei die Bernoullischen Zahlen der Folge  $[B_i]_{i=0}^{\infty}$  durch das Gleichungssystem

$$\left. \begin{array}{l} B_0 = 1 \\ 2 B_1 + 1 = 0 \\ 3 B_2 + 3 B_1 + 1 = 0 \\ 4 B_3 + 6 B_2 + 4 B_1 + 1 = 0 \\ \dots \\ \binom{n+1}{n} B_n + \binom{n+1}{n-1} B_{n-1} + \dots + \binom{n+1}{2} B_2 + \binom{n+1}{1} B_1 + 1 = 0 \\ \dots \end{array} \right\} \quad (3.47)$$

effektiv zu berechnen sind. Es gelten u. a.

$$B_0 = 1, \quad B_1 = \frac{-1}{2}, \quad B_2 = \frac{1}{6}, \quad B_3 = 0, \quad B_4 = \frac{-1}{30}, \dots \quad (3.48)$$

Herleitungsskizze. Aus (3.3) lässt sich

$$\frac{x}{\exp(x) - 1} = \frac{1}{1 + \frac{x}{2!} + \frac{x}{2!} + \frac{x^2}{3!}} + \dots$$

---

<sup>3</sup>Der Wert von  $\frac{x}{\exp(x)-1}$  an der Stelle  $x = 0$  wird auf 1 gesetzt, um die Gültigkeit von (3.46) für alle  $x \in \mathbb{R}$  zu gewährleisten.

für  $x \neq 0$  herleiten. Infolgedessen ist (3.46) äquivalent zu

$$\left(1 + \frac{x}{2!} + \frac{x^2}{3!} + \frac{x^3}{4!} + \dots\right) \cdot \left(B_0 + \frac{B_1}{1!}x + \frac{B_2}{2!}x^2 + \frac{B_3}{3!}x^3 \dots\right) \equiv 1. \tag{3.49}$$

Der Koeffizientenvergleich unter Benutzung von (3.49) als Grundgleichung, wobei zur Multiplikation von Reihen Satz A.30 angewandt wird, liefert zunächst das Gleichungssystem

$$\left. \begin{array}{l} B_0 = 1 \\ \frac{1}{2!} \frac{B_0}{0!} + \frac{1}{1!} \frac{B_1}{1!} = 0 \\ \frac{1}{3!} \frac{B_0}{0!} + \frac{1}{2!} \frac{B_1}{1!} + \frac{1}{1!} \frac{B_2}{2!} = 0 \\ \dots \dots \dots \\ \frac{1}{n!} \frac{B_0}{0!} + \frac{1}{(n-1)!} \frac{B_1}{1!} + \dots + \frac{1}{1!} \frac{B_{n-1}}{(n-1)!} = 0 \\ \dots \dots \dots \end{array} \right\}. \tag{3.50}$$

Das System (3.50) wird dadurch in (3.47) transformiert, dass die  $n$ -te Zeile mit  $n!$  multipliziert wird, dann die Reihenfolge der Summanden in der Zeile umgekehrt wird, und anschließend  $B_0$  in Anlehnung an die erste Zeile durch 1 ersetzt wird. ■

Wir betrachten im Folgenden noch eine Reihe, deren Koeffizienten durch Bernoullische Zahlen dargestellt sind. Die Herleitung ist in [Kno96] zu finden.

**Lemma 3.12** (Potenzreihe für  $\tan x$ ). [Vergl. Kno96, S. 185,208]. Es gilt für  $\tan x$  in einer Umgebung von 0 die Entwicklung

$$\tan x = \sum_{k=1}^{\infty} (-1)^{k-1} \cdot \frac{2^{2k}(2^{2k} - 1) B_{2k}}{(2k)!} \cdot x^{2k-1} = x + \frac{1}{3}x^3 + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \dots \tag{3.51}$$

## Kapitel 4

# Grundbausteine für die Manipulation von Folgen und Reihen

In diesem Kapitel geht es in erster Linie darum, Ideen für die Reihentransformation — möglichst als elementare Bausteine aufgefasst — anschaulich und systematisch darzustellen. Ideen bilden den Ausgangspunkt für die Lösung von Problemen. Außerdem macht eine gründliche Auseinandersetzung mit den zugrunde liegenden Ideen die Handhabung von scheinbar komplizierten Formalismen wesentlich leichter. Die voll entwickelten Formen der Ideen — in Richtung zu den formalen, teilweise maschinell ausführbaren Prozeduren der Reihentransformationen — werden erst in Kapitel 5 bzw. Kapitel 6 vorgestellt.

Es werden in diesem Kapitel zwei Grundideen betrachtet. Die erste ist die Kombination zwischen *Differenz* und *Doppelreihe*. Die zweite ist die *Wertverschiebung*. Durch die Auseinandersetzung mit den klassischen Beispielen der Reihentransformation von Euler, Kummer und Markoff sowie mit den neueren Beispielen von Gosper hat der Autor die Wichtigkeit der beiden Grundideen erkannt. Alle in dieser Dissertation betrachteten Methoden der Reihentransformation sind mehr oder weniger aus ihnen abgeleitet.

Differenz und Doppelreihe bilden eine Einheit. Die Idee der Differenz und Doppelreihe erscheint jedem wohl sehr naheliegend. Mit der Differenz lässt sich ein Term schrittweise in eine Summe zerlegen. Das Schema der Doppelreihe ermöglicht dann die Kombination der zerlegten Terme auf einer neuen Art und Weise, und zwar in der Hoffnung, eine besser konvergierende Reihe zu gewinnen.

Die Wertverschiebung ist durch Kummer und Gosper entstanden. Während mit dem Ansatz der Doppelreihe eine Transformation als eine komplette Lösung zu konstruieren ist, ermöglicht der Ansatz der Wertverschiebung eine schrittweise Annäherung. In einem Schritt der Wertverschiebung wird jeder Term einer Reihe in zwei Teile zerlegt. Jeder Term ist dabei an einer Stelle in der Reihe verbunden. Zur Ausführung der Verschiebung wird ein Teil um eine Stelle nach links verschoben und ein Teil verbleibt an der Stelle. So wird aus der Reihe von der am weitesten links stehenden Stelle ein Teil extrahiert und eine Restreihe bleibt, auf die die Wertverschiebung erneut angewandt werden kann.

Es wird in diesem Kapitel auch gezeigt, dass sich die gewöhnliche Anwendungsform der Eulerschen Transformation sowohl mit Differenz und Doppelreihe als auch mit Wertverschiebung herleiten lässt.



## 4.1 Differenz

**Definition 4.1** (Differenz von Folgen). Der Differenzoperator  $\Delta$  für Folge wird durch die Gleichung

$$\Delta \left( [a_i]_{i=i_0}^{\infty} \right) \stackrel{\text{def}}{=} [(a_i - a_{i+1})]_{i=i_0}^{\infty} \quad (4.1)$$

definiert. □

*Bemerkung 4.2* (Schreibweise für die Elemente der Differenzfolge). Das Element mit dem Index  $i$  von  $\Delta \left( [a_i]_{i=i_0}^{\infty} \right)$  wird vorwiegend mit  $\Delta a_i$  bezeichnet. Mit dieser Vereinbarung kann man schreiben

$$\Delta \left( [a_i]_{i=i_0}^{\infty} \right) = [\Delta a_i]_{i=i_0}^{\infty}, \quad (4.2)$$

wobei

$$\Delta a_i \stackrel{\text{def}}{=} a_i - a_{i+1} \quad (4.3)$$

festgelegt wird. □

**Lemma 4.3** (Einfache Eigenschaften der Differenz von Folgen). *Im Anschluss an Definition 4.1 gelten die folgenden Eigenschaften:*

1. Für  $i_0 \leq i_1 \leq i_2$  gilt

$$\sum_{i=i_1}^{i_2} \Delta a_i = a_{i_1} - a_{i_2+1} \quad (4.4)$$

oder in einer anderen Schreibweise

$$a_{i_1} = \left( \sum_{i=i_1}^{i_2} \Delta a_i \right) + a_{i_2+1}. \quad (4.5)$$

2. Gilt  $\lim_{i \rightarrow \infty} \Delta a_i = 0$ , so gilt auch

$$\sum_{i=i_1}^{\infty} \Delta a_i = a_{i_1} \quad (4.6)$$

für  $i_1 \geq i_0$

*Beweis.* Unmittelbar ersichtlich. ■

Durch geeignete Intanzierungen der Folge  $[a_i]_i$  können mit Lemma 4.3 nützliche Gleichungen hergeleitet werden, die sowohl für die geschlossene Auswertung von Summen als auch für die Transformation von Reihen von Bedeutung sind. Im Folgenden stellen wir einige solcher Gleichungen auf.

Beispiel 4.4 war dem Autor durch die Übungen in der Spezialschule für Mathematik bekannt.

*Beispiel 4.4* (Differenz und Summierung von  $\left[ \frac{1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+k)} \right]_n$ ). Für

$$a_n = \frac{1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+k)} \quad (4.7)$$

ist

$$\begin{aligned} \Delta a_n &= \frac{1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+k)} - \frac{1}{(\alpha+n+1)(\alpha+n+2)\dots(\alpha+n+k+1)} \\ &= \frac{k+1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+k+1)}, \end{aligned} \quad (4.8)$$

woraus sich in Anlehnung an (4.6)

$$\frac{1}{\alpha(\alpha+1)\dots(\alpha+k)} = \sum_{n=0}^{\infty} \frac{k+1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+k+1)} \quad (4.9)$$

ergibt. Wir wollen (4.9) in eine übersichtlichere Form bringen. Dazu wird zunächst die Substitution  $k+1 \xrightarrow{\text{sub}} p$  durchgeführt. Anschließend werden einige einfache Umformungen vollzogen. Als Ergebnis erhalten wir

$$\sum_{n=0}^{\infty} \frac{1}{(\alpha+n)(\alpha+n+1)\dots(\alpha+n+p)} = \frac{1}{p} \cdot \frac{1}{\alpha(\alpha+1)\dots(\alpha+p-1)}. \quad (4.10)$$

□

Beispiel 4.5 wird vom Autor konstruiert, um in Kombination mit Beispiel 4.14 die Transformation von  $\zeta(2)$  aus [Kno96, S. 274-275] nachzuvollziehen.

*Beispiel 4.5* (Differenz von  $\left[ \frac{\alpha(\alpha+1)\dots(\alpha+n)}{\beta(\beta+1)\dots(\beta+n+1)} \right]_n$ ). Für

$$a_n = \frac{\alpha(\alpha+1)\dots(\alpha+n)}{\beta(\beta+1)\dots(\beta+n+1)} \quad (4.11)$$

ist

$$\begin{aligned} \Delta a_n &= \frac{\alpha(\alpha+1)\dots(\alpha+n)}{\beta(\beta+1)\dots(\beta+n+1)} - \frac{\alpha(\alpha+1)\dots(\alpha+n+1)}{\beta(\beta+1)\dots(\beta+n+2)} \\ &= \frac{(\beta-\alpha+1)\alpha(\alpha+1)\dots(\alpha+n)}{\beta(\beta+1)\dots(\beta+n+2)}, \end{aligned} \quad (4.12)$$

woraus sich in Anlehnung an (4.5)

$$\frac{\alpha}{\beta(\beta+1)} = \left( \sum_{n=0}^{n_1} \frac{(\beta-\alpha+1)\alpha(\alpha+1)\dots(\alpha+n)}{\beta(\beta+1)\dots(\beta+n+2)} \right) + \frac{\alpha(\alpha+1)\dots(\alpha+n_1+1)}{\beta(\beta+1)\dots(\beta+n_1+2)} \quad (4.13)$$

ergibt.

Wir wollen nun aus (4.13) durch eine geeignete Instanziierung eine nützliche Gleichung herleiten, (4.18), die dann zur Transformation von  $\zeta(2)$  in Beispiel 4.14 benutzt wird. Für  $\alpha = 1$  wird (4.13) zu

$$\frac{1}{\beta(\beta+1)} = \left( \sum_{n=0}^{n_1} \frac{1 \cdot 2 \dots (1+n)}{(\beta+1)\dots(\beta+n+2)} \right) + \frac{1 \cdot 2 \dots (1+n_1+1)}{\beta(\beta+1)\dots(\beta+n_1+2)}. \quad (4.14)$$

Mit den Substitutionen  $n+1 \xrightarrow{\text{sub}} n$  und  $\beta \xrightarrow{\text{sub}} k$  wird (4.14) zu

$$\frac{1}{k(k+1)} = \left( \sum_{n=1}^{n_1+1} \frac{1 \cdot 2 \dots n}{(k+1)\dots(k+n+1)} \right) + \frac{1 \cdot 2 \dots (1+n_1+1)}{k(k+1)\dots(k+n_1+2)}. \quad (4.15)$$

Mit der Substitution  $n_1 + 1 \xrightarrow{\text{sub}} m$  wird (4.15) zu

$$\frac{1}{k(k+1)} = \left( \sum_{n=1}^m \frac{1 \cdot 2 \dots n}{(k+1) \dots (k+n+1)} \right) + \frac{1 \cdot 2 \dots (1+m)}{k(k+1) \dots (k+m+1)} . \quad (4.16)$$

Schließlich wird (4.16) mit der offensichtlichen Beziehung

$$\frac{1}{k} = \frac{1}{k+1} + \frac{1}{k(k+1)} = \frac{0!}{k+1} + \frac{1!}{k(k+1)} \quad (4.17)$$

kombiniert, um

$$\frac{1}{k} = \left( \sum_{n=0}^m \frac{n!}{(k+1) \dots (k+n+1)} \right) + \frac{(m+1)!}{k(k+1) \dots (k+m+1)} \quad (4.18)$$

zu erhalten. □

Der Rest des Abschnittes beschäftigt sich mit der mehrfachen Differenz.

**Definition 4.6** (Mehrfache Differenz von Folgen). Der Operator  $\Delta^{(k)}$  der mehrfachen Differenz von Folgen ist induktiv über  $k$  definiert. Im Folgenden sind die einzelnen Schritte der Induktion ausgeführt:

1. **Induktionsanfang.** Die 0-te Differenz einer Folge ist die Folge selbst, d. h.

$$\Delta^{(0)} \left( [a_i]_{i=i_0}^{\infty} \right) \stackrel{\text{def}}{=} [a_i]_{i=i_0}^{\infty} . \quad (4.19)$$

2. **Induktionsschritt.** Die  $k+1$ -te Differenz der Folge ist die Differenz ihrer  $k$ -ten Differenz, d. h.

$$\Delta^{(k+1)} \left( [a_i]_{i=i_0}^{\infty} \right) \stackrel{\text{def}}{=} \Delta \left( \Delta^{(k)} \left( [a_i]_{i=i_0}^{\infty} \right) \right) . \quad (4.20)$$

□

Beachtet sei, dass  $\Delta^{(1)}$  und  $\Delta$  identisch sind.

*Bemerkung 4.7* (Schreibweise für die Elemente der Differenzfolge bei mehrfacher Differenz). Das Element mit dem Index  $i$  von  $\Delta^{(k)} \left( [a_i]_{i=i_0}^{\infty} \right)$  wird vorwiegend mit  $\Delta^{(k)} a_i$  bezeichnet. Dadurch ist (4.20) in Anlehnung an (4.1) äquivalent mit

$$\Delta^{(k+1)} a_i = \Delta^{(k)} a_i - \Delta^{(k)} a_{i+1} . \quad (4.21)$$

□

**Lemma 4.8** (Explizite Formel für Mehrfache Differenz). [Vergl. Kno96, S. 252]. Es gilt

$$\Delta^{(k)} a_n = a_n - \binom{k}{1} a_{n+1} + \binom{k}{2} a_{n+2} - \dots + (-1)^k \binom{k}{k} a_{n+k} = \sum_{i=0}^k (-1)^i \binom{k}{i} a_{n+i} . \quad (4.22)$$

*Beweis.* Durch Induktion über  $k$ . Der Induktionsabschluss erfolgt durch Kombination von (4.21) und der allgemein bekannten Gleichung  $\binom{k}{i} + \binom{k}{i-1} = \binom{k+1}{i}$ . ■



wobei  $s_{n+j_0}$  die Spaltensumme bezüglich der Spalte  $n$  ist.

- (b) Wenn das Schema bezüglich aller Spalten konvergiert, dann kann die *Reihe der Spaltensummen*

$$\sum_{n=0}^{\infty} s_{n+j_0} \quad (4.25)$$

gebildet werden.

□

**Satz 4.10** (Cauchyscher Doppelreihensatz). [Vergl. Kno96, S. 144]. Der Gedankengang in Definition 4.9 wird unter Benutzung gleicher Notationen fortgesetzt. Ist für jedes  $k \in \mathbb{N}$  die Reihe der Zeile  $k$  gemäß der Gleichung

$$\sum_{j=0}^{\infty} |a_{k+i_0, j+j_0}| = \zeta_{k+i_0}$$

absolut konvergent, und ist außerdem die Reihe

$$\sum_{k=0}^{\infty} \zeta_{k+i_0}$$

konvergent; so konvergiert das Schema absolut bezüglich aller Spalten, außerdem konvergiert die Reihe der Spaltensummen gegen denselben Wert wie die Reihe der Zeilensummen, d. h.

$$\sum_{n=0}^{\infty} s_{n+j_0} = \sum_{k=0}^{\infty} z_{k+i_0} . \quad (4.26)$$

**Satz 4.11** (Großer Umordnungssatz für Reihen). [Vergl. Wal92, S. 100]. Es seien  $M$  eine abzählbare Menge,  $[I_i]_{i=1}^{\infty}$  eine Zerlegung von  $M$  in paarweise disjunkte Teilmengen  $I_i$  und  $\sum_{\alpha \in M} a_{\alpha}$  absolut konvergent. Dann ist

$$\sum_{\alpha \in M} a_{\alpha} = \sum_{i=1}^{\infty} \sum_{\alpha \in I_i} a_{\alpha} . \quad (4.27)$$

**Satz 4.12** (Satz von Markoff). [Vergl. Kno96, S. 250]. Der Gedankengang in Definition 4.9 wird unter Benutzung gleicher Notationen fortgesetzt. Die Eigenschaft der Konvergenz bezüglich aller Spalten wird nun vorausgesetzt und die Konvergenz der Reihe der Spaltensummen wird untersucht. Die Zeilenreste der Zeile  $k$  ab der Ordnungszahl  $m$  sind jeweils definiert durch

$$r_{k+i_0, m+j_0} = \sum_{n=m}^{\infty} a_{k+i_0, n+j_0} . \quad (4.28)$$

Es sei zusätzlich angenommen, dass für jedes  $m \in \mathbb{N}$  die Reihe der Zeilenreste ab Spaltenordnungszahl  $m$  gemäß der Gleichung

$$\sum_{k=0}^{\infty} r_{k+i_0, m+j_0} = R_{m+j_0} \quad (4.29)$$

konvergiert. Unter den ausgeführten Bedingungen gelten folgende Aussagen:

1.  $\sum_{n=0}^{\infty} s_{n+j_0}$  ist konvergent genau dann, wenn  $\lim_{m \rightarrow \infty} R_{m+j_0} = R$  existiert.
2. Es gilt (4.26) genau dann, wenn  $R = 0$  ist.

4.2.2 Beispiele

Wir wollen die Anwendung der Doppelreihe zuerst durch ein einfaches Beispiel illustrieren.

$$\begin{array}{rcccccccc}
 1 & = & 1 & & & & & & \\
 2x & = & x & + & x & & & & \\
 3x^2 & = & x^2 & + & x^2 & + & x^2 & & \\
 4x^3 & = & x^3 & + & x^3 & + & x^3 & + & x^3 \\
 5x^4 & = & x^4 & + & x^4 & + & x^4 & + & x^4 & + & x^4 \\
 \dots & & & & & & & & & & 
 \end{array}$$

---


$$\sum_{n=0}^{\infty} (n+1)x^n = \frac{1}{1-x} + \frac{x}{1-x} + \frac{x^2}{1-x} + \frac{x^3}{1-x} + \frac{x^4}{1-x} + \dots = \frac{1}{(1-x)^2}$$

Herleitung 4.1: Geschlossene Summation der Reihe  $\sum_{n=0}^{\infty} (n+1)x^n$  mit dem Ansatz der Doppelreihe

*Beispiel 4.13* (Geschlossene Auswertung der Reihe  $\sum_{n=0}^{\infty} (n+1)x^n$   $|x| < 1$ ). Die Reihe kann durch die Anwendung der gliedweisen Differenziation (siehe Satz A.32) auf die Reihe  $\sum_{n=0}^{\infty} x^n$  sofort hergeleitet werden. Hier begehen wir aber einen anderen Weg, nämlich durch Herleitung 4.1. Die Gültigkeit der Transformation ist durch Satz 4.10 in Kombination mit Satz A.21 feststellbar. □

Die Konstante  $\zeta(2)$  ist durch die Gleichung

$$\zeta(2) = \sum_{i=1}^{\infty} \frac{1}{i^2} , \tag{4.30}$$

definiert. Wir wollen nun ein interessantes Anwendungsbeispiel der Doppelreihe ausführen, nämlich die Transformation von  $\zeta(2)$ . Das Beispiel ist aus [Kno96, S. 274-275] entnommen.

*Beispiel 4.14* (Transformation von  $\zeta(2)$  mit Doppelreihe). Die Reihe für  $\zeta(2)$  in (4.30) wird nun mit dem Schema in Herleitung 4.2 umgeformt: (4.30) ist die Zeilenreihe, in Bezug auf Definition 4.9 werden sowohl  $i_0$  als auch  $j_0$  auf 1 gesetzt.

Der Zerlegung eines Terms zu einer Zeile

$$\frac{1}{k^2} = \frac{0!}{k(k+1)} + \frac{1!}{k(k+1)(k+2)} + \dots + \frac{(k-2)!}{k(k+1)\dots(2k-1)} + \frac{(k-1)!}{k^2(k+1)\dots(2k-1)} \tag{4.31}$$

liegt der Gedanke zugrunde, dass aus  $\frac{1}{k^2}$  einzelne Werte in der Form  $\frac{a(i)}{k(k+1)\dots(k+i)}$  möglichst ohne Rest extrahiert werden. Die Formel (4.31) wird dadurch hergeleitet, dass von (4.18) ausgehend die Substitution  $m \xrightarrow{\text{sub}} k-2$  durchgeführt wird, und dann in der resultierenden Formel die beiden Seiten mit  $\frac{1}{k}$  multipliziert werden.

$$\begin{aligned}
1 &= 1 \\
\frac{1}{2^2} &= \frac{0!}{2 \cdot 3} + \frac{1!}{2^2 \cdot 3} \\
\frac{1}{3^2} &= \frac{0!}{3 \cdot 4} + \frac{1!}{3 \cdot 4 \cdot 5} + \frac{2!}{3^2 \cdot 4 \cdot 5} \\
\frac{1}{4^2} &= \frac{0!}{4 \cdot 5} + \frac{1!}{4 \cdot 5 \cdot 6} + \frac{2!}{4 \cdot 5 \cdot 6 \cdot 7} + \frac{3!}{4^2 \cdot 5 \cdot 6 \cdot 7} \\
&\dots \\
\frac{1}{k^2} &= \frac{0!}{k(k+1)} + \frac{1!}{k(k+1)(k+2)} + \frac{2!}{k(k+1)(k+2)(k+3)} + \dots + \frac{(k-2)!}{k(k+1) \dots (2k-1)} \\
&\quad + \frac{(k-1)!}{k^2(k+1) \dots (2k-1)} \\
&\dots
\end{aligned}$$

Herleitung 4.2: Zur Verbesserung der Konvergenz von  $\zeta(2)$  mit dem Ansatz der Doppelreihen

Die Spalte  $n$  von Herleitung 4.2 lässt sich wie folgt auswerten:

$$\begin{aligned}
s_n &= (n-1)! \left[ \frac{1}{n^2(n+1)(n+2) \dots (2n-1)} + \frac{1}{n(n+1) \dots (2n)} \right] \\
&= 3 \frac{(n-1)!}{n(n+1) \dots (2n)} = 3 \frac{((n-1)!)^2}{2n!}, \quad (4.32)
\end{aligned}$$

woraus durch Anwendung von Satz 4.10 die Formel

$$\zeta(2) = \sum_{k=1}^{\infty} \frac{1}{k^2} = 3 \cdot \sum_{n=1}^{\infty} \frac{((n-1)!)^2}{(2n)!}$$

hergeleitet und ihre Gültigkeit nachgewiesen wird. Die Tatsache, dass Herleitung 4.2 die Voraussetzungen von Satz 4.10 erfüllt, ist offensichtlich, da alle Terme positiv sind.

Zum Abschluss schreiben wir das Ergebnis der Transformation in der Form

$$\zeta(2) = 3 \cdot \sum_{i=1}^{\infty} \frac{((i-1)!)^2}{(2i)!} \quad (4.33)$$

mit  $i$  als Indexvariable der Summation auf. □

Um ein weiteres interessantes Beispiel für die Anwendung der Doppelreihe zu erhalten, wird vom Autor eine Herleitung für die allgemeine Form der Eulerschen Reihentransformation ausgearbeitet, die im Folgenden ausgeführt wird. Die originale Formulierung von Satz 4.15 mit einer anderen Variante des Operator  $\Delta$  ist in [Mar96, S. 179] zu finden und mit dem Ansatz von Markoff (siehe Abschnitt 5.1) bewiesen.

**Satz 4.15** (Allgemeine Form der Eulerschen Reihentransformation). *Es gilt*

$$\sum_{i=0}^{\infty} a_i x^i = \frac{1}{1-x} \sum_{i=0}^{\infty} \left( \frac{x}{x-1} \right)^i \Delta^{(i)} a_0. \quad (4.34)$$

Zur Herleitung. Stellen wir zuerst die Hilfsgleichung

$$\sum_{n=0}^{\infty} \binom{n+k}{k} x^n = \frac{1}{(1-x)^{k+1}} \quad |x| < 1 \tag{4.35}$$

zur Verfügung, die aus der Reihe für  $\frac{1}{1-x}$  durch Induktion mit gliedweiser Differenziation oder Multiplikation hergeleitet werden kann.

$$\begin{array}{l} a_0 = (y+1) \cdot a_0 \cdot \left( \begin{array}{cccccc} 1 & -y & +y^2 & -y^3 & \pm \dots & +(-1)^k \binom{k}{0} y^k & + \dots \end{array} \right) \\ a_1 x = (y+1) \cdot a_1 \cdot \left( \begin{array}{cccccc} & y & -2y^2 & +3y^3 & \mp \dots & +(-1)^{k-1} \binom{k}{1} y^k & + \dots \end{array} \right) \\ a_2 x^2 = (y+1) \cdot a_2 \cdot \left( \begin{array}{cccccc} & & y^2 & -\binom{3}{2} y^3 & \pm \dots & +(-1)^{k-2} \binom{k}{2} y^k & + \dots \end{array} \right) \\ \dots \dots \dots \\ a_k x^k = (y+1) \cdot a_k \cdot \left( \begin{array}{cccccc} & & & & & y^k & + \dots \end{array} \right) \\ \dots \dots \dots \end{array}$$

Herleitung 4.3: Das Schema der Doppelreihe für die allgemeine Form der Eulerschen Transformation

Die Terme der linksstehenden Reihe von (4.34) werden jeweils mit der Substitution

$$x = \frac{y}{1+y} \tag{4.36}$$

unter Benutzung von (4.35) entwickelt ( $x$  ist mit  $-y$  zu instanzieren), für  $a_m x^m$  wird die Zeilenreihe

$$\begin{aligned} a_m x^m &= a_m \left( \frac{y}{1+y} \right)^m = a_m (y+1) y^m \left( \frac{1}{1+y} \right)^{m+1} \\ &= a_m (y+1) \left( \sum_{k=0}^{\infty} \binom{k+m}{m} y^{k+m} (-1)^k \right) = a_m (y+1) \left( \sum_{k=m}^{\infty} \binom{k}{m} y^k (-1)^{k-m} \right) \end{aligned}$$

gebildet. Demnach entsteht das Schema der Doppelreihe in Herleitung 4.3. Aus der Summe der Spalte für  $y^k$  wird der Faktor  $(y+1) y^k$  ausgeklammert, der andere Faktor lautet

$$\begin{aligned} \sum_{m=0}^k \binom{k}{m} a_m (-1)^{k-m} &= \sum_{m=0}^k \binom{k}{m} a_m (-1)^{k+m-2m} \\ &= \sum_{m=0}^k \binom{k}{m} a_m (-1)^{k+m} = (-1)^k \sum_{m=0}^k \binom{k}{m} a_m (-1)^m \stackrel{(4.22)}{=} (-1)^k \Delta^{(k)} a_0 . \end{aligned} \tag{4.37}$$

Darüber hinaus wird aus (4.36)

$$y = \frac{x}{1-x}$$

impliziert. Der Rest der Überführung der Reihe der Spaltensummen in die rechtsstehende Reihe von (4.34) ist offensichtlich. ■



**Korrolar 4.16** (Gewöhnliche Anwendungsform der Eulerschen Reihentransformation). *Konvergiert die Reihe  $\sum_{i=0}^{\infty} (-1)^i a_i$ , so konvergiert auch die Reihe  $\sum_{i=0}^{\infty} \frac{\Delta^{(i)} a_0}{2^{i+1}}$ , und es gilt*

$$\sum_{i=0}^{\infty} (-1)^i a_i = \sum_{i=0}^{\infty} \frac{\Delta^{(i)} a_0}{2^{i+1}} . \quad (4.38)$$

*Zur Herleitung.* Die Formel (4.38) ist eine Instanz der Formel (4.34), nämlich wenn  $x = -1$ . Darauf beachtet wird Satz A.29 angewandt. ■

Man findet die Formel (4.38) in [Kno96, S. 253], wo sie mit Hilfe von Satz 4.12 bewiesen wird. In der Fachliteratur wird häufig mit der Eulerschen Transformation nur die Formel (4.38) gemeint. Die Transformation gemäß Korrolar 4.16 kann  $\sum_{i=0}^{\infty} (-1)^i a_i$  in eine besser, gleichartig oder schlechter konvergente Reihe überführen. Ist jedoch  $a_i > 0$  für alle  $0 \leq i < \infty$  und  $[a_i]_{i=0}^{\infty}$  eine streng monotone Nullfolge, und ist außerdem stets  $\frac{a_{n+1}}{a_n} \geq a > \frac{1}{2}$ , so konvergiert die transformierte Reihe besser als die ursprüngliche. Die Transformation ist danach geeignet, praktisch wichtige Reihe, die nicht besonders gut konvergieren, in besser konvergente zu überführen. (Vergl. [JH84, S. 482]).

## 4.3 Wertverschiebung

Der Begriff Wertverschiebung wird in diesem Abschnitt formal eingeführt, um eine einheitliche Sicht für Kummersche, Gospersche und Eulersche Transformation zu schaffen. Der Inhalt des Begriffs geht auf Kummer ([Kum37], [Fab10, S. 125]) und Gosper ([Gos74]) zurück. Wir betrachten zunächst den einfachsten Spezialfall der zweistelligen Durchschnittsbildung.

### 4.3.1 Zweistellige Durchschnittsbildung

Die Bezeichnung „zweistellige Durchschnittsbildung“ wird vom Autor wegen (4.39) gewählt. Die Methode ist aus [Gos74, S. 2-4] bekannt, wo sie von Gosper angewandt wird, um die gewöhnliche Form der Eulerschen Transformation (vergl. Korrolar 4.16) herzuleiten.

**Definition 4.17** (Zweistellige Durchschnittsbildung). Die Zweistellige Durchschnittsbildung ist definiert durch

$$\cup \left( [a_i]_{i=i_0}^{\infty} \right) \stackrel{\text{def}}{=} \left[ \left( \frac{a_i + a_{i+1}}{2} \right) \right]_{i=i_0}^{\infty} . \quad (4.39)$$

Für die Bezugnahme auf die Elemente der Durchschnittsfolge sowie für die mehrfache Anwendung des Operators werden ähnliche Konventionen wie bei der Differenz benutzt. Es gelten also

$$\cup a_i = \frac{a_i + a_{i+1}}{2} , \quad (4.40)$$

$$\cup^{(0)} a_i = a_i \quad (4.41)$$

und

$$\cup^{(k+1)} a_i = \cup \left( \cup^{(k)} a_i \right) = \frac{\cup^{(k)} a_i + \cup^{(k)} a_{i+1}}{2} . \quad (4.42)$$

□

**Lemma 4.18** (Grundeigenschaft der zweistelligen Durchschnittsbildung). *Es gelten folgende Beziehungen*

$$\sum_{i=i_0}^{i_1} a_i = \frac{a_{i_0}}{2} + \left( \sum_{i=i_0}^{i_1} \cup a_i \right) - \frac{a_{i_1+1}}{2}, \quad (4.43)$$

$$\sum_{i=i_0}^{\infty} a_i = \frac{a_{i_0}}{2} + \sum_{i=i_0}^{\infty} \cup a_i \quad (4.44)$$

und

$$\sum_{i=i_0}^{\infty} a_i = \sum_{i=0}^k \frac{\cup^{(i)} a_{i_0+i}}{2} + \sum_{i=i_0}^{\infty} \cup^{(k+1)} a_i. \quad (4.45)$$

*Beweis.* Gleichung (4.43) folgt aus dem Assoziativgesetz der Addition (streng genommen durch Induktion über  $i_1$ ). Gleichung (4.44) folgt aus (4.43) durch Grenzübergang. Gleichung (4.45) folgt durch Induktion über  $k$  mit Induktionsanfang (4.44) und Induktionsschritt (4.42). ■

**Lemma 4.19** (Reihentransformation mit zweistelliger Durchschnittsbildung). *Gilt  $\lim_{k \rightarrow \infty} \sum_{i=i_0}^{\infty} \cup^{(k)} a_i = 0$ , dann gilt auch*

$$\sum_{i=i_0}^{\infty} a_i = \sum_{k=0}^{\infty} \frac{\cup^{(k)} a_{i_0}}{2}. \quad (4.46)$$

*Beweis.* Offensichtlich aus (4.45). ■

*Beispiel 4.20* (Transformation der Reihe für  $\ln 2$  in (3.22) mit zweistelliger Durchschnittsbildung). Für die Reihe gelten  $a_n = \frac{(-1)^{n+1}}{n}$  und  $i_0 = 1$ . Durch Herleitung 4.4 lässt sich die Gleichung

$$\cup^{(k)} a_n = \left(\frac{1}{2}\right)^k \cdot (-1)^{n+1} \cdot \left( \frac{k!}{n(n+1)\dots(n+k)} \right) = \frac{(-1)^{n+1} \cdot k!}{2^k \cdot n(n+1)\dots(n+k)} \quad (4.47)$$

erraten, die durch Induktion über  $k$  unter Benutzung von (4.42) leicht zu überprüfen ist. Außerdem gilt in Anlehnung an Satz A.18  $|\sum_{i=1}^{\infty} \cup^{(k)} a_i| \leq \cup^{(k)} a_1 = \frac{1}{(k+1) \cdot 2^k} \rightarrow 0$  für  $k \rightarrow \infty$ .

Folglich wird mit Lemma 4.19

$$\ln 2 = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} = \sum_{k=0}^{\infty} \frac{\cup^{(k)} a_i}{2} = \sum_{k=0}^{\infty} \frac{1}{(k+1) \cdot 2^{k+1}} = \sum_{k=1}^{\infty} \frac{1}{k \cdot 2^k} \quad (4.48)$$

hergeleitet. □

**Lemma 4.21** (Explizite Formel für mehrfache Durchschnittsbildung). *Es gilt*

$$\cup^{(k)} a_n = \frac{1}{2^k} \left( \sum_{n=0}^k \binom{k}{i} a_{n+i} \right). \quad (4.49)$$

*Beweis.* Analog zum Beweis von Lemma 4.8. ■

Die Entwicklung von (4.38) unter Benutzung von (4.22) und die Entwicklung von (4.46) unter Benutzung (4.49) führen bei geeigneter Interpretation der Ausgangsreihe zum gleichen Ergebnis. Dadurch lässt sich die gewöhnliche Anwendungsform der Eulerschen Transformation auch mit zweistelliger Durchschnittsbildung herleiten.

$$\begin{aligned}
 a_n &= \frac{(-1)^{n+1}}{n} \\
 \cup^{(1)} a_n &= \frac{1}{2} (a_n + a_{n+1}) = \frac{1}{2} \cdot (-1)^{n+1} \cdot \left( \frac{1}{n(n+1)} \right) \\
 \cup^{(2)} a_n &= \frac{1}{2} (\cup^{(1)} a_n + \cup^{(1)} a_{n+1}) = \left( \frac{1}{2} \right)^2 \cdot (-1)^{n+1} \cdot \left( \frac{2}{n(n+1)(n+2)} \right) \\
 \cup^{(3)} a_n &= \frac{1}{2} (\cup^{(2)} a_n + \cup^{(2)} a_{n+1}) = \left( \frac{1}{2} \right)^3 \cdot (-1)^{n+1} \cdot \left( \frac{2 \cdot 3}{n(n+1)(n+2)(n+3)} \right) \\
 &\dots\dots\dots \\
 \cup^{(k)} a_n &= \left( \frac{1}{2} \right)^k \cdot (-1)^{n+1} \cdot \left( \frac{k!}{n(n+1)(n+2)\dots(n+k)} \right)
 \end{aligned}$$

Herleitung 4.4: Zur Umformung der Reihe für  $\ln 2$  in (3.22) mit zweistelliger Durchschnittsbildung

**4.3.2 Wertverschiebung als Verallgemeinerung der Durchschnittsbildung**

**Definition 4.22** (Spaltungsfunktion und zweistellige Wertverschiebung). Es sei  $s : \mathbb{Z}_{\geq i_0} \rightarrow \mathbb{R}$  eine *Spaltungs-* bzw. *Verschiebungsfunktion*. Durch die Funktion wird ein Operator  $\overset{s}{\cup}$  induziert, der für alle  $[a_i]_{i=i_0}^\infty$  anwendbar und durch

$$\overset{s}{\cup} \left( [a_i]_{i=i_0}^\infty \right) \stackrel{\text{def}}{=} [a_i - s(i) \cdot a_i + s(i+1) \cdot a_{i+1}]_{i=i_0}^\infty \tag{4.50}$$

definiert ist. Für die Bezugnahme auf die Elemente von  $\overset{s}{\cup} \left( [a_i]_{i=i_0}^\infty \right)$  wird  $\overset{s}{\cup} a_i$  benutzt, so dass die Gleichung

$$\overset{s}{\cup} a_i = a_i - s(i) \cdot a_i + s(i+1) \cdot a_{i+1} \tag{4.51}$$

gilt. □

**Lemma 4.23** (Grundeigenschaft der Wertverschiebung). *Es gilt*

$$\sum_{i=i_0}^{i_1} \overset{s}{\cup} a_i = s(i_0) \cdot a_{i_0} + \left( \sum_{i=i_0}^{i_1} a_i \right) - s(i_1 + 1) \cdot a_{i_1+1} . \tag{4.52}$$

Gilt  $\lim_{i \rightarrow \infty} s(i) \cdot a_i = 0$ , so gilt auch

$$\sum_{i=i_0}^\infty a_i = s(i_0) \cdot a_{i_0} + \sum_{i=i_0}^\infty \overset{s}{\cup} a_i . \tag{4.53}$$

*Beweis.* Analog zu den entsprechenden Teilen des Beweises von Lemma 4.18. ■

Für  $s : \mathbb{Z}_{\geq i_0} \rightarrow \mathbb{R} \mid s(i) = \frac{1}{2}$  ist  $\overset{s}{\cup}$  mit  $\cup$  identisch. Dementsprechend ist die Wertverschiebung die allgemeinere Form der Durchschnittsbildung.

Die Wertverschiebung ist einer der zwei Grundgedanken der Gosperschen Reihentransformation (vergl. [Gos74]), die in Kapitel 6 vorgestellt wird. Die Schlüsselidee der Benutzung einer

Spaltungsfunktion für Reihentransformationen ist vorher schon in der Kummerschen Transformation ([Kum37], [Fab10, S. 125]) enthalten, die jedoch ursprünglich *nur für Reihen mit positiven und asymptotisch gleichen Termen* gedacht war. Die Durchschnittsbildung ist wohl für die Entstehung der Wertverschiebung aus der Kummerschen Transformation entscheidend, weil sie den Hinweis darauf gibt, dass die Idee der Spaltungsfunktion auch beim Weglassen der oben erwähnten Einschränkungen angewandt werden kann.

### 4.3.3 Kummersche Transformation als Spezialfall der Wertverschiebung

Die Spezialisierung der Wertverschiebung auf die Kummersche Transformation kommt durch Definition 4.24 zum Ausdruck, wobei die Spaltungsfunktion in Form der Transformationsfolge vorkommt. Historisch ist die Wertverschiebung aus der Kummerschen Transformation durch Generalisierung entstanden, wie es am Ende des vorigen Abschnittes erwähnt ist.

Dem Autor liegt die Darstellung der Kummerschen Transformation in [Fab10, S. 125] am nächsten. Die Darstellungen in vielen anderen Quellen (z. B. [Kno96, S. 255], [Wei] und [AS72, S. 16]) wären übermäßig vereinfacht, wodurch die Schlüsselidee des Einsatzes von Spaltungsfunktionen nicht mehr zu erkennen ist. Diese Schlüsselidee sieht man jedoch in der originalen Arbeit [Kum37] von Kummer.

**Definition 4.24** (Kummersche Transformationsfolge). Es sei  $\sum_{i=i_0}^{\infty} a_i$  eine Reihe. Eine Folge  $[\lambda_i]_{i=i_0}^{\infty}$  heißt eine *Kummersche Transformationsfolge* für  $\sum_{i=i_0}^{\infty} a_i$ , wenn sie die Eigenschaften

$$\lim_{n \rightarrow \infty} \lambda_n - \lambda_{n+1} \frac{a_{n+1}}{a_n} = 1 \quad (4.54)$$

und

$$\lim_{n \rightarrow \infty} \lambda_n \cdot a_n = 0 \quad (4.55)$$

erfüllt. □

Die Intuition von (4.55) für den Grenzübergang ist aus der Voraussetzung für (4.53) offensichtlich. Durch Betrachtung des Quotienten

$$\frac{\overset{\lambda}{\cup} a_n}{a_n} = 1 - \left( \lambda_n - \lambda_{n+1} \frac{a_{n+1}}{a_n} \right) \quad (4.56)$$

ist die Bedeutung von (4.54) ersichtlich: der Quotient soll gegen 0 streben, damit die Terme der neuen Reihe wesentlich kleiner als die Terme der originalen Reihe werden.

Das Entscheidende bei der Anwendung der Kummerschen Transformation ist die Konstruktion der Transformationsfolge. Für  $\lambda_n$  werden im Allgemeinen rationale Funktionen von  $n$  als Ansatz genommen. Außerdem gibt ein Verfahren, das die Konstruktion dadurch vereinfacht, dass eine Transformationsfolge mit der Eigenschaft (4.54) aus einer Transformationsfolge mit der schwächeren Eigenschaft (4.57) erzeugt wird. Dieses Verfahren manifestiert sich in Definition 4.25 und Lemma 4.26.

**Definition 4.25** (Unnormierte Kummersche Transformationsfolge). Die Folge  $[\lambda_i]_{i=i_0}^{\infty}$  aus

Definition 4.24 heißt eine *unnormierte Kummersche Transformationsfolge* für  $\sum_{i=i_0}^{\infty} a_i$ , wenn sie

anstelle von (4.54) die Gleichung

$$\lim_{n \rightarrow \infty} \lambda_n - \lambda_{n+1} \frac{a_{n+1}}{a_n} = \chi \neq 0 \quad (4.57)$$

erfüllt. Es werden dann

$$\chi_n = \lambda_n - \lambda_{n+1} \frac{a_{n+1}}{a_n} \quad (4.58)$$

und

$$a'_n = a_n - \frac{\lambda_n \cdot a_n - \lambda_{n+1} \cdot a_{n+1}}{\chi} \quad (4.59)$$

gesetzt.  $\square$

**Lemma 4.26** (Transformation mit unnormierter Kummerschen Folge). *Mit Definition 4.25 gelten die Gleichungen*

$$a'_n = a_n \cdot \left(1 - \frac{\chi_n}{\chi}\right) \quad (4.60)$$

und

$$\sum_{i=i_0}^{\infty} a_i = \frac{\lambda_{i_0} \cdot a_{i_0}}{\chi} + \sum_{i=i_0}^{\infty} a'_i . \quad (4.61)$$

*Beweis.* Gleichung (4.60) folgt aus (4.58) und (4.59). Gleichung (4.61) folgt aus (4.55) und (4.59).  $\blacksquare$

Wir wollen uns nun ein interessantes Anwendungsbeispiel der Kummerschen Transformation ansehen, nämlich eine Transformation der Reihe (4.30) für  $\zeta(2)$ , die zum gleichen Ergebnis wie in Beispiel 4.14 führt. Das Beispiel wird anhand [Fab10, S. 126-128] ausgearbeitet.

*Beispiel 4.27* (Kummersche Transformation und  $\zeta(2)$ ). Zuerst wollen wir die allgemeine, durch  $p$  und  $c$  parametrisierte Reihe

$$S(c, p) = \sum_{n=1}^{\infty} \frac{1}{(n+c)^2 (n+c+1)^2 \dots (n+c+p-1)^2} \quad (4.62)$$

anhand von Definition 4.25 und Lemma 4.26 umformen, um eine Rekursionsformel für  $S(c, p)$  aufzustellen.

- Für die originale Reihe (4.62) gilt

$$\frac{a_{n+1}}{a_n} = \frac{(n+c)^2}{(n+c+p)^2} . \quad (4.63)$$

Für die Anwendung der Kummerschen Transformation wird der Ansatz

$$\lambda_n = n + c + p - 1$$

herangezogen, der die Berechnungen

$$\chi_n \stackrel{(4.58)}{=} 2p - 1 - \frac{p^2}{n+c+p} , \quad \chi \stackrel{(4.57)}{=} 2p - 1 , \quad 1 - \frac{\chi_n}{\chi} = \frac{p^2}{(2p-1)(n+c+p)} ,$$

$$\text{und} \quad a'_n \stackrel{(4.60)}{=} \frac{p^2}{(2p-1)} \cdot \frac{1}{(n+c)^2 (n+c+1)^2 \dots (n+c+p-1)^2 (n+c+p)}$$

induziert, woraus sich unter Benutzung von (4.61) die Gleichung

$$S(c, p) = \frac{c+p}{(2p-1)(c+1)^2 \dots (c+2)^2 (c+p)^2} + \frac{p^2}{2p-1} \underbrace{\sum_{n=1}^{\infty} \frac{1}{(n+c)^2 (n+c+1)^2 \dots (n+c+p-1)^2 (n+c+p)}}_{U(c,p)} \quad (4.64)$$

ergibt.

- Für die neu entstehende Reihe

$$U(c, p) \stackrel{\text{def}}{=} \sum_{n=1}^{\infty} \frac{1}{(n+c)^2 (n+c+1)^2 \dots (n+c+p-1)^2 (n+c+p)} \quad (4.65)$$

gilt

$$\frac{a_{n+1}}{a_n} = \frac{(n+c)^2}{(n+c+p)(n+c+p+1)}. \quad (4.66)$$

Für die Anwendung der Kummerschen Transformation wird der Ansatz

$$\lambda_n = n + c + p$$

herangezogen, der die Berechnungen

$$\chi_n \stackrel{(4.58)}{=} 2p - \frac{p^2}{n+c+p}, \quad \chi \stackrel{(4.57)}{=} 2p, \quad 1 - \frac{\chi_n}{\chi} = \frac{p}{2(n+c+p)},$$

und  $a'_n \stackrel{(4.60)}{=} \frac{p}{2(n+c)^2 (n+c+1)^2 \dots (n+c+p)^2}$

induziert, woraus sich unter Benutzung von (4.61) die Gleichung

$$U(c, p) = \frac{1}{2p(c+1)^2 (c+2)^2 \dots (c+p)^2} + \sum_{n=1}^{\infty} \frac{p}{2(n+c)^2 (n+c+1)^2 \dots (n+c+p)^2} \quad (4.67)$$

ergibt.

- Aus (4.64) und (4.67) wird

$$S(c, p) = \frac{2c+3p}{2(2p-1)(c+1)^2 (c+2)^2 \dots (c+p)^2} + \frac{p^3}{2(2p-1)} \sum_{n=1}^{\infty} \frac{1}{(n+c)^2 (n+c+1)^2 \dots (n+c+p)^2} \\ = \frac{2c+3p}{2(2p-1)(c+1)^2 (c+2)^2 \dots (c+p)^2} + \frac{p^3}{2(2p-1)} S(c, p+1)$$

impliziert, woraus durch Abtrennung der Zwischenschritte

$$S(c, p) = \frac{2c+3p}{2(2p-1)(c+1)^2 (c+2)^2 \dots (c+p)^2} + \frac{p^3}{2(2p-1)} S(c, p+1) \quad (4.68)$$

resultiert.

Jetzt leiten wir aus der Rekursionsformel (4.68) eine Formel für  $\zeta(2)$  her. Für  $c = 0$  und  $p = 1, 2, 3 \dots, n$  werden die Instanzen von (4.68) herangezogen, um  $\zeta(2) = S(0, 1)$  Schritt für Schritt zu entwickeln. Als Ergebnis erhalten wir

$$\begin{aligned} \zeta(2) &= S(0, 1) \\ &= \sum_{i=1}^n \left( \frac{3^i}{2(2i-1)(i!)^2} \prod_{j=1}^{i-1} \frac{j^3}{2(2j-1)} \right) + \left( \prod_{i=1}^n \frac{i^3}{2(2i-1)} \right) S(c, n+1) \\ &= \left( \sum_{i=1}^n \frac{3(i-1)!}{2^i \cdot i \cdot 1 \cdot 3 \dots (2i-1)} \right) + \frac{(n!)^3}{2^n \cdot 1 \cdot 3 \dots (2n-1)} S(c, n+1) \\ &= 3 \cdot \left( \sum_{i=1}^n \frac{(i-1)!^2}{(2i)!} \right) + \frac{(n!)^3}{2^n \cdot 1 \cdot 3 \dots (2n-1)} S(c, n+1) , \end{aligned}$$

woraus sich nach Abtrennung der Zwischenschritte

$$\zeta(2) = 3 \cdot \left( \sum_{i=1}^n \frac{(i-1)!^2}{(2i)!} \right) + \frac{(n!)^3}{2^n \cdot 1 \cdot 3 \dots (2n-1)} S(c, n+1) \quad (4.69)$$

ergibt. Aus (4.69) kommt man intuitiv zu Gleichung (4.33), die auch tatsächlich gilt. Um aus (4.69) die Formel (4.33) für  $\zeta(2)$  formal herzuleiten, ist nur

$$\lim_{n \rightarrow \infty} \frac{(n!)^3}{2^n \cdot 1 \cdot 3 \dots (2n-1)} S(c, n+1) = 0$$

zu beweisen. Der Beweis ist mit Hilfe der Ungleichung

$$S(c, n+1) < \frac{1}{(n!)^2}$$

möglich.

Zum Abschluss des Beispiels bringen wir (4.69) dadurch in die kurze und explizite Form

$$\zeta(2) = 3 \cdot \left( \sum_{i=1}^n \frac{(i-1)!^2}{(2i)!} \right) + \frac{(n!)^4}{(2n)!} \sum_{k=0}^{\infty} \frac{k!^2}{(k+n+1)!^2} , \quad (4.70)$$

dass  $S(c, n+1)$  durch Anwendung von (4.62) entwickelt wird und anschließend einige einfache Umformungen durchgeführt werden.  $\square$

## Kapitel 5

# Markoffsche Differenzrechnung und WZ-Methode

In diesem Kapitel werden drei verwandte Methoden für Reihentransformationen vorgestellt, die jeweils von Markoff, Zeilberger und Wilf stammen. Sie beruhen alle auf zwei zweistelligen Funktionen, die durch eine differenzartige Funktionalgleichung abgestimmt sind. Diesbezüglich weisen sie auch eine enge Verwandtschaft zur Idee der Doppelreihe und Differenz auf. Darauf wird am Ende des Kapitels gesondert eingegangen.

Die Methoden in diesem Kapitel genießen eine starke Unterstützung durch Computeralgebrasysteme. Die Markoffsche Methode wurde zwar ohne Computeralgebrasystem entwickelt, ein schwieriger Teil der für die Anwendung notwendigen Arbeit lässt sich jedoch zur heutigen Zeit automatisch durchführen. Im Fall der Methoden von Zeilberger und Wilf kann man in gewissem Sinn sagen, dass ihre Entwicklung bzw. Entstehung durch die Bereitstellung vom Gosper-Algorithmus und Zeilberger-Algorithmus im Kontext eines Computeralgebrasystems bedingt ist.

Unter Benutzung eines Computeralgebrasystems wird der Mensch von langwierigen Routineberechnungen und -umformungen befreit. Er kann sich nun auf die Lösungsidee konzentrieren, sie als ein Programm niederschreiben, und dann Schritt für Schritt verbessern. Dadurch lassen sich auch für schwierige Probleme, die vorher nur mit einem tiefen menschlichen Einsichtsvermögen lösbar sind, automatisiert abarbeitbare Lösungsalgorithmen entwickeln. Betrachten wir dazu die folgende Bemerkung von Knuth aus [PWZ96, Vorwort]:

*Science is what we understand well enough to explain to a computer. Art is everything else we do. During the past several years an important part of mathematics has been transformed from an Art to a Science: No longer do we need to get a brilliant insight in order to evaluate sums of binomial coefficients, and many similar formulas that arise frequently in practice; we can now follow a mechanical procedure and discover the answers quite systematically.*

Im Kontext des Einsatzes von Computeralgebrasystemen gehören der Gosper-Algorithmus und der Zeilberger-Algorithmus zu den wichtigsten Errungenschaften der Mathematik in den letzten 30 Jahren. Mit ihnen lässt sich ein ansonsten sehr schwieriger Teil der für die Reihentransformation notwendigen Arbeit — Konstruktion einer Funktion aus einer anderen anhand einer differenzartigen Funktionalgleichung — automatisch durchführen. Im Rahmen der vorliegenden Dissertation werden die beiden Algorithmen nur aus Anwendersicht betrachtet. Abschnitt 5.4 führt eine kurze Charakterisierung der Algorithmen aus. Beispiele für den praktischen Einsatz unter dem Computeralgebrasystem Maple sind in Anhang B.2 und Anhang B.3 enthalten.



Bei der Ausarbeitung des Inhalts dieses Kapitels wurde Maple zu Hilfe genommen. Unwesentliche Details der Berechnung werden ausgelassen, nur Bedingungen, Initialisierungen, Ergebnisse und Bemerkungen zu den wichtigen Punkten werden ausgeführt. Dadurch kann die Lösungsidee auch besser verfolgt werden.

### 5.1 Markoffsche Rechtecktransformation

Die Darstellung der Markoffschen Methode in diesem Abschnitt basiert auf [Mar96, S. 178-179] und [Mar90, S. 1]. Die grundlegenden Beziehungen sind in Lemma 5.1 dargestellt und die Anwendungsformen sind in Satz 5.2 formuliert. Die Bezeichnung Rechtecktransformation wird vom Autor der vorliegenden Dissertation wegen der Form des Schemas (5.4) gewählt.

**Lemma 5.1** (Doppelsumme). *Es seien  $F(n, k)$  und  $G(n, k)$  zwei zweistellige diskrete Funktion, die die Gleichung*

$$F(n + 1, k) - F(n, k) = G(n, k + 1) - G(n, k) \tag{5.1}$$

erfüllen. So gilt

$$\sum_{k=0}^{k_0} F(0, k) - \sum_{k=0}^{k_0} F(n_0, k) = \sum_{n=0}^{n_0} G(n, 0) - \sum_{n=0}^{n_0} G(n, k_0) . \tag{5.2}$$

*Beweis.* Aufgrund von (5.1) setzen wir

$$d(n, k) \stackrel{\text{def}}{=} F(n, k) - F(n + 1, k) = G(n, k) - G(n, k + 1) \tag{5.3}$$

und betrachten die Doppelsumme

$$\begin{array}{cccccccc} d(0, 0) & +d(0, 1) & +d(0, 2) & + \dots & +d(0, k_0 - 1) & & & \\ + d(1, 0) & +d(1, 1) & +d(1, 2) & + \dots & +d(1, k_0 - 1) & & & \\ + d(2, 0) & +d(2, 1) & +d(2, 2) & + \dots & +d(2, k_0 - 1) & & & \\ \dots & \dots & \dots & \dots & \dots & & & \\ \dots & \dots & \dots & \dots & \dots & & & \\ + d(n_0 - 1, 0) & +d(n_0 - 1, 1) & +d(n_0 - 1, 2) & + \dots & +d(n_0 - 1, k_0 - 1) & & & \end{array} . \tag{5.4}$$

Berechnen wir die Doppelsumme spaltenweise, so ergibt sich die linke Seite von (5.2); berechnen wir die Doppelsumme zeilenweise, so ergibt sich die rechte Seite von (5.2). Infolgedessen sind die beiden Seiten von (5.2) gleich zu der Doppelsumme und zueinander. ■

**Satz 5.2** (Markoffsche Rechtecktransformation). *Der Gedankengang in Lemma 5.1 wird unter Benutzung gleicher Notationen fortgesetzt.*

1. Sind die Reihen  $\sum_{k=0}^{\infty} F(0, k)$  sowie  $\sum_{n=0}^{\infty} G(n, 0)$  konvergent und gilt außerdem

$$\lim_{n_0, k_0 \rightarrow \infty} \sum_{k=0}^{k_0} F(n_0, k) = \lim_{n_0, k_0 \rightarrow \infty} \sum_{n=0}^{n_0} G(n, k_0) = 0 ,$$

so gilt auch

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{n=0}^{\infty} G(n, 0) . \tag{5.5}$$

2. Es sei  $n_0$  eine feste Zahl. Sind die Reihen  $\sum_{k=0}^{\infty} F(0, k)$  sowie  $\sum_{k=0}^{\infty} F(n_0, k)$  konvergent und gilt außerdem  $\lim_{k \rightarrow \infty} \sum_{n=0}^{n_0} G(n, k) = 0$ , so gilt auch

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{n=0}^{n_0} G(n, 0) + \sum_{k=0}^{\infty} F(n_0, k) . \quad (5.6)$$

*Beweis.* Jeweils unter Benutzung von (5.2) durch Grenzübergang. ■

Die Markoffsche Methode kommt dadurch zur Anwendung, dass eine Reihe in Übereinstimmung mit  $\sum_{k=0}^{\infty} F(0, k)$  gebracht und dann durch die rechte Seite von (5.5) oder (5.6) ersetzt wird, wobei im Allgemeinen der erste Fall anzustreben ist. In Bemerkung 5.19 wird diskutiert, wie  $G(n, k)$  unter bestimmten Bedingungen mit dem Gosper-Algorithmus aus  $F(n, k)$  automatisch berechnet werden kann.

Wir wollen uns nun eine Anwendung der Methode ansehen. In Beispiel 5.3 wird die Transformation der Reihe (4.30) in die Reihe (4.33) ausgeführt. Das Beispiel wird anhand des Stoffes in [Mar96, S. 187] konstruiert.

*Beispiel 5.3* (Markoffsche Rechtecktransformation für  $\zeta(2)$ ). Zwei zweistellige Funktionen  $F(n, k)$  und  $G(n, k)$  mit der expliziten Definition

$$F(n, k) = \begin{cases} 1 & \text{falls } n = 0 \\ \frac{1}{(k+1)^2} & \\ \frac{1}{2} \frac{((n-1)!)^4 n^3 (k!)^2}{(2n-1)! ((n+k+1)!)^2} & \text{falls } n \geq 1 \end{cases}, \quad G(n, k) = \frac{3n+2k+3}{4n+2} \cdot F(n, k)$$

werden betrachtet.  $\sum_{k=0}^{\infty} F(0, k)$  ist der Reihe in (4.30) gleich. Mit  $G(n, k)$  erfüllt  $F(n, k)$  Gleichung (5.1).  $\sum_{k=0}^{\infty} G(n, 0)$  lässt sich in (4.33) überführen. Auch die Nebenbedingung für Satz 5.2, Punkt 1 lässt sich überprüfen. Satz 5.2, Punkt 1 kann also für  $F(n, k)$  und  $G(n, k)$  angewandt werden. Die Anwendung bewirkt die Transformation von (4.30) in (4.33). □

**Definition 5.4** (Markoff-Paar). Erfüllen  $F(n, k)$  und  $G(n, k)$  Gleichung (5.1), so heißt  $(F(n, k), G(n, k))$  ein *Markoff-Paar*.  $F(n, k)$  heißt die *linke* und  $G(n, k)$  die *rechte Adjungierte* des Markoff-Paares. In einer Beziehung der Form (5.1) ist die linke Adjungierte diejenige Funktion, deren Differenz in Bezug auf die linke Variable gebildet wird. Entsprechendes gilt auch für die rechte Adjungierte. □

Jede Transformation mit der Markoffschen Methode ist mit einem Markoff-Paar versehen. In Beispiel 5.3 ist nur das Markoff-Paar, aber nicht seine Konstruktion dargestellt. Um ein Gefühl für die Konstruktion von Markoff-Paaren zu vermitteln, führen wir Beispiel 5.5 aus. Das Beispiel basiert auf den Stoff in [Mar96, S. 185].

*Beispiel 5.5* (Ansatz der Rechtecktransformation für Schellbachsche Formel). Die Reihe

$$1 + \frac{\alpha a}{\beta b} + \frac{\alpha(\alpha+1)a(a+1)}{\beta(\beta+1)b(b+1)} + \frac{\alpha(\alpha+1)(\alpha+2)a(a+1)(a+2)}{\beta(\beta+1)(\beta+2)b(b+1)(b+2)} + \dots \quad (5.7)$$

ist gegeben. Es sei ein Markoff-Paar  $(F(n, k), G(n, k))$  zu konstruieren, so dass  $\sum_{k=0}^{\infty} F(0, k)$  mit (5.7) übereinstimmt.

Zuerst wird eine Hilfsfunktion  $S(n, k)$  durch Imitation der Terme von (5.7) konstruiert, die durch

$$\left. \begin{array}{l} i) \quad S(0, 0) = 1 \\ ii) \quad S(n+1, k) = \frac{1}{(a+k+n)(b+k+n)} S(n, k) \\ iii) \quad S(n, k+1) = \frac{(\alpha+k)(\beta+k)}{(a+k+n)(b+k+n)} S(n, k) \end{array} \right\} \quad (5.8)$$

definiert ist. Anschließend wird für  $F(n, k)$  und  $G(n, k)$  der Ansatz

$$\left. \begin{array}{l} i) \quad F(n, k) = S(n, k) A_n \\ ii) \quad G(n, k) = S(n, k) (B_n + C_n k) \end{array} \right\} \quad (5.9)$$

angenommen, in der Absicht,  $A_n$ ,  $B_n$  und  $C_n$  aufgrund von (5.1) leicht berechnen zu können.

Jetzt werden nach  $A_n$ ,  $B_n$  und  $C_n$  gesucht. Dazu entwickeln wir (5.1) unter Beachtung von (5.9), (5.8) ii) und (5.8) iii) zu einer Gleichung über  $n$ ,  $k$ ,  $A_n$ ,  $B_n$ ,  $C_n$ ,  $A_{n+1}$ ,  $B_{n+1}$ ,  $C_{n+1}$  und  $S(n, k)$ , aus der sich anschließend

$$(a+n+k)(b+n+k)A_n - A_{n+1} = (a+n+k)(b+n+k)(B_n + C_n k) - (\alpha+k)(\beta+n)(B_n + C_n k + C_n)$$

ergibt. Unter Benutzung der letzten Gleichung für den Koeffizientenvergleich in Bezug auf  $k$  erhalten wir das folgende Gleichungssystem

$$\left. \begin{array}{l} A_n = (2n+a+b-\alpha-\beta-1)C_n \\ (2n+a+b)A_n = (2n+a+b-\alpha-\beta)B_n + ((a+n)(b+n)-\alpha-\beta-\alpha-\beta)C_n \\ (a+n)(b+n)A_n - A_{n+1} = ((a+n)(b+n)-\alpha-\beta)B_n - \alpha-\beta C_n \end{array} \right\} \quad (5.10)$$

Aus (5.10) folgt

$$\left. \begin{array}{l} C_n = \frac{1}{2n+a+b-\alpha-\beta-1} A_n \\ B_n = \frac{(2n+a+b-\alpha-1)(2n+a+b-\beta-1) - (n+a-1)(n+b-1)}{(2n+a+b-\alpha-\beta-1)(2n+a+b-\alpha-\beta)} A_n \\ A_{n+1} = \frac{(n+a-\alpha)(n+a-\beta)(n+b-\alpha)(n+b-\beta)}{(2n+a+b-\alpha-\beta-1)(2n+a+b-\alpha-\beta)} A_n \end{array} \right\} \quad (5.11)$$

Mit Hilfe von (5.11) lassen sich  $A_n$ ,  $B_n$  und  $C_n$  aus  $A_0$  eindeutig bestimmen. Für  $A_0 = 1$  werden in Kombination mit (5.8) i), (5.8) iii) und (5.9) i) die Werte von  $F(0, k)$  berechnet, die  $\sum_{k=0}^{\infty} F(0, k)$  in Übereinstimmung mit (5.7) bringen. Der Ansatz führt folglich zum Ziel.  $\square$

*Bemerkung 5.6* (Rekursions- versus explizite Ansatzformeln). In der originalen Arbeit [Mar96, S. 185] wird der explizite Ansatz

$$\left. \begin{array}{l} F(n, k) = \frac{\alpha(\alpha+1)\dots(\alpha+k-1)\beta(\beta+1)\dots(\beta+k-1)}{a(a+1)\dots(a+n+k-1)b(b+1)\dots(b+n+k-1)} A_n \\ G(n, k) = \frac{\alpha(\alpha+1)\dots(\alpha+k-1)\beta(\beta+1)\dots(\beta+k-1)}{a(a+1)\dots(a+n+k-1)b(b+1)\dots(b+n+k-1)} (B_n + C_n k) \end{array} \right\}$$



Summieren wir die Terme des Schemas zeilenweise, so ergibt sich  $\sum_{i=0}^n G(i, 0)$ . Summieren wir die Terme des Schemas spaltenweise, so ergibt sich  $\sum_{i=0}^n \left( G(i, i) + F(i+1, i) - F(n+1, i) \right)$ . Die beiden Ergebnisse der Summation müssen sich gleichen, d. h. die Gleichung

$$\sum_{i=0}^n G(i, 0) = \sum_{i=0}^n \left( G(i, i) + F(i+1, i) - F(n+1, i) \right)$$

gilt, und sie ist zu (5.13) äquivalent. ■

**Satz 5.9** (Zeilbergersche Dreieckstransformation). *Zwei Funktionen  $F(n, k)$  und  $G(n, k)$  sollen als erstes die Eigenschaft (5.1) erfüllen; außerdem sei angenommen, dass alle in (5.15) vorkommenden Grenzwerte existieren. So gilt*

$$\sum_{n=0}^{\infty} G(n, 0) = \sum_{n=1}^{\infty} \left( F(n, n-1) + G(n-1, n-1) \right) - \lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} F(n, k) . \quad (5.15)$$

Gilt zusätzlich

$$\lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} F(n, k) = 0 , \quad (5.16)$$

so gilt auch

$$\sum_{n=0}^{\infty} G(n, 0) = \sum_{n=1}^{\infty} \left( F(n, n-1) + G(n-1, n-1) \right) . \quad (5.17)$$

*Beweis.* Gleichung (5.13) wird durch die Substitution  $n \xrightarrow{\text{sub}} n-1$  in der gesamten Formel und die Indexverschiebung  $i \xrightarrow{\text{shift}} 1$  in der ersten Summe der rechten Seite in

$$\sum_{i=0}^{n-1} G(i, 0) = \sum_{i=1}^n \left( F(i, i-1) + G(i-1, i-1) \right) - \sum_{i=0}^{n-1} F(n, i) \quad (5.13')$$

überführt. Der Rest des Beweises erfolgt durch Grenzübergang. ■

Die Zeilbergersche Methode kommt dadurch zur Anwendung, dass eine Reihe Übereinstimmung mit  $\sum_{n=0}^{\infty} G(n, 0)$  in gebracht und dann durch die rechte Seite von (5.15) oder (5.17) ersetzt wird, wobei im Allgemeinen der zweite Fall anzustreben ist. In Bemerkung 5.19 wird diskutiert, wie  $F(n, k)$  unter bestimmten Bedingungen mit dem Gosper-Algorithmus aus  $G(n, k)$  automatisch berechnet werden kann.

Wir wollen nun die Anwendung der Methode anhand eines Beispiels veranschaulichen. Das Markoff-Paar in Beispiel 5.10 wird vom Autor in Anlehnung an Beispiel 4.14 konstruiert.

*Beispiel 5.10* (Einbettung von Beispiel 4.14). Die Reihe in (4.30) ist zu transformieren. Für die Unifikation der Reihe mit  $\sum_{n=0}^{\infty} G(n, 0)$  wird der Ansatz

$$G(n, k) = \frac{k! n!}{(n+1)(n+k+1)!} \quad (5.18)$$

herangezogen. Daraus lässt sich mit dem Gosper-Algorithmus (siehe Beispiel B.3)

$$F(n, k) = \frac{k! n!}{(k+1)(n+k+1)!} \quad (5.19)$$

berechnen. Die Erfüllung von (5.1) sowie die Übereinstimmung von  $\sum_{n=0}^{\infty} G(n, 0)$  mit (4.30) sind leicht zu überprüfen. Aus (5.19) und (5.18) werden berechnet

$$\left. \begin{aligned} i) \quad G(n, 0) &= \frac{1}{(n+1)^2} \\ ii) \quad F(n, n-1) + G(n-1, n-1) &= 3 \frac{((n-1)!)^2}{(2n)!} \end{aligned} \right\} . \quad (5.20)$$

Außerdem gilt die Gleichung

$$\lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} F(n, k) = \lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} \frac{k! n!}{(k+1)(n+k+1)!} = 0 , \quad (5.21)$$

die in Beweis B.1 nachgewiesen wird. Aus (5.20) und (5.21) wird unter Benutzung von (5.17) die Formel

$$\sum_{n=0}^{\infty} \frac{1}{(n+1)^2} = \sum_{n=1}^{\infty} \left( 3 \frac{((n-1)!)^2}{(2n)!} \right)$$

hergeleitet. □

Wir betrachten im Folgenden ein weiteres Beispiel, das die Transformation der Konstanten  $\zeta(3)$  beinhaltet, die durch

$$\zeta(3) = \sum_{n=1}^{\infty} \frac{1}{n^3} \quad (5.22)$$

definiert ist. Das Beispiel ist aus [Zei93, S. 597] entnommen, wobei die Darstellung in [Zei93] auf [Apé79] bzw. [Poo79] zurückgeht.

*Beispiel 5.11* (Dreieckstransformation und Apéry's Formel für  $\zeta(3)$ ). Die Reihe in (5.22) ist zu transformieren. Für die Unifikation der Reihe mit  $\sum_{n=0}^{\infty} G(n, 0)$  wird der Ansatz

$$G(n, k) = \frac{(-1)^k (k!)^2 (n-k)!}{(n+k+1)!(n+1)^2} \quad (5.23)$$

herangezogen. Daraus lässt sich mit dem Gosper-Algorithmus

$$F(n, k) = \frac{1}{2} \frac{(k!)^2 (n-k-1)! (-1)^k}{(n+k+1)!(k+1)} \quad (5.24)$$

berechnen. Die Erfüllung von (5.1) durch  $F$  und  $G$  sowie die Übereinstimmung von  $\sum_{n=0}^{\infty} G(n, 0)$  mit (5.22) sind leicht zu überprüfen.

Aus (5.24) und (5.23) werden berechnet

$$\left. \begin{aligned} i) \quad G(n, 0) &= \frac{1}{(n+1)^3} \\ ii) \quad F(n, n-1) + G(n-1, n-1) &= \frac{5}{2} \frac{(-1)^{(n-11)} ((n-1)!)^2}{n (2n)!} \end{aligned} \right\} . \quad (5.25)$$

Außerdem gilt auch die Gleichung

$$\lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} F(n, k) = \lim_{n \rightarrow \infty} \sum_{k=0}^{n-1} \frac{1}{2} \frac{(k!)^2 (n-k-1)! (-1)^k}{(n+k+1)! (k+1)} = 0, \quad (5.26)$$

auf deren Beweis wir nicht weiter eingehen wollen. Aus (5.25) und (5.26) wird unter Benutzung von (5.17) die Formel

$$\sum_{n=0}^{\infty} \frac{1}{(n+1)^3} = \frac{5}{2} \left( \sum_{n=1}^{\infty} \frac{(-1)^{(n-1)} ((n-1)!)^2}{n (2n)!} \right) \quad (5.27)$$

hergeleitet. □

### 5.3 Wilfsche Methode

Die Wilfsche Methode der Reihentransformation basiert auf Gleichung (5.28), die eine allgemeinere Form als (5.1) hat. Die Methode stellt auch tatsächlich eine Verallgemeinerung der Rechtecktransformation dar.

Die Darstellung der Wilfschen Methode in diesem Abschnitt basiert auf [Wil99, S. 155-156]. Die grundlegenden Beziehungen sind in Lemma 5.12 dargestellt und die Anwendungsformen sind in Satz 5.13 formuliert.

**Lemma 5.12** (Hauptlemma der Wilfschen Transformation). *Zwei Funktionen  $F(n, k)$  und  $G(n, k)$  sollen als erstes der Gleichung*

$$a_0(n)F(n, k) + a_1(n)F(n+1, k) = G(n, k+1) - G(n, k) \quad (5.28)$$

genügen. Außerdem sei  $N_0$  eine beliebige, jedoch feste natürliche Zahl, mit der  $F(n, k)$  und  $G(n, k)$  noch die Eigenschaften

1. für jedes  $n$  mit  $0 \leq n \leq N_0$  konvergiert die Reihe  $\sum_{k=0}^{\infty} F(n, k)$ ,
2. für jedes  $n$  mit  $0 \leq n \leq N_0$  gilt  $\lim_{k \rightarrow \infty} G(n, k) = 0$  für  $0 \leq n \leq N_0$ ,

erfüllen; so lässt sich  $\sum_{k=0}^{\infty} F(0, k)$  durch  $[G(j, 0)]_{j=0}^{N_0-1}$  und  $\sum_{k=0}^{\infty} F(N_0, k)$  gemäß der Gleichung

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{j=0}^{N_0-1} \frac{G(j, 0)}{a_1(j)A(j+1)} + \frac{1}{A(N_0)} \sum_{k=0}^{\infty} F(N_0, k) \quad (5.29)$$

darstellen, wobei die Funktion  $A(n)$  durch

$$A(n) \stackrel{\text{def}}{=} \begin{cases} i) & 1 & \text{falls } n = 0 \\ ii) & \prod_{j=0}^{n-1} \left( -\frac{a_0(j)}{a_1(j)} \right) & \text{falls } n > 0 \end{cases} \quad (5.30)$$

definiert ist.

*Beweis.* Setzen wir aufgrund der vorausgesetzten Eigenschaft 1. des Lemmas

$$f(n) \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} F(n, k) \quad (5.31)$$

und summieren unter Beachtung von (5.31) die Instanzen von (5.28) für  $k = 0 \rightarrow \infty$ , so erhalten wir

$$a_0(n)f(n) + a_1(n)f(n+1) = -G(n, 0) + \lim_{k \rightarrow \infty} G(n, k)$$

als Ergebnis, woraus in Verbindung mit der Eigenschaft 2. des Lemmas die Gleichung

$$a_0(n)f(n) + a_1(n)f(n+1) = -G(n, 0) \quad (5.32)$$

impliziert wird.

Gleichung (5.32) stellt eine Rekursionsgleichung für  $f(n)$  dar. In diesem Sinn ist Gleichung (5.29) im Wesentlichen dadurch hergeleitet, dass anhand dieser Rekursion  $f(0)$  nach  $f(N_0)$  aufgelöst wird. Im Folgenden betrachten wir einen konkreten Lösungsweg näher.

Bemerken wir zunächst, dass (5.30) zu

$$A(n) \stackrel{\text{def}}{=} \begin{cases} i) & A(0) = 1 \\ ii) & A(n+1) = -\frac{a_0(n)}{a_1(n)} A(n) \end{cases} \quad (5.33)$$

äquivalent ist.

Führen wir eine Hilfsfunktion  $h(n)$  mit der Eigenschaft

$$f(n) = A(n)h(n) \quad (5.34)$$

durch  $h(n) \stackrel{\text{def}}{=} \frac{F(n)}{A(n)}$  ein. Formen wir (5.32) nun wie folgt um

$$\begin{aligned} & a_0(n) \underbrace{f(n)}_{(5.34)} + a_1(n) \underbrace{f(n+1)}_{(5.34)} = -G(n, 0) \\ \Rightarrow & a_0(n) A(n) h(n) + a_1(n) A(n+1) h(n+1) = -G(n, 0) \\ \Rightarrow & \frac{a_0(n) A(n)}{a_1(n) A(n+1)} h(n) + h(n+1) = \frac{-G(n, 0)}{a_1(n) A(n+1)} \\ \Rightarrow & \underbrace{\frac{a_0(n) A(n)}{A(n+1)}}_{(5.33)} h(n) + h(n+1) = \frac{-G(n, 0)}{a_1(n) A(n+1)} \\ \Rightarrow & -h(n) + h(n+1) = \frac{-G(n, 0)}{a_1(n) A(n+1)}, \end{aligned}$$

d. h. es gilt

$$-h(n) + h(n+1) = \frac{-G(n, 0)}{a_1(n) A(n+1)}. \quad (5.35)$$

Summieren wir die Instanzen von (5.35) für  $n = 0 \rightarrow N_0 - 1$  und erhalten als Ergebnis

$$h(N_0) = - \sum_{j=0}^{N_0-1} \frac{G(0, 0)}{a_1(j) A(j+1)} + h(0). \quad (5.36)$$



Setzen wir nun den Ausdruck für  $h(N_0)$  aus (5.36) in (5.34) ein, so erhalten wir als Ergebnis

$$f(N_0) = A(N_0) \left( h(0) - \sum_{j=0}^{N_0-1} \frac{G(j, 0)}{a_1(j)A(j+1)} \right) . \quad (5.37)$$

Aus (5.30) i) und (5.34) folgt  $h(0) = f(0)$ . Mit der Auflösung nach  $h(0) = f(0)$  aus (5.37) und unter Beachtung von (5.31) wird (5.29) impliziert. ■

**Satz 5.13** (Wilfsche Transformation). *Der Gedankengang in Lemma 5.12 wird unter Benutzung gleicher Notationen fortgesetzt. Es seien nun die Eigenschaften 1. und 2. für alle  $n \in \mathbb{N}$  erfüllt. So gilt auch*

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{j=0}^{n-1} \frac{G(j, 0)}{a_1(j)A(j+1)} + \frac{1}{A(n)} \sum_{k=0}^{\infty} F(n, k) \quad (5.38)$$

für alle  $n \in \mathbb{N}$ . Gilt zusätzlich

$$\lim_{n \rightarrow \infty} \frac{1}{A(n)} \sum_{k=0}^{\infty} F(n, k) = 0 , \quad (5.39)$$

so gilt auch

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{j=0}^{\infty} \frac{G(j, 0)}{a_1(j)A(j+1)} . \quad (5.40)$$

*Beweis.* Offensichtlich aus Lemma 5.12, bzw. durch Grenzübergang. ■

Mit der Instanziierung  $a_0(n) \equiv -1$  und  $a_1(n) \equiv 1$  wird die Wilfsche Transformation genau die Rechtecktransformation. So stellt die Wilfsche Transformation eine allgemeinere Form der Rechtecktransformation dar.

Die Wilfsche Transformation kommt dadurch zur Anwendung, dass eine Reihe in Übereinstimmung mit  $\sum_{k=0}^{\infty} F(0, k)$  gebracht und dann durch die rechte Seite von (5.38) oder (5.40) ersetzt wird.

Durch die flexible Wahl der Funktionalgleichung und der entsprechenden  $G(n, k)$  hat die Wilfsche Methode ein höheres Anwendungspotenzial als die Markoffsche Rechtecktransformation.

Wir wollen uns im Folgenden zwei Beispiele aus [Wil99] ansehen.

*Beispiel 5.14* (Wilfsche Transformation für  $\zeta(2)$ ). [Vergl. Wil99, S. 156-157]. Für die Unifikation von  $\sum_{k=0}^{\infty} F(0, k)$  mit der Reihe in (4.30) wird der Ansatz

$$F(n, k) = \frac{k!^2}{(n+k+1)!^2} \quad (5.41)$$

herangezogen. Es gilt also

$$F(0, k) = \frac{1}{(k+1)^2} . \quad (5.42)$$

Für die Suche nach einer Rekursionsgleichung der Form (5.28) für  $F(n, k)$  ist die Anwendung des Zeilberger-Algorithmus (siehe Exkurs 5.21 und Beispiel B.5) erfolgreich. Die Rekursionsgleichung lautet

$$-(4n+2)F(n, k) + (n+1)^3 F(n, k+1) = G(n, k+1) - G(n, k) , \quad (5.43)$$

wobei

$$G(n, k) = (3n + 2k + 3)F(n, k) = \frac{(3n + 2k + 3) \cdot k!^2}{(n + k + 1)!^2} \quad (5.44)$$

ist. Die Funktionen  $F(n, k)$  und  $G(n, k)$  erfüllen alle Voraussetzungen für (5.38) in Satz 5.13. Wir gehen aber nicht auf Details des Beweisens ein. Im Folgenden führen wir zur Illustrierung der Anwendung der Wilfschen Transformation nur die Berechnung der rechten Seite von (5.38) aus.

Aus (5.44) lässt sich berechnen

$$G(n, 0) = \frac{3n + 3}{(n + 1)!^2} . \quad (5.45)$$

Aus (5.43) lässt sich ablesen

$$\left. \begin{array}{l} i) \quad a_0(n) = -(4n + 2) \\ ii) \quad a_1(n) = (n + 1)^3 \end{array} \right\} . \quad (5.46)$$

Aus (5.46) lässt sich in Bezug auf (5.30) berechnen

$$A(n) = \frac{(2n)!}{n!^4} . \quad (5.47)$$

Die Instanziierung von (5.38) durch (5.42), (5.45), (5.47) und (5.46) ii) liefert als Ergebnis die Gleichung

$$\zeta(2) = \sum_{i=0}^{n-1} \frac{3 \cdot i!^2}{(2i + 2)!} + \frac{(n!)^4}{(2n)!} \sum_{k=0}^{\infty} \frac{k!^2}{(k + n + 1)!^2} .$$

Diese Gleichung ist bis auf Indexverschiebung und Ausklammerung die gleiche wie (4.70) in Beispiel 4.27, die dann durch Grenzübergang zu (4.33) wird.

In der originalen Arbeit [Wil99, S. 157] sind die Ergebnisse in einer etwas anderen Form

$$\zeta(2) = 3 \cdot \sum_{i=1}^n \frac{1}{i^2 \binom{2i}{i}} + \frac{(n!)^4}{(2n)!} \sum_{k=0}^{\infty} \frac{k!^2}{(k + n + 1)!^2} = 3 \cdot \sum_{i=1}^{\infty} \frac{1}{i^2 \binom{2i}{i}}$$

aufgeschrieben. □

*Beispiel 5.15* (Wilfsche Methode und  $\zeta(3)$ ). [Vergl. Wil99, S. 157]. Für die Unifikation von  $\sum_{k=0}^{\infty} F(0, k)$  mit der Reihe in (5.22) wird der Ansatz

$$F(n, k) = \frac{k!^3}{(k + 2n + 1)!^3}$$

herangezogen. Für die Suche nach einer Rekursionsgleichung der Form (5.28) für  $F(n, k)$  liefert die Anwendung des Zeilberger-Algorithmus das folgende Ergebnis

$$-6(3n + 2)(3n + 1)F(n, k) - 16(2n + 1)^3(n + 1)^5 = G(n, k + 1) - G(n, k) , \quad (5.48)$$

wobei die Formel für  $G(n, k)$

$$G(n, k) = \frac{207k + 784n + 1113kn + 147k^2 + 48k^3 + 1760n^4 + 6k^4 + 2728n^3 + 2084n^2 + 594k^2n + 2214kn^2 + 448n^5 + 116 + 624n^4k + 1932n^3k + 348n^3k^2 + 792n^2k^2 + 132nk^3 + 90n^2k^3 + 9k^4n}{(k + 2 + 2n)^3} \cdot F(n, k) \quad (5.49)$$

lautet. Die Instanziierung von (5.38) in Übereinstimmung mit (5.48) und (5.49) produziert die Zwischenformel

$$G(n, 0) = \frac{(2n+1)(56n^2+80n+29)}{2(n+1)(2n+1)!^3} \quad (5.50)$$

und die Ergebnisse

$$\zeta(3) = \sum_{i=1}^n (-1)^{1+i} \frac{(56i^2-32i+5)(i-1)!^3}{4(2i-1)^2(3i)!} + \frac{(-1)^n n!^3 (2n)!^3}{(3n)!} \sum_{k=0}^{\infty} \frac{k!^3}{(k+2n+1)!^3} \quad (5.51)$$

bzw.

$$\zeta(3) = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{(56i^2-32i+5)(i-1)!^3}{4(2i-1)^2(3i)!} . \quad (5.52)$$

Die manuelle Handhabung von Formeln wie (5.49) ist unpraktisch. Daran sieht man deutlich den Vorteil bzw. die Notwendigkeit des Einsatzes eines Computeralgebrasystems.  $\square$

## 5.4 Über Gosper- und Zeilberger-Algorithmus

Gosper- und Zeilberger-Algorithmus stellen Meilensteine in der Algorithmierung bzw. Computerisierung des mathematischen Denkens dar. Die Lösung der mathematischen Probleme der symbolischen Summation und der Konstruktion von Rekursiongleichungen wird durch die beiden Algorithmen stark unterstützt. Ihre Entwicklung beinhaltet tiefgreifende mathematische Ansätze und ist an sich sehr interessant. Im Rahmen der vorliegenden Dissertation werden sie jedoch nur aus Anwendersicht betrachtet. Dazu werden zuerst die mit ihnen lösbaren Probleme identifiziert, dann werden die Verknüpfungspunkte zu den Teilaufgaben der Reihentransformation erläutert. Beispiele für die Benutzung der beiden Algorithmen als ausführbare Softwarekomponenten sind in Anhang B.2 und Anhang B.3 angegeben.

Wir wollen zuerst die Klasse von Funktionen auszeichnen, für die die beiden Algorithmen anwendbar sind.

**Definition 5.16** (Hypergeometrischer Term).

1. Eine diskrete Funktion  $F(n)$  heißt *hypergeometrischer Term*, wenn der Quotient

$$\frac{F(n+1)}{F(n)}$$

eine rationale Funktion von  $n$  repräsentiert.

2. Eine mehrstellige diskrete Funktion  $F(m_1, m_2, \dots, m_k)$  heißt *hypergeometrischer Term* in  $m_1, m_2, \dots, m_k$ , wenn für jedes  $i$  mit  $1 \leq i \leq k$  der Quotient

$$\frac{F(m_1, m_2, \dots, m_i+1, \dots, m_k)}{F(m_1, m_2, \dots, m_i, \dots, m_k)}$$

eine rationale Funktion von  $m_1, m_2, \dots, m_k$  repräsentiert.  $\square$

Rationale Funktionen, Exponentialfunktionen mit rationaler Basis sowie die Fakultätsfunktion sind typische Beispiele für hypergeometrische Terme.

**Definition 5.17** (WZ-Paar). Falls in Definition 5.4 sowohl  $F(n, k)$  als auch  $G(n, k)$  hypergeometrische Terme in  $n$  und  $k$  sind, wird ein Markoffsches Paar ein WZ-Paar.  $\square$

Der Begriff WZ-Paar ist der Ausgangspunkt für eine Fülle von interessanten Methoden für die Herleitung von Identitäten, die in [PWZ96, Kap. 7] ausgeführt sind.

**Exkurs 5.18** (Gosper-Algorithmus). Der Gosper-Algorithmus dient zur geschlossenen Summation hypergeometrischer Terme. Er stellt einen Meilenstein in der Computerisierung von Summationsverfahren dar. Entsprechend der Beschreibung in der originalen Arbeit [Gos78] sucht er für einen gegebenen hypergeometrischen Term  $a(n)$  nach einer diskreten Stammfunktion  $S(n)$  mit der Eigenschaft

$$a(n) = S(n) - S(n-1) , \quad (5.53)$$

die auch ein hypergeometrischer Term sein soll. Dabei *entscheidet er auch darüber, ob ein solches  $S(n)$  existiert.*

Es gibt viele über das Internet frei verfügbare Implementationen des Algorithmus. Er ist u. a. implizit zur Berechnung von Summen in Maple eingebettet. Eine explizite Benutzung bietet z. B. das frei verfügbare Paket HYPERREG an, in dem der Algorithmus in der Funktion Gosper verankert ist. Hierbei weist die Arbeitsweise des Algorithmus eine kleine Abweichung im Vergleich zu (5.53) auf. Für einen gegebenen hypergeometrischen Term  $a(n)$  wird nämlich nach einer diskreten Stammfunktion  $S(n)$  mit der Eigenschaft

$$a(n) = S(n+1) - S(n) \quad (5.54)$$

gesucht, die (5.1) noch näher liegt. Weitere Informationen über HYPERREG stehen in Quelle B.2 und [Gau00] zur Verfügung. Die Benutzung der Funktion Gosper des Pakets HYPERREG als eine ausführbare Softwarekomponente wird durch Beispiel B.3 illustriert.  $\square$

*Bemerkung 5.19* (Gosper-Algorithmus für Rechteck- und Dreieckstransformation). Zeilberger hat bemerkt, dass der Gosper-Algorithmus auch für

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

und

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

anwendbar ist. Dadurch kann er sowohl für die Bestimmung von  $F(n, k)$  aus  $G(n, k)$  als auch für die Bestimmung von  $G(n, k)$  aus  $F(n, k)$  in Anlehnung an (5.1) angewandt werden. In Beispiel B.3 wird die Bestimmung von  $F(n, k)$  aus  $G(n, k)$  für die Transformation in Beispiel 5.10 mit der Funktion Gosper des Pakets HYPERREG ausgeführt.  $\square$

**Exkurs 5.20** (Rekursionsformeln für Summe und Zeilberger-Verfahren). Es sei  $F(n, k)$  ein hypergeometrischer Term in beiden Variablen  $n$  und  $k$  und es sei

$$A(n) = \sum_k F(n, k)$$

definiert. Eine zentrale Aufgabe der Theorie der hypergeometrischen Funktionen ist die Suche nach einer Rekursion der Form

$$\sum_{j=0}^l \sigma_j(n) A(n-j) = 0 , \quad (5.55)$$

wobei sowohl die natürliche Zahl  $l$  als auch die rationalen Funktionen  $[\sigma_j(n)]_{j=0}^l$  zu suchen sind.

Auf dem Gosper-Algorithmus basierend hat Zeilberger für die Lösung der Aufgabe ein Verfahren entwickelt, das einen Meilenstein in der Computerisierung der Theorie der hypergeometrischen Funktionen darstellt. Zeilberger hat u. a. auch wegen dieses Verfahrens im Jahre 1998 den Steele-Preis des Amerikanischen Mathematischen Verbandes bekommen. Die Entwicklung des Verfahrens ist mit einer Reihe von Publikationen [Zei82; Zei90b; Zei90a; Zei91] verbunden. Eine systematische Darstellung der Lösungsideen (darunter auch der Gosper-Algorithmus) findet man in [PWZ96]. Die entsprechende Implementierung in Maple — das Paket EKHAD — ist über das Internet frei verfügbar (siehe Quelle B.2).  $\square$

**Exkurs 5.21** (Zeilberger-Algorithmus). Der Zeilberger-Algorithmus bildet den Kern des in Exkurs 5.20 erwähnten Verfahrens. Für einen gegebenen hypergeometrischen Term  $F(n, k)$  in beiden Variablen  $n$  und  $k$  und eine gegebene natürliche Zahl  $l$  sucht er nach einem hypergeometrischen Term  $G(n, k)$  in beiden Variablen  $n$  und  $k$  und den rationalen Funktionen  $[\sigma_j(n)]_{j=0}^l$ , so dass

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

gilt. Für  $l = 1$  sucht er nach einer Funktionalgleichung, die für die Wilfsche Transformation benutzt werden kann. Der Algorithmus ist in der Funktion `ct` des oben erwähnten Maple-Pakets EKHAD verankert. Die Anwendung von `ct` als eine ausführbare Softwarekomponente zur Konstruktion der Funktionalgleichung für die Transformation in Beispiel 5.14 wird in Beispiel B.5 angegeben.  $\square$

In der Fachliteratur (z. B. [Stö96]) wird mit dem Zeilberger-Algorithmus manchmal das Gesamtverfahren in Exkurs 5.20 gemeint. Die oben ausgeführte Abgrenzung der Begriffe basiert auf [PWZ96, S. 102] bzw. [Wil99, S. 156].

## 5.5 Diskussion

Die Markoffsche Methode entsteht wohl durch die Operationalisierung des in Kapitel 4 vorgestellten Ansatzes der Differenz und Doppelreihe. Eine gemeinsame Differenz in (5.1) lässt sich als einen Term einer Doppelreihe interpretieren, wodurch Gleichung (5.5) der Tatsache entspricht, dass für die Doppelreihe die Reihe der Zeilensummen und die Reihe der Spaltensummen den gleichen Wert haben.

Die Transformation mit Doppelreihen ist seit langem bekannt, deren Operationalisierung durch (5.1) und (5.5) wurde jedoch erst von Markoff vorgenommen. Mit Wilf und Zeilberger hat (5.1) noch mehr an Bedeutung gewonnen, wodurch außer den in diesem Kapitel vorgestellten Methoden von Zeilberger und Wilf für Reihentransformationen noch eine ganze Theorie über Gleichheit (siehe u. a. [PWZ96]) entwickelt wird. Diese Entwicklung erscheint sehr interessant.

Die Zeilberger-Dreieckstransformation lässt sich in eine zur Markoffschen und Wilfschen Methode einheitliche Anwendungsform bringen, in der die linke Adjungierte zur Unifikation mit der originalen Reihe verwendet wird. Die Entsprechung für (5.17) lautet dann

$$\sum_{k=0}^{\infty} F(0, k) = \sum_{n=0}^{\infty} \left( F(n, n) + G(n, n+1) \right). \quad (5.17')$$

Gleichung (5.38) der Wilfschen Methode lässt sich dadurch *als eine Spigot-Formel interpretieren*, dass die Wertanteile  $\frac{G(j, 0)}{a_1(j)A(j+1)}$  gemäß (5.38) aus der Reihe  $\sum_{k=0}^{\infty} F(0, k)$  extrahiert werden.

Bei dem Versuch, die Wilfsche Methode zu verallgemeinern, hatte der Autor der Idee, dass auch Beziehungen der Form (5.55) eine Anwendungsmöglichkeit für die Transformation von  $\sum_{k=0}^{\infty} F(0, k)$  haben. Die Auflösung von  $f(0)$  nach  $f(N_0)$  bleibt dabei im Prinzip noch dieselbe wie bei Beziehungen der Form (5.28). In diesem Zusammenhang ist die Tatsache interessant, dass die Existenz einer nicht trivialen Beziehung der Form (5.55) nach [PWZ96, Theorem 6.2.1, S. 105] sichergestellt ist.

Wenn das Markoff-Paar in Beispiel 5.10 in Kombination mit der Rechtecktransformation eingesetzt wird, wird die Reihe in (4.30) in sich selbst transformiert, d. h., keine Beschleunigung wurde erzielt; während der Einsatz des Paares in Kombination mit der Dreieckstransformation die Beschleunigung in Beispiel 5.10 erzeugt. Man kann somit hoffen, dass *ein Markoff-Paar mit großer Wahrscheinlichkeit eine Reihenbeschleunigung induziert*. Man muss allerdings die richtige Art der Anwendung finden.

## Kapitel 6

# Gospersche Reihentransformation

Die Gopersche Methode für Reihentransformation wurde in [Gos74] entwickelt. Die bereitgestellten Verfahren sind mächtig, aus denen sich zahlreiche interessante Anwendungsbeispiele herleiten lassen. Bei der Darstellung wird aber kein großer Wert auf einen angemessenen Grad der Formalisierung gelegt. In der originalen Arbeit heißt es: "The treatment will be informal with few, if any, proof". Dies könnte ein Grund dafür sein, dass der Methode in der Fachliteratur nicht in vollem Umfang Bedeutung beigemessen wird, in [RW95, S. 199] steht z. B.: "He obtained this series by using a refinement of Euler transformation", wenn über die Reihe (6.52) gesprochen wird.

Das Ziel dieses Kapitels ist deshalb die Formalisierung der Goperschen Methode. Mit Definition 6.7, Lemma 6.9 und Satz 6.11 werden formale Handwerkzeuge bereitgestellt, mit deren Hilfe die Durchführung einer Transformation bei einer bekannten Spaltungsfunktion nur noch eine Routinearbeit sein sollte. Denkanstöße bzw. Heuristiken für die Konstruktion von Spaltungsfunktionen sind in Schema 6.14 und Beispiel 6.15 enthalten.

Die obige Äußerung in [RW95, S. 199] betreffend stellt die Gopersche Transformation eine Verallgemeinerung sowohl für die Eulersche als auch für die Kummersche Transformation dar. In diesem Zusammenhang kann man im gewissen Sinn sagen, dass der erfolgreiche Versuch von Goper, durch zweistellige Durchschnittsbildung die Eulersche Transformation in Übereinstimmung mit der Kummerschen Transformation zu bringen, die Geburtsstunde der Methode ist, wie in Abschnitt 4.3.1 verfolgt werden kann.

Der Goperschen Methode liegen zwei Grundgedanken zugrunde, die im Folgenden näher betrachtet werden.

1. Dem ersten Gedanken nach wird die Wertverschiebung in Abschnitt 4.3 herangezogen. Charakterisieren wir hier den Gedanken nochmals verbal konkreter. Jeder Term einer Reihe wird anhand einer Spaltungsfunktion in zwei Teile zerlegt, der zweite Teil des  $n$ -ten Terms wird dann mit dem ersten Teil des  $(n + 1)$ -ten Terms kombiniert, um eine neue (Rest)Reihe zu bilden, der erste Teil des ersten Terms bleibt übrig und wird extrahiert; durch geschickte Konstruktion von Spaltungsfunktionen soll der Vorgang sooft wiederholt werden, dass die Restreihe gegen 0 strebt und die Reihe der Extrahierten schneller als die originale Reihe konvergiert. Der Gedanke wird in Abschnitt 6.1.2 operationalisiert.
2. Dem zweiten Gedanken nach wird eine starke Hervorhebung auf die *Termquotienten* anstatt auf die Terme der Reihe selbst gesetzt. Termquotienten sind im Allgemeinen einfacher als Terme und sagen dennoch mehr über die Konvergenzgeschwindigkeit einer Reihe aus. Dieser Gedanke wird in Abschnitt 6.1.3 operationalisiert.

Mit dem ersten Gedanken *weist die Gopersche Methode auch Spigot-Charakter auf*. Die Spigot-Vorgehensweise lässt sich beispielsweise durch Gleichung (6.20) deutlich erkennen, mit der die Wertanteile der Form  $s(k, j_0) a_{k, j_0} \mid k \in \mathbb{Z}_{\geq i_0}$  aus der Reihe in (6.16) nacheinander extrahiert werden. Die Extraktion von Wertanteilen geschieht hierbei auf *symbolischer Ebene*, so dass im Gegensatz zu der Extraktion auf numerischer Ebene in Kapitel 2 eine *verlustfreie unendliche Darstellung* als Ergebnis möglich ist.

Stellen wir nun die Gopersche Methode und die Methoden in Kapitel 5 gegenüber. Die Gopersche Methode beruht auch auf zweistelligen Funktionen, die durch Funktionalgleichungen abgestimmt sind (siehe u. a. die Gleichungen in Lemma 6.9). Zwei prinzipielle Unterschiede fallen u. a. jedoch auf. Erstens kann bei der Goperschen Methode die Konstruktion dieser Funktionen auch *schrittweise* erfolgen. Zweitens genießt die Konstruktion noch keine so starke Computerunterstützung, wie es für die Methoden in Kapitel 5 der Fall ist.

Die Gopersche Methode wurde ursprünglich mit  **$\mathcal{R}$ -Ausdrücken** entwickelt, wodurch der Termquotient in den Vordergrund gebracht wird.  $\mathcal{R}$ -Ausdruck und geschachtelte Summe sind zwei äquivalente Ausdrucksformen, weisen starke strukturelle Ähnlichkeiten auf und lassen sich leicht ineinander konvertieren. Dies kommt durch (6.5) deutlich zum Ausdruck. Die beiden Formen können am besten als zwei Implementierungsvarianten derselben Datenstruktur verstanden werden, wobei der Hauptunterschied in der Indizierung von Faktoren und Koeffizienten liegt.

Einerseits könnte die Gopersche Methode in die Sprache der geschachtelten Summen umformuliert werden, um den Begriff  $\mathcal{R}$ -Ausdruck zu sparen. Andererseits könnte der numerische Spigot-Ansatz in Kapitel 2 auch in der Sprache von  $\mathcal{R}$ -Ausdrücken formuliert werden, um den Begriff geschachtelte Summe zu sparen. Die beiden Ausdrucksformen existieren jedoch nebeneinander aus historischem Grund und jede hat ihren Vorteil. In der vorliegenden Dissertation wird deshalb die Vorgehensweise gewählt, dass die Gopersche Methode wie ursprünglich in der Sprache der  $\mathcal{R}$ -Ausdrücke formalisiert wird und die Beziehung bzw. die Konvertierungsmöglichkeiten zwischen geschachtelten Summen und  $\mathcal{R}$ -Ausdrücken aufgezeigt werden.

Gosper ist u. a. ein Pionier beim Einsatz von Computeralgebrasystem zur Lösung von mathematischen Problemen. Insbesondere hat er im Projekt zur Entwicklung des Computeralgebrasystems *Macsyma* mitgewirkt. Gerade bei der Entwicklung der Goperschen Reihentransformation wurde *Macsyma* zu Hilfe genommen.

## 6.1 Formalisierung der Goperschen Methode

### 6.1.1 $\mathcal{R}$ -Ausdruck

In der originalen Arbeit von Gosper wird  $\mathcal{R}$ -Ausdruck intuitiv und nur für den unendlichen Fall eingeführt. Um die — sowohl formale als auch intuitive — Handhabung von  $\mathcal{R}$ -Ausdrücken zu vereinfachen, sowie um die Beziehung zwischen  $\mathcal{R}$ -Ausdrücken und geschachtelten Summen deutlich aufzuzeigen, werden in diesem Abschnitt sowohl endliche als auch unendliche  $\mathcal{R}$ -Ausdrücke formal definiert. Die formale Konstruktion der Semantik für unendliche  $\mathcal{R}$ -Ausdrücke durch Grenzübergang wird ausgelassen, sie ist analog zu der für INE in Abschnitt 2.2.1.

Die Erweiterung des Begriffssystems hat eine Erweiterung des Notationssystems zur Folge. Außerdem gibt es eine leichte Modifikation in der Notation im Vergleich zu der originalen Arbeit [Gos74]. In der vorliegenden Dissertation wird  $b_i \overset{\infty}{\mathcal{R}} r_i$  anstelle von  $b_i \underset{i \geq i_1}{\mathcal{R}} r_i$  benutzt.

Führen wir zuerst den Begriff  $\mathcal{R}$ -Ausdruck ein.



**Definition 6.1** (Goperscher  $\mathcal{R}$ -Ausdruck). Wir definieren im Folgenden die einzelnen Fälle des  $\mathcal{R}$ -Ausdrucks:

1. **Allgemeiner endlicher  $\mathcal{R}$ -Ausdruck.** Die Semantik des Ausdrucks  $b_i \overset{i_2}{\mathcal{R}}_{i=i_1} r_i$  ist definiert durch

$$b_i \overset{i_2}{\mathcal{R}}_{i=i_1} r_i \stackrel{\text{def}}{=} b_{i_1} + r_{i_1}(b_{i_1+1} + r_{i_1+1}(b_{i_1+2} + r_{i_1+2}(b_{i_1+3} + \dots + r_{i_2-1}(b_{i_2} + r_{i_2}(b_{i_2+1})) \dots))) . \quad (6.1)$$

2. **Endlicher Implizit-1- $\mathcal{R}$ -Ausdruck.** Die Semantik des Ausdrucks  $\overset{i_2}{\mathcal{R}}_{i=i_1} r_i$  ist definiert durch

$$\overset{i_2}{\mathcal{R}}_{i=i_1} r_i \stackrel{\text{def}}{=} 1 + r_{i_1}(1 + r_{i_1+1}(1 + r_{i_1+2}(1 + \dots + r_{i_2-1}(1 + r_{i_2}) \dots))) . \quad (6.1a)$$

3. **Allgemeiner unendlicher  $\mathcal{R}$ -Ausdruck.** Die Semantik des Ausdrucks  $b_i \overset{\infty}{\mathcal{R}}_{i=i_1} r_i$  ist definiert durch

$$b_i \overset{\infty}{\mathcal{R}}_{i=i_1} r_i \stackrel{\text{def}}{=} b_{i_1} + r_{i_1}(b_{i_1+1} + r_{i_1+1}(b_{i_1+2} + r_{i_1+2}(b_{i_1+3} + \dots \quad (6.2)$$

(vergl. [Gos74, S. 8]).

4. **Unendlicher Implizit-1- $\mathcal{R}$ -Ausdruck.** Die Semantik des Ausdrucks  $\overset{\infty}{\mathcal{R}}_{i=i_1} r_i$  ist definiert durch

$$\overset{\infty}{\mathcal{R}}_{i=i_1} r_i \stackrel{\text{def}}{=} 1 + r_{i_1}(1 + r_{i_1+1}(1 + r_{i_1+2}(1 + \dots \quad (6.2a)$$

(vergl. [Gos74, S. 7]).

□

*Bemerkung 6.2* (Gültigkeitsbereich der Indexvariablen des  $\mathcal{R}$ -Ausdrucks). Im Gegensatz zum  $\Sigma$ - und  $\Pi$ -Ausdruck umfasst der Gültigkeitsbereich der Indexvariablen des  $\mathcal{R}$ -Ausdrucks auch den Bereich links vor dem  $\mathcal{R}$ -Zeichen. Infolgedessen hängt die Semantik von  $B \overset{\infty}{\mathcal{R}}_{i=i_1} r_i$  entscheidend davon ab, ob  $B$  ein Ausdruck der Variablen  $i$  ist. Diese Abhängigkeit bestimmt, ob  $B \overset{\infty}{\mathcal{R}}_{i=i_1} r_i$  mit (6.2) oder (6.2a) zu interpretieren ist. Um Mehrdeutigkeit zu vermeiden, sei es empfohlen, Klammern ausreichend einzusetzen. □

Als nächstes werden einige Möglichkeiten der Umformung von  $\mathcal{R}$ -Ausdrücken in andere mathematische Ausdrücke betrachtet.

**Lemma 6.3** (Umformung von  $\mathcal{R}$ -Ausdrücken). Für die Umformung von  $\mathcal{R}$ -Ausdrücken gelten folgende Gleichungen:

$$b_i \overset{i_2}{\mathcal{R}}_{i=i_1} r_i = b_{i_1} + \sum_{i=i_1}^{i_2} \left( \prod_{j=i_1}^i r_j \right) b_{i+1} , \quad (6.3)$$

$$b_i \overset{\infty}{\mathcal{R}}_{i=i_1} r_i = b_{i_1} + \sum_{i=i_1}^{\infty} \left( \prod_{j=i_1}^i r_j \right) b_{i+1} , \quad (6.4)$$

$$\begin{aligned}
b_i \underset{i=i_1}{\overset{i_2}{\mathcal{R}}} r_i &= b_{i_1} + \underbrace{r_{i_1}}_{\Pi} \cdot b_{i_1+1} + \underbrace{(r_{i_1} \cdot r_{i_1+1})}_{\Pi} \cdot b_{i_1+2} + \dots + \underbrace{(r_{i_1} \cdot r_{i_1+1} \cdots r_{i_2})}_{\Pi} \cdot b_{i_2+1} \\
&\quad \underbrace{\hspace{15em}}_{\Sigma} \\
&= b_{i_1} + \sum_{i=0}^{i_2-i_1} \left( \prod_{j=0}^i r_{i_1+j} \right) b_{i_1+i+1} \stackrel{i \xrightarrow{\text{shft}} i_1+1, j \xrightarrow{\text{shft}} i_1+1}{=} b_{i_1} + \sum_{i=i_1+1}^{i_2+1} \left( \prod_{j=i_1+1}^i r_{j-1} \right) b_i \\
&\quad \stackrel{(2.13): k \xrightarrow{\text{sub}} i_1, c_i \xrightarrow{\text{sub}} b_i, \frac{a_j}{b_j} \xrightarrow{\text{sub}} r_{j-1}}{=} b_{i_1} + \text{RNE} \left( \left[ (r_{i-1}, b_i) \right]_{i=i_1+1}^{i_2+1} \right)
\end{aligned}$$

Herleitung 6.1: Umformung eines endlichen  $\mathcal{R}$ -Ausdrucks

$$b_i \underset{i=i_1}{\overset{i_2}{\mathcal{R}}} r_i = b_{i_1} + \text{RNE} \left( \left[ (r_{i-1}, b_i) \right]_{i=i_1+1}^{i_2+1} \right) \quad (6.5)$$

und

$$b_i \overset{\infty}{\mathcal{R}} r_i = b_{i_1} + \text{RNE} \left( \left[ (r_{i-1}, b_i) \right]_{i=i_1+1}^{\infty} \right) . \quad (6.6)$$

*Beweis.* Der Beweis von (6.3) und (6.5) ist in Herleitung 6.1 enthalten. Durch Grenzübergang werden (6.4) und (6.6) entsprechend aus (6.3) und (6.5) impliziert. ■

**Korrolar 6.4** (Umformung von Implizit-1- $\mathcal{R}$ -Ausdrücken). *Für die Umformung von Implizit-1- $\mathcal{R}$ -Ausdrücken gelten folgende Gleichungen:*

$$\underset{i=i_1}{\overset{i_2}{\mathcal{R}}} r_i = 1 + \sum_{i=i_1}^{i_2} \prod_{j=i_1}^i r_j , \quad (6.7)$$

$$\overset{\infty}{\mathcal{R}} r_i = 1 + \sum_{i=i_1}^{\infty} \prod_{j=i_1}^i r_j , \quad (6.8)$$

$$\underset{i=i_1}{\overset{i_2}{\mathcal{R}}} r_i = 1 + \text{RNE} \left( \left[ (r_{i-1}, 1) \right]_{i=i_1+1}^{i_2+1} \right) \quad (6.9)$$

und

$$\overset{\infty}{\mathcal{R}} r_i = 1 + \text{RNE} \left( \left[ (r_i, 1) \right]_{i=i_1}^{\infty} \right) . \quad (6.10)$$

*Beweis.* Die Gleichungen in Korrolar 6.4 resultieren aus den entsprechenden Gleichungen in Lemma 6.3, wenn alle  $b_i$  durch 1 ersetzt werden. ■

Sehen wir uns nun einige Möglichkeiten der Überführung anderer mathematischer Ausdrücke in  $\mathcal{R}$ -Ausdrücke an.

**Lemma 6.5** (Überführung anderer mathematischer Ausdrücke in  $\mathcal{R}$ -Ausdrücke). *Es gelten folgende Umformungen:*

$$\sum_{i=i_1}^{i_2} a_i = a_{i_1} \left( \mathcal{R}_{i=0}^{i_2-i_1-1} \frac{a_{i_1+i+1}}{a_{i_1+i}} \right), \quad (6.11)$$

$$\sum_{i=i_1}^{\infty} a_i = a_{i_1} \left( \mathcal{R}_{i=0}^{\infty} \frac{a_{i_1+i+1}}{a_{i_1+i}} \right), \quad (6.12)$$

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{k+n} \right) = c_i \mathcal{R}_{i=k}^{k+n-1} \frac{a_{i+1}}{b_{i+1}} \quad (6.13)$$

und

$$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{\infty} \right) = c_i \mathcal{R}_{i=k}^{\infty} \frac{a_{i+1}}{b_{i+1}}. \quad (6.14)$$

*Beweis.* Gleichung (6.11) wird dadurch bewiesen, dass der  $\mathcal{R}$ -Ausdruck der rechten Seite durch Anwendung von (6.7) (oder gar (6.1a)) entwickelt wird, und anschließend die elementaren Regeln der Arithmetik herangezogen werden. Gleichung (6.12) folgt aus (6.11) durch Grenzübergang. Gleichung (6.13) wird aus (6.5) durch Vertauschung der Seiten, geeignete Instanziierung und geeignete Indexverschiebung hergeleitet. Gleichung (6.14) resultiert aus Gleichung (6.13) durch Grenzübergang. ■

**Korrolar 6.6** (Allgemeiner und Implizit-1- $\mathcal{R}$ -Ausdruck).

$$b_i \mathcal{R}_{i=i_1}^{\infty} r_i = b_{i_1} \left( \mathcal{R}_{i=i_1}^{\infty} \frac{b_{i+1}}{b_i} r_i \right) \quad (6.15)$$

*Beweis.* Die linke Seite von (6.15) wird unter Benutzung von (6.4) und anschließend durch Anwendung von (6.12) in die rechte Seite überführt. ■

### 6.1.2 Spaltungsfunktion und schrittweise Extraktion von Wertanteilen aus einer Reihe

Die Extraktion von Wertanteilen aus einer Reihe durch Wertverschiebung, die dem ersten Grundgedanken der Gosperschen Methode entspricht, wird in diesem Abschnitt formalisiert bzw. operationalisiert.

**Definition 6.7** (Schrittweise Extraktion von Wertanteilen aus einer Reihe mit Gosperscher Methode). Es sei eine Reihe gegeben, die durch eine geeignete Indizierung die Form

$$\sum_{j=j_0}^{\infty} a_{i_0, j} \quad (6.16)$$

einnehmen soll.

Die *Extraktion von Wertanteilen aus (6.16)* ist durch eine zweistellige *Gospersche Spaltungsfunktion*  $s(i, j) \mid i, j \in \mathbb{Z}, i \geq i_0, j \geq j_0$  parametrisiert und mit der folgenden induktiven Konstruktion der *Extraktionssumme*  $S_i \mid i \geq i_0$  und der *Restreihe*  $R_i \mid i \geq i_0$  für jeden

mit  $i$  nummerierten Schritt gleichzusetzen, wobei auf die einzelnen Terme von  $R_i$  gemäß der Gleichung

$$R_i = \sum_{j=j_0}^{\infty} a_{i,j} \quad (6.17)$$

Bezug genommen wird. Die induktive Konstruktion von  $S_i$  und  $R_i$  lautet im Einzelnen wie folgt:

1. **Induktionsanfang.** Für  $i = i_0$  werden festgelegt

$$S_{i_0} = 0 \quad (6.18)$$

und

$$R_{i_0} = \sum_{j=j_0}^{\infty} a_{i_0,j} . \quad (6.19)$$

2. **Induktionsschritt.** Es seien  $S_k$  und  $R_k = \sum_{j=j_0}^{\infty} a_{k,j}$  definiert,  $S_{k+1}$  und  $R_{k+1}$  werden nun aus  $S_k$  und  $R_k = \sum_{j=j_0}^{\infty} a_{k,j}$  gemäß Herleitung 6.2 konstruiert. Demnach lauten die

$$\begin{aligned} S_k + R_k &= S_k + \sum_{j=j_0}^{\infty} a_{k,j} = S_k + \sum_{j=j_0}^{\infty} a_{k,j} (s(k,j) + 1 - s(k,j)) \\ &= S_k + \sum_{j=j_0}^{\infty} a_{k,j} s(k,j) + \sum_{j=j_0}^{\infty} a_{k,j} (1 - s(k,j)) \\ &= S_k + s(k, j_0) a_{k, j_0} + \underbrace{\sum_{j=j_0+1}^{\infty} a_{k,j} s(k,j)}_{j \xrightarrow{\text{shift}} -1} + \sum_{j=j_0}^{\infty} a_{k,j} (1 - s(k,j)) \\ &= S_k + s(k, j_0) a_{k, j_0} + \underbrace{\sum_{j=j_0}^{\infty} a_{k, j+1} s(k, j+1)} + \sum_{j=j_0}^{\infty} a_{k,j} (1 - s(k,j)) \\ &= \underbrace{S_k + s(k, j_0) a_{k, j_0}}_{S_{k+1}} + \underbrace{\sum_{j=j_0}^{\infty} ((1 - s(k,j)) a_{k,j} + s(k, j+1) a_{k, j+1})}_{R_{k+1}} \end{aligned}$$

Herleitung 6.2: Ein Schritt der Gosperschen Extraktion

Rekursionsformeln für  $S_k$  und  $R_k$

$$S_{k+1} = S_k + s(k, j_0) a_{k, j_0} \quad (6.20)$$

bzw

$$R_{k+1} = \sum_{j=j_0}^{\infty} ((1 - s(k,j)) a_{k,j} + s(k, j+1) a_{k, j+1}) . \quad (6.21)$$

Unter Beachtung von (6.17) ist (6.21) äquivalent zu

$$a_{k+1,j} = (1 - s(k, j)) a_{k,j} + s(k, j + 1) a_{k,j+1} . \quad (6.22)$$

Die Zahl  $i_0$  wird *Restreihenanzfangsindex* und die Zahl  $d_0$  *Termanfangsindex* der Extraktion genannt.  $\square$

**Lemma 6.8** (Invarianz der Gosperschen Extraktion). *Gemäß Definition 6.7 gilt für alle  $k$*

$$S_k + R_k = R_{i_0} . \quad (6.23)$$

*Beweis.* Durch Induktion über  $k$  mit Hilfe von (6.18) und Herleitung 6.2.  $\blacksquare$

### 6.1.3 Kombinationsquotient und explizite Formel für Extraktionssumme

Der zweite Grundgedanke der Gosperschen Methode, wonach der Termquotient in den Vordergrund gebracht werden soll, wird in diesem Abschnitt formalisiert bzw. operationalisiert. Für die Operationalisierung ist die Einführung von *Kombinationsquotienten* durch Gosper wesentlich. Durch Kombinationsquotienten sind Spaltungsfunktion und Termquotienten von Restreihen sinnvollerweise verbunden. Infolgedessen wird die schrittweise Konstruktion der Spaltungsfunktion vereinfacht.

Die Bezeichnung Kombinationsquotient wird vom Autor der vorliegenden Dissertation wegen (6.26) gegeben.

**Lemma 6.9** (Kombinationsquotient und grundlegende Beziehungen der Gosperschen Methode). *Der Gedankengang in Definition 6.7 wird unter Benutzung gleicher Notationen fortgesetzt. Die Termquotienten der Restreihen werden durch*

$$r_{i,j} \stackrel{\text{def}}{=} \frac{a_{i,j+1}}{a_{i,j}} \quad (6.24)$$

bezeichnet. Außerdem wird der Kombinationsquotient

$$u_{i,j} \stackrel{\text{def}}{=} 1 - s(i, j) + r_{i,j} s(i, j + 1) . \quad (6.25)$$

eingeführt. So gelten die Beziehungen

$$a_{i+1,j} = u_{i,j} a_{i,j} \quad (6.26)$$

und

$$r_{i+1,j} = \frac{u_{i,j+1}}{u_{i,j}} r_{i,j} . \quad (6.27)$$

*Beweis.* Zum Beweis von (6.26) wird Herleitung 6.3 herangezogen.

Zum Beweis von (6.27) wird die Umformung

$$r_{i+1,j} \stackrel{(6.24)}{=} \frac{\overbrace{a_{i+1,j+1}}^{(6.26)}}{\underbrace{a_{i+1,j}}_{(6.26)}} = \frac{u_{i,j+1} a_{i,j+1}}{u_{i,j} a_{i,j}} = \frac{u_{i,j+1}}{u_{i,j}} \frac{a_{i,j+1}}{\underbrace{a_{i,j}}_{(6.24)}} = \frac{u_{i,j+1}}{u_{i,j}} r_{i,j}$$

durchgeführt.  $\blacksquare$

$$\begin{aligned}
a_{i+1,j} &\stackrel{(6.22)}{=} (1 - s(i,j)) a_{i,j} + s(i,j+1) \underbrace{a_{i,j+1}} \\
&= (1 - s(i,j)) a_{i,j} + s(i,j+1) \underbrace{\frac{a_{i,j+1}}{a_{i,j}}}_{(6.24)} a_{i,j} = (1 - s(i,j)) \underbrace{a_{i,j}} + s(i,j+1) r_{i,j} \underbrace{a_{i,j}} \\
&= (1 - s(i,j) + \underbrace{s(i,j+1) r_{i,j}}_{(6.25)}) a_{i,j} = \underbrace{(1 - s(i,j) + r_{i,j} s(i,j+1))}_{(6.25)} a_{i,j} = u_{i,j} a_{i,j} .
\end{aligned}$$

### Herleitung 6.3: Termquotienten und Kombinationsquotienten

Nun wird aus den Rekursionsgleichungen in Lemma 6.9 die explizite Formel für die Extraktionssumme hergeleitet.

**Lemma 6.10** (Explizite Formel für die Extraktionssumme). *Der gemeinsame Sachverhalt in Definition 6.7 und Lemma 6.9 wird unter Benutzung entsprechender Notationen fortgesetzt. Es gilt*

$$S_{k+1} = \sum_{i=i_0}^k s(i, j_0) a_{i, j_0} = a_{i_0, j_0} \left( s(i, j_0) \mathcal{R}_{i=i_0}^{k-1} u_{i, j_0} \right) . \quad (6.28)$$

*Beweis.* Die Gleichheit  $S_{k+1} = \sum_{i=i_0}^k s(i, j_0) a_{i, j_0}$  folgt direkt aus (6.18) und (6.20) durch Induktion. Es bleibt zu zeigen, dass

$$\sum_{i=i_0}^k s(i, j_0) a_{i, j_0} = a_{i_0, j_0} \left( s(i, j_0) \mathcal{R}_{i=i_0}^{k-1} u_{i, j_0} \right) \quad (6.29)$$

gilt.

Aus (6.26) wird

$$u_{k,j} = \frac{a_{k+1,j}}{a_{k,j}} \quad (6.30)$$

impliziert. Von (6.30) ausgehend wird  $j = j_0$  festgesetzt und  $k$  die Werte von  $i - 1$  abwärts bis zu  $i_0$  zugewiesen, demnach gilt die Umformung

$$\begin{aligned}
a_{i, j_0} &= \frac{a_{i, j_0}}{a_{i-1, j_0}} \cdot \frac{a_{i-1, j_0}}{a_{i-2, j_0}} \cdots \frac{a_{i_0+1, j_0}}{a_{i_0, j_0}} a_{i_0, j_0} = u_{i-1, j_0} \cdot u_{i-2, j_0} \cdots u_{i_0, j_0} \cdot a_{i_0, j_0} \\
&= a_{i_0, j_0} \prod_{l=i_0}^{i-1} u_{l, j_0} ,
\end{aligned}$$

oder nach Abtrennung der Zwischenschritte

$$a_{i, j_0} = a_{i_0, j_0} \prod_{l=i_0}^{i-1} u_{l, j_0} \quad (6.31)$$

für

$$i \geq i_0 + 1 . \quad (6.32)$$

Unter Benutzung von (6.31) kann nun die Umformung in Herleitung 6.4 durchgeführt werden, die durch Abtrennung der Zwischenschritte zu (6.29) wird.  $\blacksquare$

$$\begin{aligned}
\sum_{i=i_0}^k s(i, j_0) a_{i, j_0} &= s(i_0, j_0) a_{i_0, j_0} + \sum_{i=i_0+1}^k s(i, j_0) \underbrace{a_{i, j_0}}_{(6.31)} \\
&= s(i_0, j_0) \underbrace{a_{i_0, j_0}} + \sum_{i=i_0+1}^k s(i, j_0) \underbrace{a_{i_0, j_0}} \prod_{l=i_0}^{i-1} u_{l, j_0} \\
&= a_{i_0, j_0} \left( s(i_0, j_0) + \sum_{i=i_0+1}^k \left( \prod_{l=i_0}^{i-1} u_{l, j_0} \right) s(i, j_0) \right) \\
&\stackrel{i \xrightarrow{\text{shft}} -1}{=} a_{i_0, j_0} \underbrace{\left( s(i_0, j_0) + \sum_{i=i_0}^{k-1} \left( \prod_{l=i_0}^i u_{l, j_0} \right) s(i+1, j_0) \right)}_{(6.3): b_i \xrightarrow{\text{sub}} s(i, j_0), r_j \xrightarrow{\text{sub}} u_l} \\
&= a_{i_0, j_0} \left( s(i, j_0) \mathcal{R}_{i=i_0}^{k-1} u_{i, j_0} \right)
\end{aligned}$$

Herleitung 6.4: Überführung der Gosperschen Extraktionssumme in einen  $\mathcal{R}$ -Ausdruck

#### 6.1.4 Anwendungsformen und Anwendungsheuristik

Formulieren wir zuerst die häufigen Anwendungsformen der Gosperschen Methode in Satz 6.11 und Korrolar 6.12.

**Satz 6.11** (Grundformel der Gosperschen Reihentransformation). *Der Gedankengang in Definition 6.7 und Lemma 6.9 wird unter Benutzung gleicher Notationen fortgesetzt. Gilt  $\lim_{k \rightarrow \infty} R_k = 0$ , so gilt auch*

$$\sum_{j=j_0}^{\infty} a_{i_0, j} = a_{i_0, j_0} \left( s(i, j_0) \mathcal{R}_{i=i_0}^{\infty} u_{i, j_0} \right). \quad (6.33)$$

Insbesondere entsteht bei  $a_{i_0, j_0} = 1$  aus (6.33) die Gleichung

$$\mathcal{R}_{j=j_0}^{\infty} r_{i_0, j} = s(i, j_0) \mathcal{R}_{i=i_0}^{\infty} u_{i, j_0}. \quad (6.34)$$

*Beweis.* Ersichtlich aus (6.23), (6.28) und (6.17). ■

**Korrolar 6.12** (Grundformel der Gosperschen Reihentransformation im Ausdruck der geschachtelten Summe). *Im Anschluss an Satz 6.11 gilt auch*

$$\sum_{j=j_0}^{\infty} a_{i_0, j} = a_{i_0, j_0} \left( s(i_0, j_0) + \text{RNE} \left( \left[ (u_{i-1, j_0}, s(i, j_0)) \right]_{i=i_0+1}^{\infty} \right) \right). \quad (6.35)$$

*Beweis.* Offensichtlich aus (6.33) und (6.6). ■

*Bemerkung 6.13* (Gospersche Reihentransformation ohne  $\mathcal{R}$ -Ausdruck). Mit einer leichten Modifikation von Herleitung 6.4 kann die Gospersche Transformation wie in Lemma 6.12 ohne  $\mathcal{R}$ -Ausdruck formuliert werden. □

Die Gopersche Methode kommt dadurch zur Anwendung, dass eine Reihe in Übereinstimmung mit (6.17) gebracht und dann durch die rechte Seite von (6.33) bzw. (6.35) ersetzt wird. Das Entscheidende dabei ist die *Konstruktion einer passenden Spaltungsfunktion*  $s(i, j) \mid i, j \in \mathbb{Z}, i \geq i_0, j \geq j_0$ , mit deren Hilfe die Kombinationsquotienten  $u_{i,j}$  und anschließend die neue Reihe gebildet werden *können*. Die Spaltungsfunktion soll außerdem der Bedingung genügen, dass  $\lim_{k \rightarrow \infty} R_k = 0$  gilt. Wenn eine passende Spaltungsfunktion bekannt ist, ist die Bildung der neuen Reihe unter Benutzung von (6.25), (6.27) und (6.33) bzw. (6.35) nur noch eine Routinearbeit.

Mit Schema 6.14 und Beispiel 6.15 soll die Heuristik für einen effektiven Einsatz der Goperschen Methode vermittelt werden. Dabei wird die schrittweise Konstruktion einer passenden Spaltungsfunktion ausführlich betrachtet. Es sei jedoch zu bemerken, dass der Erfolg der Konstruktion gewisse Erfahrungen und Inspirationen voraussetzt und dennoch mitunter eine Glückssache ist.

**Schema 6.14** (Heuristische Vorgehensweise bei der Anwendung der Goperschen Methode). [Vergl. Gos74, S. 21]. Als erstes wird die zu beschleunigende Reihe in Übereinstimmung mit (6.16) gebracht. Dann erfolgt die Konstruktion einer passenden Spaltungsfunktion. Die einzelnen Schritte der Konstruktion werden im Folgenden ausgeführt, wobei die Laufvariable  $k$  den Bezug auf einen Schritt der Extraktion gemäß Definition 6.7 gibt und mit  $i_0$  initialisiert wird.

1. Schreibe den Termquotienten  $r_{k,j}$  als eine analytische Formel über  $j$  auf.
2. Wähle einen Ansatz, d. h. eine analytische Formel der Variablen  $j$  mit unbekanntem Koeffizienten, für  $s(k, j)$ .
3. Berechne den Kombinationsquotienten  $u_{k,j}$  aus  $r_{k,j}$  und  $s(k, j)$  unter Benutzung von (6.25) und bestimme die unbekanntem Koeffizienten so, dass  $u_{k,j}$  möglichst einfach ist.
4. Berechne  $r_{k+1,j}$  aus  $s(k, j)$  und  $u_{k,j}$  unter Benutzung von (6.27) analytisch.
5. Setze  $k$  auf  $k+1$  und wiederhole den Prozess in den Schritten 1 bis 4 solange, bis eine allgemeine Formel für  $r_{i,j}$  zu erraten ist.
6. Leite die allgemeinen Formeln für  $s(i, j)$  und  $u_{i,j}$  aus der allgemeinen Formel  $r_{i,j}$  unter Benutzung von (6.25) und (6.27) her.

Zum Schluss wird die Anwendung der Goperschen Methode unter Benutzung der oben konstruierten Spaltungsfunktion ausgeführt. Gleichung (6.34) wird herangezogen, um die neue Reihe zu berechnen. Außerdem ist die Bedingung  $\lim_{k \rightarrow \infty} R_k = 0$  zu überprüfen.  $\square$

Schema 6.14 wird durch Beispiel 6.15 konkretisiert.

*Beispiel 6.15* (Anwendung der Goperschen Methode zur Verbesserung der Konvergenz von  $\zeta(2)$  und schrittweise Konstruktion der Spaltungsfunktion). [Vergl. Gos74, S. 18-20]. Die Reihe für  $\zeta(2)$  in (4.30) ist mit der Goperschen Methode zu transformieren. Als erstes wird die Reihe in Form eines  $\mathcal{R}$ -Ausdrucks aufgeschrieben:

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \mathcal{R}_{j=1}^{\infty} \frac{j^2}{(j+1)^2}. \quad (6.36)$$

Die Reihe aus (6.36) wird in Übereinstimmung mit (6.16) gebracht, wodurch  $j_0 = 1$  impliziert wird. Außerdem wird auch  $i_0$  auf 1 gesetzt. Die einzelnen Schritte für die Konstruktion der Spaltungsfunktion sind im Folgenden ausgeführt:



1. Es gilt für den Termquotienten der originalen Reihe

$$r_{1,j} = \frac{j^2}{(j+1)^2} \cdot \quad (6.37)$$

2. Für die Zerlegung der Terme der originalen Reihe wird der Ansatz

$$s(1, j) = a j + b \quad (6.38)$$

gewählt.

3. Aus (6.38) und (6.37) wird unter Benutzung von (6.25)

$$u_{1,j} = \frac{(1-a)n^2 + (2-a-2b)n + (1-b)}{(j+1)^2} \quad (6.39)$$

berechnet. Um die Formel für  $u_{1,j}$  möglichst einfach zu machen, sollten  $a$  und  $b$  so gewählt werden, dass die Koeffizienten für  $n$  und  $n^2$  im Zähler von (6.39), d. h.  $1-a$  und  $2-a-2b$ , gleich 0 sind. Daraus werden  $a = 1$  und  $b = \frac{1}{2}$  berechnet und (6.39) wird zu

$$u_{1,j} = \frac{1}{2(j+1)^2} \cdot \quad (6.40)$$

4. Aus (6.37) und (6.40) wird unter Benutzung von (6.27)

$$r_{2,j} = \frac{j^2}{(j+2)^2} \quad (6.41)$$

berechnet.

5. Von (6.41) ausgehend wird der Prozess in den Schritten 1 bis 4 wiederholt, wobei für  $s(2, j)$  der gleiche Ansatz (6.38) benutzt wird. Als Ergebnisse werden die Gleichungen

$$u_{2,j} = \frac{4}{3(j+2)^2} \quad (6.42)$$

und

$$r_{3,j} = \frac{j^2}{(j+3)^2} \quad (6.43)$$

hergeleitet. Aus (6.37), (6.41) und (6.43) gelangt man intuitiv zur Vermutung

$$r_{i,j} = \frac{j^2}{(j+i)^2} \cdot \quad (6.44)$$

6. Der Prozess in den Schritten 1 bis 4 wird nun von (6.44) ausgehend ausgeführt. Als Ergebnis werden die Gleichungen

$$s(i, j) = \frac{2j + 3i - 2}{2(2i - 1)} \quad (6.45)$$

und

$$u_{i,j} = \frac{i^3}{2(2i - 1)(j+i)^2} \quad (6.46)$$

hergeleitet, welche in Kombination mit (6.44) die Rekursionsgleichungen (6.25) und (6.27) tatsächlich erfüllen.

Nun erfolgt die Berechnung der neuen Reihe aus der oben konstruierten Spaltungsfunktion, d. h. die eigentliche Anwendung der Goperschen Methode bei einer bekannten Spaltungsfunktion. Aus (6.45) und (6.46) wird unter Benutzung von (6.34) die Gleichung

$$\zeta(2) = \mathcal{R}_{j=1}^{\infty} \frac{j^2}{(j+1)^2} = \frac{3i}{2(2i-1)} \mathcal{R}_{i=1}^{\infty} \frac{i^3}{2(2i-1)(1+i)^2} \quad (6.47)$$

impliziert. Der Beweis der Bedingung  $\lim_{i \rightarrow \infty} R_i = 0$  wird dabei ausgelassen. Eine vereinfachte Form von (6.47) ist

$$\zeta(2) = \frac{3}{2k} \mathcal{R}_{k=1}^{\infty} \frac{1}{4} \frac{2k}{(2k+1)}, \quad (6.48)$$

die im Wesentlichen durch Anwendung von (6.4) (in beiden Richtungen) aus (6.47) gewonnen werden kann.  $\square$

## 6.2 Anwendungsbeispiele

Um die Anwendungsvielfalt der Goperschen Methode zu illustrieren, werden in diesem Abschnitt einige Beispiele aus [Gos74] mit Nachrechnung und Überarbeitung ausgeführt. Übersichtlichkeit und leichte Nachvollziehbarkeit sind angestrebt. Die notwendigen symbolischen Berechnungen werden mit Hilfe von Maple durchgeführt. Unwesentliche Details der Berechnung werden ausgelassen.

In Beispiel 6.16 werden die Routineschritte von einem Ansatz zur entsprechenden Ergebnisreihe kenntlich gemacht. In anderen Beispielen werden nur der Ansatz und die Ergebnisreihe angegeben. Die Routineschritte vom Ansatz zum Ergebnis erfolgen in allen Beispielen einheitlich nach dem gleichen Schema.

Die Überprüfungen der Bedingung  $\lim_{k \rightarrow \infty} R_k = 0$  im Satz 6.11 für die Beispiele sind wichtig, werden hier auch wie in der originalen Arbeit [Gos74] ausgelassen. Der Autor vermutet, dass diese Überprüfungen analog zu Beweis B.1 in Anhang B auszuführen seien.

### 6.2.1 Reihen für $\pi$

*Beispiel 6.16* (Eulersche Transformation als ein spezieller Fall der Goperschen Methode und die Gregory-Leibniz-Euler Reihe für  $\pi$ ). Um die Reihe für  $\frac{\pi}{4}$  in (3.32) zu transformieren, wird der Ansatz

$$i_0 = 1, j_0 = 1, r_{i,j} = -\frac{2j-1}{2(j+i)-1}, s(i,j) = \frac{1}{2}, u_{i,j} = \frac{i}{2(j+i)-1} \quad (6.49)$$

eingesetzt.

1. **Überprüfung der grundlegenden Beziehung der Goperschen Methode.** Der Wert von  $r_{i_0,j}$  ist tatsächlich der Termquotient der Reihe in (3.32) beim Index  $j$ . Die Gleichungen (6.25) und (6.27) sind durch  $r_{i_0,j}$  und  $s(i,j)$  erfüllt.
2. **Ergebnisreihe.** Die Ergebnisreihe lässt sich aus (6.49) unter Benutzung von (6.33) bzw. (6.35) zu

$$\frac{\pi}{4} = \frac{1}{2} \mathcal{R}_{i=1}^{\infty} \frac{i}{2i+1} = \frac{1}{2} \left( 1 + \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{j}{2j+1} \right) \quad (6.50)$$

berechnen. Die Äquivalenz zwischen (6.50) und (1.8) ist ersichtlich.

Die Transformation in diesem Beispiel ist ähnlich zu der in Beispiel 4.20 und entspricht der gewöhnlichen Anwendungsform der Eulerschen Transformation in Korollar 4.16.  $\square$

*Beispiel 6.17* (Eine Gospersche Reihe für  $\pi$  mit mehr als einer Dezimalziffer per Term). [Vergl. Gos74, S. 30-32]. Aus (3.32) wird mit dem Ansatz

$$i_0 = 0, j_0 = 0, r_{i,j} = -\frac{(2j+i+1)(2j+i+2)}{(2j+3i+2)(2j+3i+3)}, s(i,j) = \frac{2j+5i+3}{4j+6i+4},$$

$$u_{i,j} = \frac{(i+1)(2i+1)(2j+i+1)}{(2j+3i+2)(2j+3i+3)(2j+3i+4)} \quad (6.51)$$

durch Anwendung der Gosperschen Methode die Gleichung

$$\frac{\pi}{4} = \frac{5i+3}{6i+4} \mathcal{R}_{i=0}^{\infty} \frac{(i+1)^2(2i+1)}{(3i+2)(3i+3)(3i+4)} \quad (6.52)$$

hergeleitet, die durch Anwendung von (6.4) in

$$\frac{\pi}{4} = \frac{5i+3}{4} \mathcal{R}_{i=0}^{\infty} \frac{(2i+1)(i+1)}{(3i+4)(3i+5)} \frac{1}{3} = \frac{5i+3}{4} \mathcal{R}_{i=0}^{\infty} \frac{(i+\frac{1}{2})(i+1)}{(i+\frac{4}{3})(i+\frac{5}{3})} \frac{2}{27} = \sum_{i=0}^{\infty} \frac{(5i+3)(2i)!i!}{2^{i+1}(3i+2)!} \quad (6.53)$$

umgeformt werden kann.  $\square$

### 6.2.2 Reihen für $\zeta(2)$

*Beispiel 6.18* (Eine alternierende Gospersche Reihe für  $\zeta(2)$  mit 2 bits per Term). [Vergl. Gos74, S. 74]. Aus (4.30) wird mit dem Ansatz

$$i_0 = 0, j_0 = 1, r_{i,j} = \frac{(j+i)(j+2i)}{(j+2i+1)^2},$$

$$s(i,j) = \frac{(j+3i+2)(j+i)(j+2i)}{(i+1)(j+2i+1)^2}, u_{i,j} = -\frac{(j+2i)(j+i)(i+1)}{(j+2+2i)^2(j+2i+1)} \quad (6.54)$$

durch Anwendung der Gosperschen Methode die Gleichung

$$\zeta(2) = \frac{1}{4} \frac{10i+7}{i+1} \mathcal{R}_{i=0}^{\infty} - \frac{1}{2} \frac{(i+1)(2i+1)}{(2i+3)^2} \quad (6.55)$$

hergeleitet, die durch Anwendung von (6.4) in

$$\zeta(2) = \frac{10i-3}{4i(2i-1)} \mathcal{R}_{i=1}^{\infty} - \frac{i}{2(2i+1)} \quad (6.56)$$

umgeformt werden kann.  $\square$

*Beispiel 6.19* (Eine positive Gospersche Reihe für  $\zeta(2)$  mit 6 bits per Term). [Vergl. Gos74, S. 49]. Aus (4.30) wird mit dem Ansatz

$$i_0 = 1, j_0 = 0, r_{i,j} = \frac{(j+i)^2}{(j+2i)^2}, s(i,j) = 1 + \frac{(2j+5i)(j+i)^2}{2(2i-1)(j+2i)^2},$$

$$u_{i,j} = \frac{i^3(j+i)^3}{2(2i-1)(j+2i)^2(j+2i+1)^2} \quad (6.57)$$

durch Anwendung der Gosperschen Methode die Gleichung

$$\zeta(2) = \frac{21i-8}{8(2i-1)} \mathcal{R}_{i=1}^{\infty} \frac{i^3}{8(2i-1)(2i+1)^2} \quad (6.58)$$

hergeleitet, die durch Anwendung von (6.4) in

$$\zeta(2) = \frac{21i-8}{8} \mathcal{R}_{i=1}^{\infty} \frac{i^3}{8(2i+1)^3} = \frac{1}{8} \left( \sum_{i=0}^{\infty} \frac{(21i+13)i!^6}{(2i+1)!^3} \right) \quad (6.59)$$

umgeformt werden kann.  $\square$

### 6.2.3 Reihen für $\zeta(3)$

*Beispiel 6.20* (Eine alternierende Gospersche Reihe für  $\zeta(3)$  mit 2 bits per Term). [Vergl. Gos74, S. 36-38]. Aus (5.22) wird mit dem Ansatz

$$i_0 = 0, j_0 = 1, r_{i,j} = \frac{j(j+i)^2}{(j+i+1)^2(j+2i+1)}$$

$$s(i,j) = 1 + \frac{(j+i)^2}{2(i+1)(j+2i+1)}, u_{i,j} = \frac{(i+1)^2(j+i)^2}{(j+2i+2)(j+i+1)^2(j+2i+1)} \quad (6.60)$$

durch Anwendung der Gosperschen Methode die Gleichung

$$\zeta(3) = \frac{5}{4} \mathcal{R}_{i=1}^{\infty} - \frac{1}{2} \frac{(i+1)^3}{(2i+3)(i+2)^2} \quad (6.61)$$

hergeleitet, die durch Anwendung von (6.4) in

$$\zeta(3) = \frac{5}{4i^2} \mathcal{R}_{i=1}^{\infty} \frac{-i}{2(2i+1)} = \frac{5}{2} \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i^3 \binom{2i}{i}} \quad (6.62)$$

umgeformt werden kann.

Durch einfache Umformung lässt sich (6.62) in (5.27) überführen. Wir kommen hier durch einen anderen Weg zum gleichen Ergebnis wie in Beispiel 5.26.  $\square$

*Beispiel 6.21* (Eine positive Gospersche Reihe für  $\zeta(3)$  mit 4 bits per Term). [Vergl. Gos74, S. 58]. Aus (5.22) wird mit dem Ansatz

$$i_0 = 0, j_0 = 1, r_{i,j} = \frac{j^2(j+i)}{(j+i+1)(j+2i+1)^2}$$

$$s(i,j) = \frac{(i+1)(j+3i+2)}{4(j+2i+1)^2} + \frac{j+3i+1}{2(2i+1)}, u_{i,j} = \frac{(i+1)^4(j+i)}{(j+2+2i)^2(j+2i+1)^2(j+i+1)} \quad (6.63)$$

durch Anwendung der Gosperschen Methode die Gleichung

$$\zeta(3) = \frac{1}{16} \frac{19+30i}{1+2i} \mathcal{R}_{i=0}^{\infty} \frac{1}{4} \frac{(i+1)^3}{(2i+3)^2(i+2)} \quad (6.64)$$

hergeleitet, die durch Anwendung von (6.4) in

$$\zeta(3) = \frac{30i+19}{16(i+1)(2i+1)} \mathcal{R}_{i=0}^{\infty} \frac{(i+1)^2}{4(2i+3)^2} = \sum_{i=1}^{\infty} \frac{30i-11}{4(2i-1)i^3 \binom{2i}{i}^2} \quad (6.65)$$

umgeformt werden kann.  $\square$

## Teil III

# Anwendung und offene Probleme

# Kapitel 7

## Wurzelberechnung

Der Gegenstand dieses Kapitels ist die Anwendung des Spigot-Ansatzes auf die Wurzelberechnung. Der Inhalt wird aus [Do01] übernommen. Die Darstellung erfolgt nun mit den in Kapitel 2 bereitgestellten formalen Grundlagen und wird dadurch kürzer, exakter und systematischer. Die Beweisführung wird hinsichtlich der Anschaulichkeit und der Strukturiertheit noch verbessert. Ein Fehler in [Do01, Lemma 3.2] wird kenntlich gemacht und behoben. Die Auswirkung des Fehlers wird auch diskutiert.

Alle Wurzeln der Form  $\sqrt[k]{\frac{r}{s}} \mid r, s \in \mathbb{N} \setminus \{0\}$  lassen sich mit der Methode in diesem Kapitel zuerst effektiv über die Zugehörigkeit in  $\mathbb{Q}$  entscheiden. Falls eine Wurzel eine rationale Zahl ist, wird sie als rationaler Bruch exakt berechnet. Falls sie eine irrationale Zahl ist, wird der Spigot-Algorithmus angewandt, wodurch eine Berechnung der Wurzel zu einer Basis  $g$  mit beliebiger Genauigkeit und unter alleiniger Benutzung von Integer-Arithmetik möglich ist.

Der Anwendung des Spigot-Ansatzes liegt die klassische Idee<sup>1</sup> der Wurzelberechnung zugrunde, die durch die Gleichung

$$\sqrt[k]{\frac{r}{s}} = \frac{p}{q} \left(1 - \frac{u}{v}\right)^{-\frac{1}{k}} \quad (7.1)$$

zum Ausdruck kommt. Dabei geht (7.1) auf die charakteristische Funktionalgleichung (3.34) für Potenz und Wurzel zurück. Um mit (7.1) leichter zu operationalisieren, wird Definition 7.1 ausgeführt.

**Definition 7.1** (Grundgleichung für Wurzelberechnung). Eine Gleichung der Form (7.1) nennen wir eine Grundgleichung für die Berechnung der Wurzel  $\sqrt[k]{\frac{r}{s}}$ . Die Brüche  $\frac{p}{q}$  und  $\frac{u}{v}$  heißen der *Hilfsfaktor* bzw. der *Abweichungsradius*.  $\square$

Zur Kombination des Spigot-Ansatzes mit der klassischen Idee der Wurzelberechnung erinnere man sich an das komplexe Beispiel der Berechnung von  $\sqrt{2}$  in Abschnitt 2.5 bzw. an Schema 2.83. Dementsprechend lässt sich die Berechnung einer Wurzel auf die Herleitung einer INE für die Wurzel und die Konstruktion der entsprechenden Spigot-Schranken und Restschranken zurückführen. Entsprechend der in diesem Kapitel vorgestellten Methode der Wurzelberechnung erfolgt die Herleitung der INE über eine Grundgleichung der Form (7.1). Dadurch wird die Wurzelberechnung im Wesentlichen auf die Überführung der rechten Seite von (7.1) in eine INE und die Konstruktion von Restschranken und Spigot-Schranken für diese INE zurückgeführt. Die Teil-

---

<sup>1</sup>findet man z.B. in [Kno96, S. 266] oder [Fab10, S. 106-108]

laufgaben werden entsprechend in Abschnitt 7.2 und Abschnitt 7.1 behandelt. In Abschnitt 7.3 sind numerische Beispiele und Diskussion ausgeführt.

**7.1 Eigenschaften des Ausdrucks  $\frac{p}{q}(1 - \frac{u}{v})^{\frac{-1}{k}}$  mit  $0 < u < v \in \mathbb{N}$**

**Lemma 7.2** (ine-Darstellung für  $\frac{p}{q}(1 - \frac{u}{v})^{\frac{-1}{k}}$  mit  $0 < u < v \in \mathbb{N}$ ). *Es seien  $p, q \in \mathbb{N}$  und  $0 < u < v \in \mathbb{N}$ . Für  $\frac{p}{q}(1 - \frac{u}{v})^{\frac{-1}{k}}$  gibt es die ine-Darstellung*

$$\frac{p}{q}(1 - \frac{u}{v})^{\frac{-1}{k}} = 0 + \text{RNE} \left( \left[ \left( \frac{p}{q}, 1 \right) \right]_{i=1}^1 \overset{+}{\text{seq}} \left[ \left( \frac{u(1+k(i-2))}{k v(i-1)}, 1 \right) \right]_{i=2}^{\infty} \right) . \quad (7.2)$$

*In Anlehnung an die Kompaktnotation (2.5) mit dem Anfangsindex Null ist diese INE wie folgt charakterisiert:*

1. Die Koeffizienten sind durch

$$c_0 = 0, \quad c_n = 1 \quad \forall n \geq 1 \quad (7.2a)$$

gegeben.

2. Das Faktorsystem ist

$$\left[ \frac{p}{q} \right]_{i=1}^1 \overset{+}{\text{seq}} \left[ \frac{u(1+k(i-2))}{k v(i-1)} \right]_{i=2}^{\infty} \quad (7.2b)$$

und lässt sich wie folgt berechnen:

$$\frac{a_1}{b_1} = \frac{p}{q}, \quad \frac{a_n}{b_n} = \frac{u(1+k(n-2))}{k v(n-1)} \quad \forall n \geq 2 . \quad (7.2c)$$

*Beweis.* Die Binomialreihe (findet man u. a. in Lemma 3.10) für  $(1 - \frac{u}{v})^{\frac{-1}{k}}$  wird mit (2.77a) zu einer INE *INE* gemacht. Die INE für  $\frac{p}{q}(1 - \frac{u}{v})^{\frac{-1}{k}}$  ist  $(0, \frac{p}{q}) \overset{+}{\text{head}} \text{INE}$ . ■

Eine INE der Form (7.2) wird auch *eine BINE für die Wurzelberechnung* genannt. Die Abkürzung BINE deutet eine INE an, die sich aus einer Binomialreihe herleiten lässt.

**Lemma 7.3** (Restschranke für (7.2b)). *Das Faktorsystem (7.2b) hat für  $n \geq 2$  die folgende Restschranke*

$$p + \frac{p}{q} \frac{u}{(1 - \frac{u}{v})^2} \left( 1 + (k-1) \frac{u}{v} \right) , \quad (7.3)$$

*die von dem Grad der Approximation  $n$  unabhängig ist.*

*Beweis.* Wir benutzen Lemma 2.61 als Startpunkt und bauen Herleitung 7.1 auf. ■

**Satz 7.4** (Spigot-Schranke für (7.2b)). *Die zweistellige, jedoch von dem zweiten Argument unabhängige Funktion*

$$f(m, n) \stackrel{\text{def}}{=} \begin{cases} i) & p + \frac{p}{q} \frac{u}{(1 - \frac{u}{v})^2} \left( 1 + (k-1) \frac{u}{v} \right) & \text{falls } m = 1 \\ ii) & \frac{u}{(1 - \frac{u}{v})^2} \cdot \left( \left( (m-2) - (m-3) \frac{u}{v} \right) \cdot k + \left( 1 - \frac{u}{v} \right) \right) & \text{falls } m \geq 2 \end{cases}$$

*stellt eine Spigot-Schranke für (7.2b) dar.*

$$\begin{aligned}
a_1 + \sum_{i=2}^n a_i \prod_{j=1}^{i-1} \frac{a_j}{b_j} &< a_1 + \sum_{i=2}^{\infty} a_i \prod_{j=1}^{i-1} \frac{a_j}{b_j} = a_1 + \sum_{i=2}^{\infty} a_i \frac{a_1}{b_1} \prod_{j=2}^{i-1} \frac{a_j}{b_j} \stackrel{(a)}{\leq} p + \sum_{i=2}^{\infty} a_i \frac{p}{q} \prod_{j=2}^{i-1} \left(\frac{u}{v}\right) \\
&= p + \sum_{i=2}^{\infty} \underbrace{a_i}_{(7.2c)} \frac{p}{q} \left(\frac{u}{v}\right)^{i-2} = p + \sum_{i=2}^{\infty} u(1+k(i-2)) \frac{p}{q} \left(\frac{u}{v}\right)^{i-2} \\
&= p + \frac{p}{q} u \sum_{i=2}^{\infty} (1+k(i-2)) \left(\frac{u}{v}\right)^{i-2} = p + \frac{p}{q} u \sum_{i=0}^{\infty} (1+ki) \left(\frac{u}{v}\right)^i \\
&= p + \frac{p}{q} u \left( \underbrace{\sum_{i=0}^{\infty} \left(\frac{u}{v}\right)^i}_{(b)} + k \underbrace{\sum_{i=0}^{\infty} i \left(\frac{u}{v}\right)^i}_{(c)} \right) = p + \frac{p}{q} u \left( \frac{1}{1-\frac{u}{v}} + k \frac{\frac{u}{v}}{\left(1-\frac{u}{v}\right)^2} \right) \\
&= p + \frac{p}{q} \frac{u}{\left(1-\frac{u}{v}\right)^2} \left(1 + (k-1) \frac{u}{v}\right).
\end{aligned}$$

- 
- (a) Ersetze  $a_1 = p$ ,  $\frac{a_1}{b_1} = \frac{p}{q}$ . Außerdem  $\frac{a_j}{b_j} = \frac{a_j}{b_j} = \frac{u(1+k(j-2))}{kv(j-1)} < \frac{u}{v}$  für  $j \geq 2$  nach (7.2c).
- (b)  $\frac{1}{1-x} = \sum_{i=0}^{\infty} x^i$ , falls  $|x| < 1$ .
- (c)  $\frac{x}{(1-x)^2} = \sum_{i=0}^{\infty} ix^i$ , falls  $|x| < 1$ .

Herleitung 7.1: Eine Restschranke für (7.2b)

*Beweis.* Für den Fall  $m = 1$  entspricht der Satz genau Lemma 7.3. Für den Fall  $m \geq 2$  ist der Beweis ganz analog. Dabei wurde zur Ausführung der notwendigen symbolischen Summation Maple zu Hilfe genommen. ■

**Lemma 7.5** (Restschranke für die BINE (7.2)). *Die Funktion*

$$f(n) \stackrel{\text{def}}{=} \frac{p}{q} \frac{\left(\frac{u}{v}\right)^{n-1}}{1-\frac{u}{v}}$$

stellt eine Restschranke für die BINE (7.2) dar.

*Beweis.* Durch Anwendung von Lemma 2.60. ■

*Bemerkung 7.6* (Die wichtige Rolle des Abweichungsradius für BINEs). Aus Satz 7.4 und Lemma 7.5 lässt sich die maßgebliche Bedeutung des Abweichungsradius  $\frac{u}{v}$  für die Spigot-Eignung von BINEs deutlich erkennen. Je mehr sich  $\frac{u}{v}$  den Wert 0 annähert, desto besser ist die entsprechende BINE für den Spigot-Algorithmus geeignet. Wenn  $\frac{u}{v}$  von 1 nur gering abweicht, ist der Einsatz des Spigot-Algorithmus unter Benutzung der entsprechenden BINE unpraktikabel. Ein praktikabler Einsatz ergibt sich, wenn die Bedingung  $\frac{u}{v} \leq \frac{1}{2}$  erfüllt ist. □

## 7.2 Überführung von $\sqrt[k]{\frac{r}{s}}$ in $\frac{p}{q} \left(1 - \frac{u}{v}\right)^{\frac{-1}{k}}$

Führen wir zunächst Lemma 7.7 als einen Bezugspunkt für die Überführung aus.



**Lemma 7.7** (Bestimmung des Abweichungsradius aus dem Hilfsfaktor). *Die Gleichung (7.1) ist äquivalent zu*

$$\frac{u}{v} = \frac{r - s \frac{p^k}{q^k}}{r} = \frac{r q^k - s p^k}{r q^k} . \tag{7.4}$$

*Beweis.* Unmittelbar ersichtlich. ■

Um eine Grundgleichung für die Berechnung der Wurzel  $\sqrt[k]{\frac{r}{s}}$  zu konstruieren, gibt es zwei mögliche gedankliche Wege. Die Bestimmung von  $\frac{p}{q}$  aus  $\frac{u}{v}$  ist komplizierter als die Berechnung der Wurzel selbst. Dagegen ist Bestimmung von  $\frac{u}{v}$  aus  $\frac{p}{q}$  ganz einfach wegen Lemma 7.7. Infolgedessen wird bei der Konstruktion der Grundgleichung zuerst der Hilfsfaktor bestimmt. In Abschnitt 7.2.1 wird die Berechnung des Hilfsfaktors auf die Berechnung von  $\lfloor \sqrt[k]{n} \rfloor \mid n \in \mathbb{N}$  zurückgeführt. Die Berechnung von  $\lfloor \sqrt[k]{n} \rfloor \mid n \in \mathbb{N}$  ist der Gegenstand von Abschnitt 7.2.2.

**7.2.1 Zurückführung der Berechnung eines Hilfsfaktors auf die Berechnung von  $\lfloor \sqrt[k]{n} \rfloor$**

Wir führen zuerst die Lemmata für die Berechnung eines Hilfsfaktors und die Entscheidung über die Zugehörigkeit von  $\sqrt[k]{\frac{r}{s}}$  zu  $\mathbb{Q}$  aus.

**Lemma 7.8** (Identität für Potenz von rationalen Zahlen). *Gilt für reduzierte  $\frac{r}{s}$  und  $\frac{p}{q} \sqrt[k]{\frac{r}{s}} = \frac{p}{q}$ , dann muss  $r = p^k$  und  $s = q^k$  gelten.*

*Beweis.* Betrachten wir das folgende Schließen

$$\left. \begin{array}{l} \sqrt[k]{\frac{r}{s}} = \frac{p}{q} \Rightarrow \frac{r}{s} = \frac{p^k}{q^k} \\ \frac{p}{q} \text{ ist reduziert} \Rightarrow \left. \begin{array}{l} \frac{p^k}{q^k} \text{ ist reduziert} \\ \frac{r}{s} \text{ ist reduziert} \end{array} \right\} \Rightarrow r = p^k, s = q^k . \end{array} \right\}$$

■

**Lemma 7.9** (Zwangspotenz für Numerator). *Gilt  $\sqrt[k]{\frac{n}{q_1^k}} = \frac{p_2}{q_2}$  für  $n, q_1, q_2, p_2 \in \mathbb{N}$ , so existiert  $p_1 \in \mathbb{N}$  mit  $n = p_1^k$ .*

*Beweis.* Betrachten wir das folgende Schließen

$$\sqrt[k]{\frac{n}{q_1^k}} = \frac{p_2}{q_2} \Rightarrow \frac{n}{q_1^k} = \frac{p_2^k}{q_2^k} \Rightarrow \frac{p_2^k q_1^k}{q_2^k} = n . \tag{i}$$

Der Bruch  $\frac{p_2 q_1}{q_2}$  lässt sich in die reduzierte Form  $\frac{p_1}{q}$  überführen, in Bezug auf die das folgende Schließen

$$\frac{p_1^k}{q^k} = \left(\frac{p_1}{q}\right)^k = \left(\frac{p_2 q_1}{q_2}\right)^k = \frac{p_2^k q_1^k}{q_2^k} \stackrel{(i)}{=} n = \frac{n}{1}$$

erfolgt. Zwei reduzierte Brüche  $\frac{p_1^k}{q^k}$  und  $\frac{n}{1}$  sind gleich, infolgedessen müssen  $p_1^k = n$  und  $q^k = 1$  gelten. ■

**Definition 7.10** (Potenzkomplement). Es seien  $n, k \in \mathbb{N}$  gegeben. Ein *Potenzkomplement* des Grades  $k$  für  $n$  ist eine  $h \in \mathbb{N}$ , so dass  $n \times h = q^k$  für irgendeine  $q \in \mathbb{N}$ .  $\square$

**Lemma 7.11** (Einfaches Potenzkomplement). Es sei  $s \in \mathbb{N}$  gegeben. Für eine beliebige  $q_1 \in \mathbb{N}$  wird  $s$  durch das Potenzkomplement  $h = s^{k-1}q_1^k$  zu der Potenz  $sh = q^k$  gemacht, wobei sich  $q = sq_1$  berechnen lässt.

*Beweis.* Unmittelbar ersichtlich.  $\blacksquare$

Der Hauptsatz kann nun formuliert und bewiesen werden.

**Satz 7.12** (Berechnung eines Hilfsfaktors und Entscheidung über die Zugehörigkeit von  $\sqrt[k]{\frac{r}{s}}$  zu  $\mathbb{Q}$ ). Es sei eine Wurzel  $\sqrt[k]{\frac{r}{s}}$  und ein Potenzkomplement  $h$  des Grad  $k$  für  $s$  mit

$$sh = q^k \quad (7.4)$$

gegeben. Die natürliche Zahl  $p$  soll der Gleichung

$$p^k \leq rh < (p+1)^k \quad (7.5)$$

genügen, d. h. es gilt  $p = \left\lfloor \sqrt[k]{rh} \right\rfloor$ .

Dann erfüllt der entsprechende Abweichungsradius  $\frac{u}{v}$  für den Hilfsfaktor  $\frac{p}{q}$  die folgende Ungleichung:

$$\frac{u}{v} < \left(1 + \frac{1}{p}\right)^k - 1. \quad (7.6)$$

Außerdem ist die Wurzel  $\sqrt[k]{\frac{r}{s}}$  genau dann eine rationale Zahl, wenn  $p^k = rh$  gilt. In diesem Fall gelten auch  $\frac{u}{v} = 0$  und  $\sqrt[k]{\frac{r}{s}} = \frac{p}{q}$ .

*Beweis.* Der Beweis ist wie folgt gegliedert:

1. **Herleitung von (7.6).** Betrachten wir dazu das folgende Schließen:

$$\frac{u}{v} \stackrel{(7.4)}{=} \frac{rq^k - sp^k}{rq^k} \stackrel{(7.4)}{=} \frac{rsh - sp^k}{rsh} = \frac{rh - p^k}{rh} \stackrel{(7.5)}{<} \frac{(p+1)^k - p^k}{p^k} = \left(1 + \frac{1}{p}\right)^k - 1.$$

2.  $\sqrt[k]{\frac{r}{s}}$  ist rational, wenn  $p^k = rh$ . Betrachten wir dazu das folgende Schließen

$$\sqrt[k]{\frac{rh}{sh}} \stackrel{p^k=rh}{=} \sqrt[k]{\frac{p^k}{q^k}} = \frac{p}{q}.$$

3.  $p^k = rh$ , wenn  $\sqrt[k]{\frac{r}{s}}$  ist rational. Betrachten wir dazu das folgende Schließen

$$\begin{aligned} \sqrt[k]{\frac{r}{s}} \in \mathbb{Q} &\Rightarrow \frac{p_1}{q_1} = \sqrt[k]{\frac{r}{s}} = \sqrt[k]{\frac{rh}{sh}} = \sqrt[k]{\frac{rh}{q^k}} \Rightarrow \frac{p_1}{q_1} = \sqrt[k]{\frac{rh}{q^k}} \\ \stackrel{\exists p' \in \mathbb{N}}{\Rightarrow} \stackrel{\text{Lemma 7.9}}{r h = p'^k} &\stackrel{(7.5)}{\Rightarrow} p^k \leq p'^k < (p+1)^k \Rightarrow p \leq p' < p+1 \\ &\Rightarrow p = p' = rh. \end{aligned}$$

$\blacksquare$

Als nächstes wird der Prozess der Konstruktion einer BINE für die Berechnung einer Wurzel in einem Schema zusammengefasst.

**Schema 7.13** (Konstruktion einer BINE für  $\sqrt[k]{\frac{r}{s}}$  und Entscheidung über die Zugehörigkeit zu  $\mathbb{Q}$ ). Eine Wurzel  $\sqrt[k]{\frac{r}{s}}$  sei gegeben. Die Berechnung eines Hilfsfaktors  $\frac{p}{q}$  für  $\sqrt[k]{\frac{r}{s}}$  setzt sich aus den nachstehenden Schritten zusammen:

1. Wähle ein Potenzkomplement  $h$  des Grades  $k$  für  $s$ , z.B. mit Lemma 7.11. Merke  $q \in \mathbb{N}$  mit  $sh = q^k$ .
2. Bestimme von  $p$  aus  $r$  und  $h$  anhand von (7.5). Für  $k = 2$  erweist sich die Anwendung von Algorithmus 7.1 als effektiv. Für  $k \geq 3$  kann die Verallgemeinerung in Algorithmus 7.2 angewandt werden. Außerdem bietet [Ber98] bzw. [Ber95] eine interessante Alternative.
3. Wenn  $p^k = h$  gilt, so ist  $\sqrt[k]{\frac{r}{s}} = \frac{p}{q}$  eine rationale Zahl. Bringe  $\frac{p}{q}$  in die reduzierte Form. Die Berechnung ist beendet.
4. Wenn  $p^k > h$  gilt, so ist  $\sqrt[k]{\frac{r}{s}}$  keine rationale Zahl. Für  $\sqrt[k]{\frac{r}{s}}$  wird eine BINE wie folgt konstruiert:
  - (a) Bringe  $\frac{p}{q}$  in die reduzierte Form.
  - (b) Berechne  $\frac{u}{v}$  aus  $r, s, p, q$  anhand von (7.4).
  - (c) Konstruiere nun die BINE für  $\sqrt[k]{\frac{r}{s}}$  anhand von (7.2).

Nach Satz 7.12 erfüllt der Abweichungsradius  $\frac{u}{v}$  die Ungleichung (7.6).

□

*Bemerkung 7.14* (Ein Hilfsfaktor als eine Approximation für die Wurzel und die Rolle des Spigot-Ansatzes). Ein Hilfsfaktor ist selbst eine rationale Approximation für die Wurzel, aus dem mit dem Spigot-Ansatz wiederum Approximationen für die Wurzel berechnet werden. Der Sinn des Einsatzes des Spigot-Ansatzes besteht darin, *Approximationen mit beliebiger Genauigkeit* unter Benutzung *einer Approximation mit angemessener Genauigkeit* zu berechnen. □

### 7.2.2 Die Berechnung von $\lfloor \sqrt[k]{n} \rfloor$ mit $n \in \mathbb{N}$

Die Bestimmung von  $p$  aus  $r$  und  $h$  anhand von (7.5) ist im Wesentlichen die Berechnung von  $\lfloor \sqrt[k]{n} \rfloor$  mit  $n \in \mathbb{N}$ . Die Lösung dieser Berechnungsaufgabe wird zunächst durch den einfachsten Fall motiviert, nämlich  $k = 2$ . Algorithmus 7.1 entsteht durch Modifikation Heron's Iteration und berechnet aus einer gegebenen natürlichen Zahl  $n$  die Zahl  $p$  mit der Eigenschaft  $p^2 \leq n < (p+1)^2$ . Die Korrektheit von Algorithmus 7.1 wird mit Satz 7.15 festgestellt. Durch den Beweis von Satz 7.15 kommt die Arbeitsweise von Algorithmus 7.1 deutlich zum Ausdruck.

**Satz 7.15** (Korrektheit der Berechnung des ganzen Teils der quadratischen Wurzel). *Die Spezifikation in Algorithmus 7.1 ist gültig, d.h. die Berechnung gibt eine natürliche Zahl  $p$  mit  $p^2 \leq n < (p+1)^2$  zurück.*

*Beweis.* Die Repeat-Schleife (von Zeile 4 bis 7) ist der Kern des Algorithmus. Die Werte von  $x$  und  $y$  vor der Ausführung der Schleife werden mit  $x_0$  und  $y_0$ ; unmittelbar nach dem  $k$ -ten Iterationsschritt mit  $x_k$  und  $y_k$  bezeichnet. Die Zeile 5 bzw. 6 des Algorithmus implizieren

$$y_k = \left\lfloor \frac{x_{k-1} + y_{k-1}}{2} \right\rfloor \quad (7.7a)$$

**Algorithmus 7.1** Berechnung von  $\lfloor \sqrt{n} \rfloor$  mit Heron's Iteration

---

```

1: DECLARE AS: HeronIntPartSqrt (n, s, Out root)
2: local  $y := s, x$ 
3:  $x := \left\lfloor \frac{n}{y} \right\rfloor$ 
4: repeat
5:    $y := \left\lfloor \frac{x + y}{2} \right\rfloor$ 
6:    $x := \left\lfloor \frac{n}{y} \right\rfloor$ 
7: until  $(y - x) \leq 1$ 
8: if  $(y \times y \leq n)$  then
9:   root :=  $y$ 
10: else
11:   root :=  $y - 1$ 
12: end if
13: return with root
Ensure:  $\text{root}^2 \leq n < (\text{root} + 1)^2$ 

```

---

bzw.

$$x_k = \left\lfloor \frac{n}{y_k} \right\rfloor . \quad (7.7b)$$

Der Beweis ist wie folgt gegliedert:

1. **Die Existenz von  $p \in \mathbb{N}$  mit**

$$p^2 \leq n < (p + 1)^2 . \quad (7.7c)$$

*Beweis:* Die Menge  $\{k \mid k^2 \leq n\}$  ist nach oben beschränkt, z. B. durch  $n$ ,  $p$  ist das Maximum der Menge.

2. **Invarianz der Repeat-Schleife.** Die zwei wichtigen Eigenschaften der Schleife sind:

(a) **Eigenschaft der Folge  $\{y_k\}$  :**

$$k \geq 1 \quad \Rightarrow \quad y_k \geq p . \quad (7.7d)$$

*Beweis:*

$$\begin{aligned}
 x_{k-1} \stackrel{k \xrightarrow{\text{sub}} k-1}{=} \left\lfloor \frac{n}{y_{k-1}} \right\rfloor &\stackrel{(7.7c)}{\Rightarrow} x_{k-1} y_{k-1} \geq n \geq p^2 && \stackrel{\frac{x+y}{2} \geq \sqrt{xy}}{\Rightarrow} \frac{x_{k-1} + y_{k-1}}{2} \geq p \\
 &&& \Rightarrow \left\lfloor \frac{x_{k-1} + y_{k-1}}{2} \right\rfloor \geq p \quad \Rightarrow \quad y_k \geq p .
 \end{aligned}$$

(b) **Eigenschaft der Folge  $\{y_k - x_k\}$  :**

$$x_{k-1} \leq p \quad \Rightarrow \quad \left. \begin{array}{l} i) \quad x_{k-1} \leq y_{k-1} \\ ii) \quad y_k - x_k \leq \frac{y_{k-1} - x_{k-1}}{2} \end{array} \right\} . \quad (7.7e)$$

*Beweis:*

$$x_{k-1} \leq p \wedge n \geq p^2 \stackrel{k \xrightarrow{\text{sub}} k-1}{\Rightarrow (7.7b)} x_{k-1} \leq y_{k-1} \stackrel{(7.7a)}{\Rightarrow} y_k \leq y_{k-1} \stackrel{(7.7b)}{\Rightarrow} x_k \geq x_{k-1} \stackrel{(7.7a)}{\Rightarrow} y_k - x_k \leq \frac{y_{k-1} - x_{k-1}}{2} .$$

3. **Der Algorithmus terminiert und der Wert von root nach der Termination ist  $p$ .**

*Beweis:* Nur zwei folgende Fälle können vorkommen:

(a) **Die Ungleichung  $x_k \leq p$  ist für alle  $k \geq 1$  gültig.** Aus (7.7e) lässt sich schließen, dass ein  $l \in \mathbb{N}$  mit

$$y_l - x_l \leq 1 \quad (7.7f)$$

existiert. Der Algorithmus terminiert  $x_l$  und  $y_l$ . Es muss außerdem gelten

$$y_l \leq p + 1 , \quad (7.7g)$$

denn das Gegenteil davon

$$y_l \geq p + 2 \quad (7.7h)$$

führt zu folgendem Schließen

$$y_l - x_l \stackrel{k \xrightarrow{\text{sub}} l}{\Rightarrow (7.7b)} y_l - \left\lfloor \frac{n}{y_l} \right\rfloor \stackrel{(7.7h)}{\geq} p + 2 - \left\lfloor \frac{n}{p + 2} \right\rfloor \stackrel{(7.7c)}{\geq} p + 2 - \left\lfloor \frac{(p + 1)^2 - 1}{p + 2} \right\rfloor = 2 > 1 ,$$

was ein Widerspruch zu (7.7f) darstellt. Wir können nun über das Verhalten des Algorithmus wie folgt schließen

$$(7.7g) \wedge (7.7d) \Rightarrow \begin{cases} y_l = p & \Rightarrow \text{Zeile 9 ausgeführt} \Rightarrow \text{root} = p \\ y_l = p + 1 & \Rightarrow \text{Zeile 11 ausgeführt} \Rightarrow \text{root} = p \end{cases} .$$

(b) **Die Ungleichung  $x_k \leq p$  ist nicht für alle  $k \geq 1$  gültig.** d. h. es existiert in diesem Fall eine  $l \in \mathbb{N}$  mit

$$x_l \geq p + 1 . \quad (7.7i)$$

Es muss wiederum (7.7g) gelten, denn aus dem Gegenteil (7.7h) lässt sich wie folgt schließen

$$x_l = \left\lfloor \frac{n}{y_l} \right\rfloor \leq \left\lfloor \frac{n}{p + 2} \right\rfloor \leq \left\lfloor \frac{(p + 1)^2 - 1}{p + 2} \right\rfloor = p ,$$

was ein Widerspruch zu (7.7i) darstellt. Es wird nun weiter wie bei Fall 3a vorgegangen. ■

Nun kommen wir zu einer Lösung für den allgemeinen Fall der Berechnung von  $\lfloor \sqrt[k]{n} \rfloor$  mit  $n \in \mathbb{N}$ . Diese Lösung wird aus dem Newton-Verfahren abgeleitet und ist in Algorithmus 7.2 dargestellt. Die Korrektheit von Algorithmus 7.2 wird mit Satz 7.16 festgestellt.

---

**Algorithmus 7.2** Berechnung von  $\lfloor \sqrt[k]{n} \rfloor$  mit Newton's Verfahren

---

```

1: DECLARE AS: NewtonSqrt_k_of_n (n, k, s, Out root)
2: local  $y := s, x$ 
3:  $x := \left\lfloor \frac{n}{y^{k-1}} \right\rfloor$ 
4: repeat
5:    $y := \left\lfloor \frac{x + (k-1)y}{k} \right\rfloor$ 
6:    $x := \left\lfloor \frac{n}{y^{k-1}} \right\rfloor$ 
7: until  $(y-x) \leq 1$ 
8: if  $(y \leq x)$  then
9:   root := y
10: else
11:   root := y - 1
12: end if
13: return with root
Ensure:  $\text{root}^k \leq n < (\text{root} + 1)^k$ 

```

---

**Satz 7.16** (Korrektheit der Berechnung des ganzen Teils der allgemeinen Wurzel). *Die Spezifikation in Algorithmus 7.2 ist korrekt, d.h. die Berechnung gibt eine natürliche Zahl  $p$  mit  $p^k \leq n < (p+1)^k$  zurück.*

*Beweis.* Analog zum Beweis von Satz 7.15, wobei (7.7e) ii) durch

$$y_k - x_k \leq (y_{k-1} - x_{k-1}) \cdot \frac{k-1}{k}$$

zu ersetzen ist. ■

### 7.3 Schlussbemerkungen und numerische Beispiele

Wir haben bisher die Aufgabe der Herleitung von INEs für Wurzeln sowie der Konstruktion der entsprechenden Restschranken und Spigot-Schranken gelöst. Unter Beachtung von Schema 2.83 ist damit auch die gesamte Aufgabe der Wurzelberechnung mit dem Spigot-Ansatz zu lösen. Die Berechnung einer allgemeinen Wurzel kann dabei analog zu der Berechnung von  $\sqrt{2}$  in Abschnitt 2.5 ohne Problem erfolgen.

In Tabelle 7.1 sind einige numerische Beispiele ausgeführt. Die Herleitung von INEs sowie die Bestimmung von Approximationsgraden und Iterationszahlen erfolgen analog Beispiel 2.79. Die größten benötigten Zahlen werden mit Überwachung im Programm ermittelt.

Die Wurzelberechnung mit dem Spigot-Ansatz wird zum ersten Mal in [Do01] systematisch untersucht. Bezüglich der Konstruktion von Restschranken gibt es leider einen Fehler in der erwähnten Arbeit, der mit [Do01, Lemma 3.2] beginnt. Die Konsequenz des Fehlers wird im Folgenden diskutiert.

*Bemerkung 7.17* (Der Fehler in [Do01, Lemma 3.2] und seine Auswirkung). Lemma 7.3 ist die

$k$	$q_1$	$\frac{p}{q}$	$\frac{u}{v}$	Anschauliche Form der BINE	Approximationsgrad und Iterationszahl	größte benötigte Zahl	numerische Ergebnisse obere Zeile: Spigot untere Zeile: Macsyma
2	1	$\frac{7}{1}$	$\frac{3}{101}$	$0 + \frac{7}{1}(1 + \frac{3}{202}(1 + \frac{9}{404}(1 + \frac{15}{606}(1 + \dots$	(14,23)	24130	7.10633520177594774848 7.1063352017759477485
3	1	$\frac{7}{2}$	$\frac{61}{404}$	$0 + \frac{7}{2}(1 + \frac{61}{1212}(1 + \frac{244}{2424}(1 + \frac{427}{3636}(1 + \dots$	(26,24)	306040	3.69627089585685777047 3.6962708958568577705
4	1	$\frac{5}{2}$	$\frac{183}{808}$	$0 + \frac{5}{2}(1 + \frac{183}{1616}(1 + \frac{915}{6464}(1 + \frac{1647}{9696}(1 + \dots$	(33,25)	1205875	2.66577103326147419304 2.6657710332614741931
5	1	$\frac{2}{1}$	$\frac{37}{101}$	$0 + \frac{2}{1}(1 + \frac{37}{505}(1 + \frac{222}{1010}(1 + \frac{407}{1515}(1 + \dots$	(49,24)	302310	2.19108020662270563103 2.191080206622705631

Tabelle 7.1: Spigot-Berechnung von  $\sqrt[k]{\frac{101}{2}} \mid 2 \leq k \leq 5$  bis zu 20 Nachkommaziffern.

Soll-Form von [Do01, Lemma 3.2]. Die Restschranke lautet dort

$$p + \frac{p}{q} u \left( \frac{1}{1 - \frac{u}{v}} + k \frac{\frac{u}{v}}{1 - (\frac{u}{v})^2} \right). \quad (7.3^{!*})$$

Eine andere Form von (7.3) ist

$$p + \frac{p}{q} u \left( \frac{1}{1 - \frac{u}{v}} + k \frac{\frac{u}{v}}{(1 - \frac{u}{v})^2} \right). \quad (7.3')$$

Weil

$$\frac{1}{3} < \frac{1 - \frac{u}{v}}{1 + \frac{u}{v}} < \frac{\overbrace{(7.3^{!*})}^{\text{EqE}}}{\underbrace{(7.3')}_{\text{EqE}}} < 1$$

für  $0 < \frac{u}{v} < \frac{1}{2}$  gilt, kann in Anlehnung an (2.86) bzw. (2.100) die Abweichung bei der Abschätzung des Konvertierungsfehlers, die durch den unrichtigen Einsatz von (7.3^{!\*}) anstelle von (7.3) in (2.100) verursacht wird, durch die Erhöhung der Iterationszahl um 1 kompensiert werden.

Tabelle 7.1 entspricht im Wesentlichen Tabelle 3 in [Do01], wobei die Bestimmung der unteren Grenze für Iterationszahlen nun auf der richtigen Restschranke in Lemma 7.3 basiert aber zugleich die Aufrundung genauer gemacht wird. In der Tat sind numerische Ergebnisse in den beiden Tabellen identisch und die unteren Grenze für Iterationszahlen weichen von einander nur an einer Stelle ab. Die ungenaue Restschranke in [Do01] wurde also durch Aufrundung bei der Bestimmung der Iterationszahlen kompensiert. Das Gleiche gilt auch für Tabelle 7.2 und Tabelle 1 in [Do01].  $\square$

Die Abhängigkeit des Hilfsfaktors und des Abweichungsradius von dem entsprechenden Potenzkomplement für den Fall der Berechnung von  $\sqrt{2}$  wird in Tabelle 7.2<sup>2</sup> dargestellt. Aus der Tabelle lässt sich schlussfolgern, dass die Erhöhung des Wertes von  $h$  nicht immer eine Verkleinerung von  $\frac{u}{v}$  bewirkt, obwohl wir anhand von (7.6) (7.5) und (7.4) den Wert von  $\frac{u}{v}$  beliebig klein durch Vergrößerung von  $h$  machen können. Die Berechnung von  $\sqrt{2}$  mit der BINE aus der letzten Zeile ist optimal.

Die Näherungsbrüche in der Kettenbruchsdarstellung einer Wurzel, die kleiner als die Wurzel sind, dienen am besten als Hilfsfaktor für die Berechnung der Wurzel nach dem Spigot-Ansatz. So lässt sich der Spigot-Ansatz auch mit dem Konzept des Kettenbruchs kombinieren. Die Entwicklung einer quadratischen (d. h.  $k = 2$ ) Wurzel in (periodische) Kettenbrüche lässt sich algorithmisch ausführen. So ist zumindest bei der Berechnung von quadratischen Wurzeln eine Kombination des Spigot-Ansatzes mit dem Konzept des Kettenbruchs erfolgversprechend. Dem Autor ist unbekannt, wie es bei der Entwicklung von Wurzeln in Kettenbrüche für den Fall  $k \geq 3$  aussieht.

$h$	$\frac{p}{q}$	$\frac{u}{v}$	<b>Anschauliche Form der BINE</b>	<b>Approximationsgrad und Iterationszahl für Genauigkeit von <math>\frac{1}{10^l}</math></b>
$1^2$	$\frac{1}{1}$	$\frac{1}{2}$	$0 + \frac{1}{1}(1 + \frac{1}{4}(1 + \frac{3}{8}(1 + \frac{5}{12}(1 + \dots$	( $3.322 \times (l + 1), l + 2$ )
$2^2$	$\frac{2}{2}$	$\frac{1}{2}$	$0 + \frac{2}{2}(1 + \frac{1}{4}(1 + \frac{3}{8}(1 + \frac{5}{12}(1 + \dots$	( $3.322 \times (l + 1), l + 3$ )
$3^2$	$\frac{4}{3}$	$\frac{1}{9}$	$0 + \frac{4}{3}(1 + \frac{1}{18}(1 + \frac{3}{36}(1 + \frac{5}{54}(1 + \dots$	( $1.048 \times (l + 1), l + 3$ )
$4^2$	$\frac{5}{4}$	$\frac{7}{32}$	$0 + \frac{5}{4}(1 + \frac{7}{64}(1 + \frac{21}{128}(1 + \frac{35}{192}(1 + \dots$	( $1.520 \times (l + 1), l + 3$ )
$5^2$	$\frac{7}{5}$	$\frac{1}{50}$	$0 + \frac{7}{5}(1 + \frac{1}{100}(1 + \frac{3}{200}(1 + \frac{5}{300}(1 + \dots$	( $0.589 \times (l + 1), l + 3$ )

Tabelle 7.2: Verschiedene BINE für die Berechnung von  $\sqrt{2}$  nach Schema 7.13

<sup>2</sup>Die Tabelle entspricht Tabelle 1 in [Do01], wobei die Bestimmung der unteren Grenze für Iterationszahlen nun auf der richtigen Restschranke in Lemma 7.3 basiert, aber zugleich die Aufrundung genauer gemacht wird. Die beide Tabellen sind fast identisch.



# Kapitel 8

## Logarithmenberechnung

Der Gegenstand dieses Kapitels ist die Anwendung des Spigot-Ansatzes auf die Logarithmenberechnung. Der Inhalt wird aus [Do04] übernommen. Die Darstellung erfolgt nun mit den in Kapitel 2 bereitgestellten formalen Grundlagen und wird dadurch kürzer, exakter und systematischer. Die INE für  $\ln 2$  wird hier als ein spezieller Fall von  $\ln(1 + \frac{p}{q})$  betrachtet. Hinsichtlich der Beweisführung gibt es auch Verbesserungen.

Mit der Methode in diesem Kapitel lassen sich alle Logarithmen der Form  $\ln Q \mid Q \in \mathbb{Q}$  berechnen, wobei mit einem Höchstmaß an Zuverlässigkeit beliebige Genauigkeit unter alleiniger Benutzung der Integer-Arithmetik zu bewerkstelligen ist.

Die Logarithmenberechnung in diesem Kapitel erfolgt über eine *Zerlegungsgleichung* der Form

$$\ln Q = k \ln 2 + \ln\left(1 + \frac{p}{q}\right), \quad (8.1)$$

die auf die charakteristische Funktionalgleichung (3.12) für Logarithmen zurückgeht. Nach der Zerlegung wird jeder von beiden Summanden mit dem bisher bekannten Spigot-Ansatz *zur selben Zielbasis* berechnet. Dann wird das Endergebnis als die *normierte Summe* (siehe Definition 8.2) der Teilergebnisse mit Integer-Arithmetik gebildet.

Als Grundlage für die Berechnung werden zuerst in Abschnitt 8.1 die relevanten INEs, Restschranken und Spigot-Schranken untersucht. Darauf aufbauend beschäftigt sich Abschnitt 8.2 mit der Spigot-Berechnung der Teilergebnisse. Abschnitt 8.3 beinhaltet die Zerlegung von  $\ln Q$  und die Kombination der Teilergebnisse. In Abschnitt 8.4 sind Schlussbemerkungen und numerische Beispiele ausgeführt.

Führen wir zunächst den Begriff *normierte Summe* ein.

**Definition 8.1** (Kompatible Faktorsysteme). Es seien zwei Faktorsysteme  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  und  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  zu betrachten.

1.  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  ist in  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  enthalten, wenn  $n \leq n'$  und  $a_k = a'_k, b_k = b'_k \quad \forall k \leq n$  gültig sind.
2.  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  und  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  sind zueinander *kompatibel*, wenn  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  in  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  oder  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  in  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  enthalten.

□

**Definition 8.2** (Normierte Summe zweier FNE mit kompatiblen Faktorsystemen). Es seien zwei FNEs  $c_0 + \text{RNE}\left(\left[\left(\frac{a_i}{b_i}, c_i\right)\right]_{i=1}^n\right)$  und  $c'_0 + \text{RNE}\left(\left[\left(\frac{a'_i}{b'_i}, c'_i\right)\right]_{i=1}^{n'}\right)$  zu betrachten, wobei  $\left[\frac{a_i}{b_i}\right]_{i=1}^n$  in  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$  enthalten ist.

1. Die *direkte Summe* von  $c_0 + \text{RNE}\left(\left[\left(\frac{a_i}{b_i}, c_i\right)\right]_{i=1}^n\right)$  und  $c'_0 + \text{RNE}\left(\left[\left(\frac{a'_i}{b'_i}, c'_i\right)\right]_{i=1}^{n'}\right)$  ist die folgende FNE:

- (a) Das Faktorsystem ist  $\left[\frac{a'_i}{b'_i}\right]_{i=1}^{n'}$ .
- (b) Für  $0 \leq k \leq n$  ist der  $k$ -te Koeffizient  $c_k + c'_k$ .
- (c) Für  $n < k \leq n'$  ist der  $k$ -te Koeffizient  $c'_k$ .

2. Die *normierte Summe* ist die normierte Form von der direkten Summe .

□

## 8.1 INEs, Restschranken und Spigot-Schranken für Logarithmenberechnung

**Lemma 8.3** (ine-Darstellung für  $\ln\left(1 + \frac{p}{q}\right)$  mit  $0 < p \leq q \in \mathbb{Z}$ ). *Es seien  $p$  und  $q$  natürliche Zahlen mit  $0 < p < q$ . Für  $\ln\left(1 + \frac{p}{q}\right)$  gibt es die folgende ine-Darstellung*

$$\begin{aligned} \ln\left(1 + \frac{p}{q}\right) &= 0 + \text{RNE}\left(\left[\left(\frac{2p}{p+2q}, 1\right)\right]_{i=1}^1 \overset{+seq}{+} \left[\left(\frac{(2i-3)p^2}{(2i-1)(p+2q)^2}, 1\right)\right]_{i=2}^{\infty}\right) \\ &= 0 + \frac{2p}{p+2q}\left(1 + \frac{p^2}{3(p+2q)^2}\left(1 + \frac{3p^2}{5(p+2q)^2}\left(1 + \frac{5p^2}{7(p+2q)^2}\left(1 + \dots\right)\right)\right)\right) \end{aligned} \quad (8.2)$$

*In Anlehnung an die Kompaktnotation (2.5) mit dem Anfangsindex Null ist diese INE wie folgt charakterisiert:*

1. Die Koeffizienten sind durch

$$c_0 = 0, \quad c_n = 1 \quad \forall n \geq 1 \quad (8.2a)$$

gegeben.

2. das Faktorsystem ist

$$\left[\frac{2p}{p+2q}\right]_{i=1}^1 \overset{+seq}{+} \left[\frac{(2i-3)p^2}{(2i-1)(p+2q)^2}\right]_{i=2}^{\infty} \quad (8.2b)$$

und lässt sich wie folgt berechnen:

$$\frac{a_1}{b_1} = \frac{2p}{p+2q}, \quad \frac{a_n}{b_n} = \frac{(2n-3)p^2}{(2n-1)(p+2q)^2} \quad \forall n \geq 2. \quad (8.2c)$$

$$\begin{aligned}
 a_1 + \sum_{i=2}^n a_i \prod_{j=1}^{i-1} \frac{a_j}{b_j} &< a_1 + \sum_{i=2}^{\infty} a_i \prod_{j=1}^{i-1} \frac{a_j}{b_j} = a_1 + \sum_{i=2}^{\infty} a_i \frac{a_1}{b_1} \prod_{j=2}^{i-1} \frac{a_j}{b_j} \\
 &= a_1 + \sum_{i=2}^{\infty} \frac{a_1}{b_1} a_i \prod_{j=2}^{i-1} \frac{a_j}{b_j} = 2p + \sum_{i=2}^{\infty} \left( \frac{2p}{p+2q} (2i-3)p^2 \prod_{j=2}^{i-1} \frac{(2j-3)p^2}{(2j-1)(p+2q)^2} \right) \\
 &= 2p + \frac{2p}{p+2q} p^2 \sum_{i=2}^{\infty} (2i-3) \prod_{j=2}^{i-1} \frac{(2j-3)p^2}{(2j-1)(p+2q)^2} \\
 &= 2p + \frac{2p}{p+2q} p^2 \sum_{i=2}^{\infty} \left( \frac{p^2}{(p+2q)^2} \right)^{i-2} = 2p + \frac{2p}{p+2q} p^2 \sum_{i=0}^{\infty} \left( \frac{p^2}{(p+2q)^2} \right)^i \\
 &= 2p + 2p^2 \frac{p}{(p+2q)} \left( \frac{1}{1 - \left( \frac{p}{p+2q} \right)^2} \right) <^1 2p + 2p^2 \frac{1}{(3)} \left( \frac{1}{1 - \left( \frac{1}{3} \right)^2} \right) \\
 &= 2p + \frac{3p^2}{4} < \frac{3(p+2)^2}{4} .
 \end{aligned}$$

---

<sup>1</sup> Durch  $\frac{p}{p+2q} < \frac{1}{3}$ , weil  $p < q$

Herleitung 8.1: Eine Restschränke für (8.2b)

*Beweis.* Setze  $x = \frac{p}{p+2q}$  in (3.23), um

$$\ln \left( 1 + \frac{p}{q} \right) = 2 \left( \sum_{k=0}^{\infty} \frac{p^{2k+1}}{(2k+1)(p+2q)^{2k+1}} \right) \quad (8.3)$$

zu bekommen. Dann wird (2.77b) auf (8.3) angewandt. ■

**Lemma 8.4** (Restschränke für (8.2b)). *Das Faktorsystem (8.2b) hat für  $n \geq 2$  die folgende Restschränke*

$$\frac{3(p+2)^2}{4} ,$$

*die von dem Grad der Approximation  $n$  unabhängig ist.*

*Beweis.* Analog zum Beweis von Lemma 7.3. Wir benutzen Lemma 2.61 als Startpunkt und bauen Herleitung 8.1 auf. ■

In [Do04] wurde der Beweis von Lemma 8.4 ohne  $\sum \prod$ -Notation geführt. Das gleiche wurde erreicht.

**Lemma 8.5** (Restschränke für rechtsbündige Teilkette von (8.2b)). *Für  $m \geq 2$  hat das Faktorsystem  $\left[ \frac{a_i}{b_i} \right]_{i=m}^{\infty}$  mit  $a_i, b_i$  aus (8.2c) eine konstante Restschränke, nämlich*

$$\frac{p^2(p+2q)^2}{4q(p+q)} \cdot (2m-3) .$$

*Beweis.* Analog zum Beweis von Lemma 8.4. ■

**Satz 8.6** (Spigot-Schranke für (8.2b)). *Die zweistellige, jedoch von dem zweiten Argument unabhängige Funktion*

$$f(m, n) \stackrel{\text{def}}{=} \begin{cases} \frac{3(p+2)^2}{4} & \text{falls } m = 1 \\ \frac{p^2(p+2q)^2}{4q(p+q)} \cdot (2m-3) & \text{falls } m \geq 2 \end{cases}$$

stellt eine Spigot-Schranke für (8.4b) dar.

*Beweis.* Man erinnere sich an Definition 2.57-1. und Definition 2.55-1. Der Satz folgt direkt aus Lemma 8.4 und Lemma 8.5. ■

**Korollar 8.7** (ine-Darstellung für  $\ln 2$ ). *Es gibt für  $\ln 2$  die folgende ine-Darstellung*

$$\begin{aligned} \ln 2 &= 0 + \text{RNE} \left( \left[ \left( \frac{2}{3}, 1 \right) \right]_{i=1}^1 \stackrel{+}{\text{seq}} \left[ \left( \frac{2i-3}{9(2i-1)}, 1 \right) \right]_{i=2}^{\infty} \right) \\ &= 0 + \frac{2}{3} \left( 1 + \frac{1}{3 \cdot 9} \left( 1 + \frac{3}{5 \cdot 9} \left( 1 + \frac{5}{7 \cdot 9} \left( 1 + \dots \right) \right) \right) \right) \end{aligned} \quad (8.4)$$

In Anlehnung an die Kompaktnotation (2.5) mit dem Anfangsindex Null ist diese INE wie folgt charakterisiert:

1. Die Koeffizienten sind durch

$$c_0 = 0, \quad c_n = 1 \quad \forall n \geq 1 \quad (8.4a)$$

gegeben.

2. Das Faktorsystem ist

$$\left[ \frac{2}{3} \right]_{i=1}^1 \stackrel{+}{\text{seq}} \left[ \frac{2i-3}{9(2i-1)} \right]_{i=2}^{\infty} \quad (8.4b)$$

und lässt sich wie folgt berechnen:

$$\frac{a_1}{b_1} = \frac{2}{3}, \quad \frac{a_n}{b_n} = \frac{2n-3}{9(2n-1)} \quad \forall n \geq 2. \quad (8.4c)$$

*Beweis.* In Lemma 8.3 setzen wir  $p = q = 1$ . ■

**Korollar 8.8** (Spigot-Schranke für (8.4b)). *Die zweistellige, jedoch von dem zweiten Argument unabhängige Funktion*

$$f(m, n) \stackrel{\text{def}}{=} \begin{cases} \frac{9}{4} & \text{falls } m = 1 \\ \frac{9}{8} \cdot (2m-3) & \text{falls } m \geq 2 \end{cases}$$

stellt eine Spigot-Schranke für (8.4b) dar.

*Beweis.* Das Lemma folgt aus Satz 8.6 mit  $p = q = 1$ . ■

## 8.2 Spigot-Berechnung von $k \times \ln 2$ und $\ln \left(1 + \frac{p}{q}\right)$

**Lemma 8.9** (Spigot-Berechnung von  $k \times \ln 2$ ). *Die Spigot-Berechnung von  $k \times \ln 2$  durch die INE*

$$INE \stackrel{\text{def}}{=} 0 + \text{RNE} \left( \left[ \left( \frac{2}{3}, k \right) \right]_{i=1}^1 \overset{\text{seq}}{+} \left[ \left( \frac{2i-3}{9(2i-1)}, k \right) \right]_{i=2}^\infty \right)$$

erfüllt in Bezug auf Fehlerabschätzung die Ungleichung

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_1, i_1) \leq \frac{3}{10^{i_1}} + \frac{3k}{4 \cdot 9^{n_1}}. \quad (8.5)$$

*Beweis.* Satz 2.62 wird angewandt. Das Faktorsystem von  $INE$  ist (8.4b) und hat nach Korollar 8.8 die konstante Schranke  $3 > \frac{9}{4}$ . Für die Konstruktion einer funktionalen Restschranke für  $INE$  wird Lemma 2.60 mit  $m = 1$ ;  $q = \frac{1}{9}$  (es gilt also  $\frac{2i-3}{9(2i-1)} < \frac{1}{9}$  für  $i \geq 2$ ) angewandt.

Die Anwendung von Satz 2.62 liefert zunächst die Ungleichung

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_1, i_1) \leq \frac{3}{10^{i_1}} + \frac{2k}{3} \left(\frac{1}{9}\right)^n \left(\frac{1}{1 - \frac{1}{9}}\right),$$

die sich dann leicht zu (8.5) überführen lässt. ■

**Lemma 8.10** (Spigot-Berechnung von  $\ln \left(1 + \frac{p}{q}\right)$ ). *Die Spigot-Berechnung von  $\ln \left(1 + \frac{p}{q}\right)$  durch die INE  $INE$  in (8.2) erfüllt in Bezug auf Fehlerabschätzung die folgenden Ungleichungen:*

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_2, i_2) \leq \frac{3(p+2)^2}{4 \cdot 10^{i_2}} + \frac{2p}{p+2q} \left(\frac{p}{p+2q}\right)^{2n_2} \frac{1}{1 - \left(\frac{p}{p+2q}\right)^2}, \quad (8.6)$$

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_2, i_2) \leq \frac{3(p+2)^2}{4 \cdot 10^{i_2}} + \frac{3}{4} \left(\frac{p}{p+2q}\right)^{2n_2} \quad (8.7)$$

und

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_2, i_2) \leq \frac{3(p+2)^2}{4 \cdot 10^{i_2}} + \frac{3}{4 \cdot 9^{n_2}}. \quad (8.8)$$

*Beweis.*

1. Zur Herleitung von (8.6) wird analog zum Beweis von Lemma 8.9 vorgegangen. Das Faktorsystem (8.2b) hat nach Lemma 8.4 die konstante Restschranke  $\frac{3(p+2)^2}{4}$ . Der Termquotient von (8.2) ist ab dem Index 1 durch  $\frac{p^2}{(p+2q)^2}$  nach oben beschränkt. Der 1-te Term ist  $\frac{2p}{p+2q}$ . Durch Anwendung von Satz 2.62 und Lemma 8.4 wird zunächst die Ungleichung

$$\overset{\text{spigot}}{\text{ERR}}_{\text{ine}}(INE, 10, n_2, i_2) \leq \frac{3(p+2)^2}{4 \cdot 10^{i_2}} + \frac{2p}{p+2q} \left(\frac{p^2}{(p+2q)^2}\right)^{n_2} \frac{1}{1 - \left(\frac{p}{p+2q}\right)^2}$$

hergeleitet, die sich dann leicht zu (8.6) überführen lässt.

2. Zur Herleitung von (8.7) wird eine Oberschranke für den Teilterm

$$\frac{2p}{p+2q} \left( \frac{p^2}{(p+2q)^2} \right)^{n_2} \frac{1}{1 - \left( \frac{p}{p+2q} \right)^2} \quad (8.9)$$

von (8.6) konstruiert. Die Ungleichung  $\frac{p}{p+2q} \leq \frac{1}{3}$  wird benutzt, um obere Schranken für den ersten und den dritten Faktor von (8.9), dann (8.9) selbst, zu bekommen. Ersetze das Vorkommen von (8.9) in (8.6) durch seine obere Schranke, um (8.7) zu bekommen.

3. Zur Herleitung von (8.8) wird die Ungleichung  $\frac{p}{p+2q} \leq \frac{1}{3}$  benutzt, um nun alle drei Faktoren von (8.9) zu beschranken. Dann wird das Vorkommen von (8.9) in (8.6) durch seine obere Schranke ersetzt. ■

**Korrolar 8.11** (Obere Schranke der Zahlen in der Spigot-Berechnung mit (8.4b) zur Basis 10).  
Die Schranke ist

$$10 \times \left( \frac{162n - 107}{8} \right) \quad (8.10)$$

für den Grad der Approximation  $n \geq 2$ .

*Beweis.* Durch Anwendung von Lemma 2.42 in Kombination mit Korrolar 8.8. ■

**Korrolar 8.12** (Obere Schranke der Zahlen in der Spigot-Berechnung mit (8.2b) zur Basis 10).  
Die Schranke ist

$$10 \times \left( \frac{(2n-3)p^2}{1 - \frac{p^2}{(p+2q)}} + ((2n-1)(p+2q)^2 - 1) \right) \quad (8.11)$$

oder noch einfacher

$$10 \times \left( \frac{9(2n-3)p^2}{8} + ((2n-1)(p+2q)^2 - 1) \right) \quad (8.12)$$

für den Grad der Approximation  $n \geq 2$ .

*Beweis.* Die Schranke (8.11) wird analog wie (8.10) hergeleitet. Unter Beachtung von  $p < q$  ist

$$\underbrace{(8.11)}_{\text{EqE}} < \underbrace{(8.12)}_{\text{EqE}}$$

ersichtlich. ■

### 8.3 Prozeduren zur Berechnung von $\ln Q$

**Schema 8.13** (Berechnung von  $\ln Q$ ). Die Berechnung von  $\ln Q$  setzt sich aus den folgenden Schritten zusammen.

1. Berechne  $k$  aus  $Q$  durch  $2^k \leq Q < 2^{k+1}$ , beispielsweise mit Algorithmus 8.1.

2. Berechne  $p$  und  $q$ , so dass  $\frac{p+q}{q}$  die reduzierte Form von  $\frac{Q}{2^k}$  ist .
3. Führe die Spigot-Berechnungen für  $k \times \ln 2$  wie in Lemma 8.9 und  $\ln(1 + \frac{p}{q})$  wie in Lemma 8.10 aus.
4. Die Ergebnisse von Schritt 3 sind zwei kompatible FNEs und werden nun laut Definition 8.2 mit Integer-Arithmetik zum Endergebnis kombiniert.

□

Die Funktionstüchtigkeit von Schema 8.13 wird mit Lemma 8.14 begründet.

**Lemma 8.14** (Korrektheit der Zerlegung von  $\ln Q$ ). *In Bezug auf Schema 8.13 gelten die Beziehungen*

$$\ln Q = k \times \ln 2 + \ln\left(1 + \frac{p}{q}\right) \quad (8.13a)$$

und

$$p < q . \quad (8.13b)$$

*Beweis.* Gleichung (8.13a) ist durch

$$\frac{Q}{2^k} = \frac{p+q}{q} \Rightarrow \frac{Q}{2^k} = 1 + \frac{p}{q} \Rightarrow Q = 2^k \times \left(1 + \frac{p}{q}\right) \ln Q = k \times \ln 2 + \ln\left(1 + \frac{p}{q}\right)$$

und die Ungleichung (8.13b) ist durch

$$2^k \leq Q < 2^{k+1} \Rightarrow 1 \leq \frac{Q}{2^k} < 2 \Rightarrow 1 \leq \frac{p+q}{q} < 2 \Rightarrow 1 \leq 1 + \frac{p}{q} < 2 \Rightarrow 0 \leq \frac{p}{q} < 1 \Rightarrow p < q$$

bewiesen. ■

Die Teilprozedur der Berechnung von  $k$  mit  $2^k \leq Q < 2^{k+1}$  in Schema 8.13 kann beispielsweise durch Algorithmus 8.1 erfolgen. Die Arbeitsweise des Algorithmus ist intuitiv klar. Lemma 8.15 wird nur der Vollständigkeit halber ausgeführt.

---

**Algorithmus 8.1** Berechnung von  $\lfloor \log_2 Q \rfloor$

---

**Require:**  $1 < Q \in \mathbb{Q}$

1: **DECLARE AS:** *IntLog2ofQ* ( $Q$ , **Out**  $\text{exp}$ )

2: **local**  $bound := \lceil Q \rceil$

3:  $\text{exp} := 0$

4: **local**  $power := 2$

5: **while** ( $power \leq bound$ ) **do**

6:    $\text{exp} := \text{exp} + 1$

7:    $power := 2 \times power$

8: **end while**

9: **return with**  $\text{exp}$

**Ensure:**  $2^{\text{exp}} \leq Q < 2^{\text{exp}+1}$

---

**Lemma 8.15.** *Die Spezifikation in Algorithmus 8.1 ist korrekt.*

Wenden wir uns nun der Bewerkstelligung einer gewünschten Obergrenze des absoluten Fehlers zu.

**Satz 8.16** (Sicherstellung der Schranke  $\frac{1}{10^l}$ ). Die Berechnung von  $\ln Q = k \times \ln 2 + \ln(1 + \frac{p}{q})$  in Schema 8.13 bewirkt unter den folgenden Bedingungen:

1.  $i_1 \geq l + 2$ ,
2.  $n_1 \geq \log_9(3k) + l \times \log_9 10$ ,
3.  $i_2 \geq l + \log_{10}(3(p+2)^2)$ ,
4.  $n_2 \geq 1 + l \times \log_9 10$ .

einen absoluten Fehler kleiner als  $\frac{1}{10^l}$ .

*Beweis.* Mit Lemma 8.14 ist die Voraussetzung für Lemma 8.10 erfüllt. Durch Lemma 8.10, (8.8), Lemma 8.9 und (8.5) haben wir die obere Schranke

$$\frac{3}{10^{i_1}} + \frac{3k}{4 \cdot 9^{n_1}} + \frac{3(p+2)^2}{4 \cdot 10^{i_2}} + \frac{3}{4 \cdot 9^{n_2}}.$$

für den absoluten Fehler. Unter Beachtung der im Lemma ausgeführten Bedingungen ist ersichtlich, dass alle vier Summanden kleiner als  $\frac{1}{4 \times 10^l}$  sind. ■

**Satz 8.17** (Eine Modifikation der hinreichenden Bedingungen für die Schranke  $\frac{1}{10^l}$ ). Mit dem Kontext in Satz 8.16 kann die untere Schranke für  $n_2$  durch  $1 + l \times \log\left(\frac{p+2q}{p}\right)^2 10$  ersetzt

werden. Der absolute Fehler ist noch kleiner als  $\frac{1}{10^l}$ .

*Beweis.* Beim Beweis von Satz 8.16 benutze man (8.7) anstelle von (8.8). ■

**Schema 8.18** (Berechnung von  $\ln Q$  mit einem absoluten Fehler kleiner als  $\frac{1}{10^l}$ ). Es seien eine rationale Zahl  $Q \in \mathbb{Q}$ ,  $Q > 1$  und eine natürliche Zahl  $l$  gegeben. Die Berechnung von  $\ln Q$  zur Basis 10 setzt sich aus den unten folgenden Schritten zusammen.

1. Berechne  $k$ ,  $p$  und  $q$  nach Schema 8.13.
2. Berechne  $i_1$ ,  $n_1$ ,  $i_2$  und  $n_2$  nach den Bedingungen von Satz 8.17 bzw. Satz 8.16, beispielsweise durch entsprechende Modifikationen von Algorithmus 8.1.
3. Führe die Prozedur in Schema 8.13 mit den Parametern  $i_1$ ,  $n_1$ ,  $i_2$  und  $n_2$  aus.

Nach Satz 8.17 bzw. Satz 8.16 wird eine dezimale Approximation von  $\ln Q$  mit einem absolutem Fehler kleiner als  $\frac{1}{10^l}$  produziert. □



Q	k	$\frac{p}{q}$	$i_1$	$n_1$	$i_2$	$n_2$	ln Q - Obere Zeile: Berechnung mit der Prozedur in Schema 8.18 - Untere Zeile: Berechnung mit Student Edition von Maple V Release 4
$\frac{3}{2}$	0	$\frac{1}{2}$	—	—	41	29	0.4054651081081643819780131154643491365718 0.4054651081081643819780131154643491365720
2	1	—	42	43	—	—	0.69314718055994530941723212145817656807548 0.69314718055994530941723212145817656807550
$\frac{7}{3}$	1	$\frac{1}{6}$	42	43	41	18	0.84729786038720361371010750652065402498948 0.84729786038720361371010750652065402498959
3	1	$\frac{1}{2}$	42	43	41	29	1.09861228866810969139524523692252570464728 1.09861228866810969139524523692252570464749
4	2	—	42	43	—	—	1.38629436111989061883446424291635313615098 1.38629436111989061883446424291635313615100
5	2	$\frac{1}{4}$	42	43	41	21	1.60943791243410037460075933322618763952558 1.60943791243410037460075933322618763952560
6	2	$\frac{1}{2}$	42	43	41	29	1.79175946922805500081247735838070227272278 1.79175946922805500081247735838070227272299
7	2	$\frac{3}{4}$	42	43	42	36	1.94591014905531330510535274344317972963696 1.94591014905531330510535274344317972963708
9	3	$\frac{1}{8}$	42	43	41	17	2.19722457733621938279049047384505140929490 2.19722457733621938279049047384505140929498

Tabelle 8.1: Numerische Beispiele der Berechnung von  $\ln Q = k \times \ln 2 + \ln(1 + \frac{p}{q})$  mit absolutem Fehler kleiner als  $\frac{1}{10^l}$  für  $l = 40$

## 8.4 Schlussbemerkungen und numerische Beispiele

In Abschnitt 8.3 wird nur die Sicherstellung einer Obergrenze des absoluten Fehlers bei der Berechnung zur Basis 10 behandelt. Die Vorgehensweise lässt sich jedoch einfach auf die Berechnung zu einer anderen Basis übertragen.

Die numerischen Beispiele in Tabelle 8.1 sind aus [Do04] entnommen. Der Wert  $l = 40$  wird wegen einer optimalen Ansicht gewählt. Der Spigot-Algorithmus wurde vom Autor in C implementiert und mit dem freien Compiler von Borland Version 5.5 übersetzt.

Zur Berechnung von  $k \ln 2$  stellt die Reihe (4.48) für  $\ln 2$  eine interessante Alternative zu (3.24) dar. Sie ist langsamer konvergent, aber maschinell einfacher zu berechnen durch die Potenz von 2.

Anstelle von (8.1) können auch andere Zerlegungen von  $\ln Q$  eingesetzt werden. Ein Beispiel ist  $\ln Q = k \times \ln(1 + \frac{p_1}{q_1}) + \ln(1 + \frac{p_2}{q_2})$ . Für  $\ln 2$  ist die Zerlegung  $\ln 2 = 7 \ln \frac{16}{15} + 5 \ln \frac{25}{24} + 3 \ln \frac{81}{80}$  interessant.

# Kapitel 9

## Ausblick

Mit der vorliegenden Dissertation wird der Spigot-Algorithmus vollständig formalisiert und in das Konzept des Spigot-Ansatzes eingebettet. Dabei versteht sich der Spigot-Algorithmus als eine Prozedur für die numerische Berechnung von Reihen mit beliebiger Genauigkeit unter alleiniger Benutzung der Integer-Arithmetik, wodurch ein Höchstmaß an Zuverlässigkeit erreicht wird; während der Spigot-Ansatz die gesamte Vorgehensweise bzw. die Werkzeugkiste für den effektiven Einsatz des Spigot-Algorithmus beinhaltet. Die Hoffnung des Autors ist, durch die in der vorliegenden Dissertation geschaffenen formalen Grundlagen einen Beitrag zur Entwicklung zuverlässiger numerischer Verfahren zu leisten.

Es ist besonders interessant, dass viele Reihen aus der Praxis *hypergeometrische* Reihe sind, d. h. ihre Terme sind hypergeometrische Terme in Bezug auf den Index. Solche Reihen lassen sich leicht gemäß (2.77a) bzw. (2.77b) in INEs überführen. So hat der Spigot-Algorithmus ein hohes Anwendungspotential.

Als Verknüpfungspunkte zum Spigot-Algorithmus werden die Methoden für Reihentransformationen von Euler, Kummer, Markoff, Gosper, Zeilberger und Wilf zusammengestellt und gründlich systematisiert. Diese Kenntnisse können eventuell als Grundlagen für die Computerisierung der Reihentransformation benutzt werden.

Nach dem Spigot-Ansatz sind mit der vorliegenden Dissertation neue Methoden für die Berechnung von Wurzeln und Logarithmen entwickelt worden.

Die Methode nach dem Spigot-Ansatz kann als eine vertrauenswürdige Vergleichsbasis dazu dienen, andere Methoden zur numerischen Berechnung von natürlichen Logarithmen auf Zuverlässigkeit zu überprüfen. Im Gegensatz zur Wurzelberechnung lässt sich bei der Logarithmenberechnung die Überprüfung des Ergebnisses durch inverse Berechnung nicht einfacher als die Berechnung selbst durchführen, weil die Exponentialfunktion zur Basis  $e$  bei einem rationalem Argument numerisch nicht genau berechnet werden kann.

Die Ausführungen in Abschnitt 3.4 und Kapitel 7 haben gezeigt, dass die Wurzelberechnung im Wesentlichen auf (3.34) zurückzuführen ist. Ähnliches gilt auch für die Logarithmenberechnung und (3.15). Außerdem wird eine gewünschte Genauigkeit bei der Berechnung mit dem Spigot-Algorithmus im Wesentlichen unter Benutzung von (2.50) bewerkstelligt. Diese Erkenntnisse, dass sich das Wesen eines Sachverhalts in den erwähnten Fällen *durch eine Gleichung* darstellen lässt, erscheinen sehr interessant.

Beispiele für den Einsatz von charakteristischen Funktionalgleichungen in der Reihenentwicklung von Funktionen sowie für andere Umformungen in bessere Berechnungsformen werden zusammengestellt bzw. entwickelt und systematisiert. Die Hoffnung des Autors ist, dadurch Ideen für

die Lösung von neuen Berechnungsproblemen zu fördern.

Durch die Ansätze und Methoden, die in der vorliegenden Dissertation entwickelt bzw. systematisiert werden, wird die Herleitung von Spigot-Berechnungen für spezielle Werte unterstützt. Ein Verfahren für die Berechnung von  $\zeta(2)$  nach dem Spigot-Ansatz mit Einbeziehung der Zeilbergerschen Dreieckstransformation kann z. B. anhand von Beispiel 5.18 hergeleitet werden.

Im Zusammenhang mit dem Spigot-Ansatz sind die nachfolgend ausgeführten Forschungsprobleme interessant.

1. *Transformation spezieller Reihen.* Es sind dem Autor u. a. noch keine Transformationen für  $\zeta(4)$ ,  $\zeta(5)$ , ... in ähnlicher Form wie in Beispiel 5.10, Beispiel 5.11 und Beispiel 6.20 aus der Literatur bekannt.
2. *Algorithmische Reihentransformation.* In Anlehnung an den Algorithmen von Gosper, Zeilberger und Petkovsek (siehe Abschnitt 5.4 bzw. [PWZ96]) ist z. B. zu untersuchen, ob und in welchem Umfang sich die Gopersche Transformation (siehe Kapitel 6) automatisieren lässt. Diesbezüglich steht die Untersuchung einer Möglichkeit der Verallgemeinerung der Wilfschen Methode (vergl. Abschnitt 5.5) am nächsten.
3. *Reihenentwicklung für spezielle Konstante.* Es sind dem Autor u. a. noch keine Reihen für die Feigenbaumkonstante,  $\sqrt{\pi}$  ... aus der Literatur bekannt.
4. *Entscheidbarkeit der Entwicklung in hypergeometrische Reihen.* Als Beispiel ist eine Problemstellung gegeben durch die Frage: Kann man eine Zahl *konstruieren*, für die keine hypergeometrische Reihe existiert.
5. *Tiefgehender Vergleich des Spigot-Ansatzes mit der BBP-Methode.* Die Methode in [BBP97] (Stichwort in Englisch: *digit extracting*) ist eingehender zu untersuchen und mit dem Spigot-Ansatz zu vergleichen.
6. *Umwandlungsmöglichkeiten zwischen Iteration und Reihenentwicklung.* In [BBB89] ist erwähnt, dass ein Iterationsverfahren aus einer Reihenentwicklung für  $\pi$  hergeleitet wird. Die Problemstellung in der Gegenrichtung ist, ob man aus einem Iterationsverfahren eine Reihe gewinnen kann und wenn ja, auf welche Weise. Der Ansatz der charakteristischen Gleichung in Kapitel 3 könnte ein Hinweis für die Lösung sein.
7. *Einsatzmöglichkeit des Spigot-Ansatzes zur Verbesserung numerischer Lösungen für Probleme in der Technik.* Genannt sei hier ein Beispiel für ein zu lösendes Problem. Durch die Mitwirkung in der Forschungsarbeit in [ZSMD03] kam der Autor zur Erkenntnis, dass für die Lösung von bestimmten Differenzialgleichungssystemen in der Kinetik die in der Hardware festverdrahte numerische Berechnung von  $e^x$  und  $\ln(x)$  benutzt wird, über deren Genauigkeit noch keine Aussagen gemacht werden. Eine Frage ist z. B., ob der Einsatz des Spigot-Algorithmus zur Berechnung von  $e^x$  und  $\ln(x)$  *direkt im Zahlenformat des Computers* eine Verbesserung der Lösung des Gesamtproblems bewirkt.

□

# Anhang A

## Grundlegendes über infinitesimale Rechnungen und Potenzreihen

In diesem Anhang werden die wichtigsten Kenntnisse für den Umgang mit Reihen reeller Zahlen systematisiert, um formale Anwendungsschnittstellen für Reihenentwicklung und Reihentransformation bereitzustellen. Die Sachverhalte sind aus der Mathematik gut bekannt und werden zwecks der Anwendung deshalb kompakt ohne Herleitung und meist mit Quellenverweisen ausgeführt. Die Ausdrucksweise ist auch mehr oder weniger vom Stil des Autors geprägt. Der Anhang ist in erster Linie als eine Basis zum Nachschlagen gedacht. Eine leicht verständliche Darstellung in sinnvollen Zusammenhängen wurde angestrebt.

### A.1 Die Vollständigkeit der Menge $\mathbb{R}$

Die Vollständigkeit der Menge  $\mathbb{R}$  ist die theoretische Voraussetzung für die Existenz von Grenzwerten, auf denen die ganze Theorie über Reihen basiert. Die Eigenschaften von  $\mathbb{R}$  werden häufig anhand ihrer Untermengen formuliert, die in Definition A.1 charakterisiert sind.

**Definition A.1** (Spezielle Untermengen von  $\mathbb{R}$ ). [Vergl. Wal92, S. 9]. Es seien  $A$  die Bezeichnung für eine nicht leere Menge von reellen Zahlen und  $\xi$  die Bezeichnung für eine reelle Zahl.

- Wenn  $a \leq \xi$  für alle  $a \in A$  gilt, dann heißt  $\xi$  eine *obere Schranke* von  $A$ . Eine Menge mit einer oberen Schranke heißt *nach oben beschränkt*.
- Wenn  $\xi \leq a$  für alle  $a \in A$  gilt, dann heißt  $\xi$  eine *untere Schranke* von  $A$ . Eine Menge mit einer unteren Schranke heißt *nach unten beschränkt*.
- Eine *beschränkte* Menge ist sowohl nach oben als auch nach unten beschränkt.
- Für  $a < b$  und  $a, b \in \mathbb{R}$  heißt

$$(a, b) \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid a < x < b\}$$

das *offene* und

$$[a, b] \stackrel{\text{def}}{=} \{x \in \mathbb{R} \mid a \leq x \leq b\}$$

das *geschlossene Intervall* jeweils mit der Untergrenze  $a$  und Obergrenze  $b$ .

- Es sei  $\epsilon$  eine beliebige positive reelle Zahl, so heißt das offene Intervall

$$(\xi - \epsilon, \xi + \epsilon) = \{x \in \mathbb{R} \mid \xi - \epsilon < x < \xi + \epsilon\}$$

die  $\epsilon$ -Umgebung von  $\xi$  (vergl. [Wal92, S. 12]).

□

**Definition A.2** (Supremum und Infimum). Besitzt eine nicht leere Menge  $A \subset \mathbb{R}$  eine obere Schranke  $\eta$  mit der Eigenschaft, dass keine andere obere Schranke  $\xi$  von  $A$  kleiner als  $\eta$  ist, so heißt  $\eta$  das<sup>1</sup> *Supremum* von  $A$ . Dementsprechend ist das Supremum  $\eta$  von  $A$  durch zwei folgende Eigenschaften

$$\forall a \in A. a \leq \eta \tag{A.1}$$

und

$$\forall \xi \in \mathbb{R}. ((\forall a \in A. a \leq \xi) \Rightarrow (\eta \leq \xi)) \tag{A.2}$$

eindeutig bestimmt. Der Sachverhalt, dass  $\eta$  das Supremum von  $A$  ist, wird mit

$$\eta = \sup A \tag{A.3}$$

bezeichnet.

Das *Infimum* wird analog als diejenige untere Schranke von  $A$  definiert, so dass keine andere untere Schranke von  $A$  größer als sie ist. Der Sachverhalt, dass  $\eta$  das Infimum von  $A$  ist, wird mit

$$\eta = \inf A \tag{A.4}$$

bezeichnet. □

**Proposition A.3** (Vollständigkeitsaxiom). [Vergl. Wal92, S. 10]. Jede nicht leere, nach oben beschränkte Menge von reellen Zahlen besitzt ein Supremum.

**Proposition A.4** (Prinzip des Infimums). Jede nicht leere, nach unten beschränkte Menge von reellen Zahlen besitzt ein Infimum.

## A.2 Konvergenzverhalten von Folgen und Reihen reeller Zahlen

**Definition A.5** (Reelle Zahlenfolgen und ihre einfachen Eigenschaften). Eine *Folge*

$$[a_i]_{i=i_0}^{\infty} \mid a_i \in \mathbb{R}$$

von reellen Zahlen lässt sich als eine Funktion

$$a : \mathbb{Z}_{\geq i_0} \longrightarrow \mathbb{R}$$

auffassen. Für die Bezeichnung des Funktionswertes wird anstelle von  $a(n)$  vorwiegend  $a_n$  benutzt. Die Zahl  $i_0$  heißt der *Anfangsindex* und die  $a_i$  werden die *Glieder* bzw. die *Terme* der Folge genannt (vergl. [Wal92, S. 58]). □

**Definition A.6** (Einfache Eigenschaften reeller Zahlenfolgen). Es sei  $[a_i]_{i=i_0}^{\infty} \mid a_i \in \mathbb{R}$  eine Folge.

<sup>1</sup>Aus (A.1) und (A.2) ist ableitbar, dass eine Menge höchstens ein Supremum hat.

- Die Folge heißt *beschränkt* bzw. *nach oben beschränkt* oder *nach unten beschränkt*, wenn eine Konstante  $K$  existiert, so dass  $|a_i| \leq K$  bzw.  $a_i \leq K$  oder  $a_i \geq K$  für alle  $i \in \mathbb{Z}_{\geq i_0}$  gilt (vergl. [Wal92, S. 39,58]).
- Die Folge heißt *monoton wachsend* bzw. *streng monoton wachsend*, wenn aus  $i_1 < i_2$  die Ungleichung  $a_{i_1} \leq a_{i_2}$  bzw.  $a_{i_1} < a_{i_2}$  folgt.
- Die Folge heißt *monoton fallend* bzw. *streng monoton fallend*, wenn aus  $i_1 < i_2$  die Ungleichung  $a_{i_1} \geq a_{i_2}$  bzw.  $a_{i_1} > a_{i_2}$  folgt.
- Eine Folge  $[b_j]_{j=j_0}^{\infty}$  heißt eine *Indexverschiebung* von  $[a_i]_{i=i_0}^{\infty}$ , wenn

$$\forall i \in \mathbb{N}. b_{j_0+i} = a_{i_0+i}$$

gilt. Die Zahl  $j_0 - i_0$  heißt der *Offset* der Verschiebung. □

**Definition A.7** (Grenzwert und Häufungswert). Es sei  $[a_i]_{i=i_0}^{\infty} \mid a_i \in \mathbb{R}$  eine Folge.

- Die Folge heißt eine *Nullfolge*, bezeichnet mit  $\lim_{n \rightarrow \infty} a_n = 0$ , wenn die Eigenschaft

$$\forall \epsilon > 0, \exists N(\epsilon) \in \mathbb{N}. ((i > N(\epsilon)) \Rightarrow |a_i| < \epsilon) \quad (\text{A.5})$$

gilt (vergl. [Wal92, S. 58]).

- Die Folge heißt *konvergent gegen den Grenzwert* bzw. den *Limes*  $\xi$ , wenn die Folge  $[a_i - \xi]_{i=i_0}^{\infty}$  eine Nullfolge ist. Man sagt auch, dass  $[a_i]_{i=i_0}^{\infty}$  gegen  $\xi$  *konvergiert*. (Vergl. [Wal92, S. 60]).
- Die reelle Zahl  $\xi$  wird ein *Häufungswert* von  $[a_i]_{i=i_0}^{\infty}$  genannt, wenn jede  $\epsilon$ -Umgebung von  $\xi$  unendlich viele Glieder von  $[a_i]_{i=i_0}^{\infty}$  enthält (vergl. [Wal92, S. 71]). Im Ausdruck der Prädikatenlogik lautet die Voraussetzung

$$\forall \epsilon > 0, \forall n \in \mathbb{N}, \exists m(\epsilon, n) \in \mathbb{N}. ((m(\epsilon, n) > n) \wedge (a_{m(\epsilon, n)} \in (\xi - \epsilon, \xi + \epsilon))) \quad (\text{A.6})$$

□

*Bemerkung A.8* ( $\epsilon$ -Umgebung und der Zusammenhang zwischen Grenzwert und Häufungswert). Die notwendige und hinreichende Bedingung dafür, dass  $[a_i]_{i=i_0}^{\infty}$  gegen  $\xi$  konvergiert, lässt sich ohne „Überbrückung durch eine Nullfolge“ durch die prädikatenlogische Aussage

$$\forall \epsilon > 0, \exists N(\epsilon) \in \mathbb{N}. ((i > N(\epsilon)) \Rightarrow a_i \in (\xi - \epsilon, \xi + \epsilon)) \quad (\text{A.7})$$

formulieren. In Worten der Sprache der  $\epsilon$ -Umgebung heißt es: Für jedes  $\epsilon > 0$  gehören fast alle Glieder der Folge zu der  $\epsilon$ -Umgebung von  $\xi$ . In der gleichen Sprache der  $\epsilon$ -Umgebung ist der semantische Zusammenhang zwischen Grenzwert und Häufungswert ersichtlich, der durch Satz A.9 zum Ausdruck kommt. Satz A.9 ist dabei ein mächtiges formales Werkzeug für die Untersuchung des Konvergenzverhaltens einer Folge. □

**Satz A.9** (Satz von Bolzano-Weierstraß über Grenzwert und Häufungswert). [Vergl. Wal92, S. 72,73]. Es gelten für eine beschränkte Folge  $[a_i]_{i=p}^{\infty} \mid a_i \in \mathbb{R}$  folgende Eigenschaften:

- Die Folge besitzt einen einzigen größten Häufungswert, bezeichnet mit  $\limsup_{n \rightarrow \infty} a_n$ , und einen einzigen kleinsten Häufungswert, bezeichnet mit  $\liminf_{n \rightarrow \infty} a_n$ . (Es gilt also

$$\liminf_{n \rightarrow \infty} a_n \leq \xi \leq \limsup_{n \rightarrow \infty} a_n$$

für jeden Häufungswert  $\xi$  der Folge.)

- Für jede beliebige positive reelle Zahl  $\epsilon$  gibt es nur höchstens endlich viele Glieder der Folge, die nicht zum offenen Intervall

$$\left( \liminf_{n \rightarrow \infty} a_n - \epsilon, \limsup_{n \rightarrow \infty} a_n + \epsilon \right)$$

gehören. Im Ausdruck der Prädikatenlogik heißt es

$$\forall \epsilon > 0, \exists N(\epsilon) \in \mathbb{N}. \left( (i > N(\epsilon)) \Rightarrow a_i \in \left( \liminf_{n \rightarrow \infty} a_n - \epsilon, \limsup_{n \rightarrow \infty} a_n + \epsilon \right) \right). \quad (\text{A.8})$$

- Die Folge ist genau dann konvergent, wenn

$$\liminf_{n \rightarrow \infty} a_n = \limsup_{n \rightarrow \infty} a_n \quad (\text{A.9})$$

gilt. In dem Fall, dass (A.9) gilt, gilt auch

$$\lim_{n \rightarrow \infty} a_n = \liminf_{n \rightarrow \infty} a_n = \limsup_{n \rightarrow \infty} a_n .$$

**Definition A.10** (Reihen mit reellen Gliedern). Es sei eine reelle Zahlenfolge  $[a_i]_{i=i_0}^{\infty}$  gegeben.

Der formale Ausdruck der Gestalt

$$\sum_{i=i_0}^{\infty} a_i \quad (\text{A.10})$$

bzw.

$$a_{i_0} + a_{i_0+1} + a_{i_0+2} + \dots \quad (\text{A.11})$$

wird als die *unendliche Reihe* mit der Gliederfolge  $[a_i]_{i=i_0}^{\infty}$  definiert. Ferner werden vereinbart:

- Die ganze Zahl  $i_0$  heißt der *Anfangsindex*.
- Die reelle Zahl  $a_i$  heißt das *Glied mit dem Index  $i$* .
- Die Summe

$$s_n = \sum_{i=i_0}^n a_i \quad (\text{A.12})$$

heißt die *Teilsumme bis zum Index  $n$* .

- Die Summe

$$r_n = \sum_{i=n+1}^{\infty} a_i \quad (\text{A.13})$$

heißt der *Rest durch Abschneidung ab Index  $n + 1$* . □

**Definition A.11** (Indexverschiebung und Umordnung von Reihen). Es seien  $\sum_{i=i_0}^{\infty} a_i$  und  $\sum_{j=j_0}^{\infty} b_j$  zwei Reihen.

- Die Reihe  $\sum_{j=j_0}^{\infty} b_j$  heißt eine *Indexverschiebung* von  $\sum_{i=i_0}^{\infty} a_i$  wenn

$$\forall i \in \mathbb{N}. b_{j_0+i} = a_{i_0+i}$$

gilt. Die Zahl  $j_0 - i_0$  heißt der *Offset* der Verschiebung.

- Ist  $i_0 = j_0$  und entsteht  $[b_j]_{j=i_0}^{\infty}$  aus  $[a_j]_{i=i_0}^{\infty}$  durch Umordnung der Glieder, so heißt  $\sum_{j=j_0}^{\infty} b_j$  eine *Umordnung* von  $\sum_{i=i_0}^{\infty} a_i$ . In formaler Ausdrucksweise heißt es, dass es eine bijektive Abbildung

$$\varphi : \mathbb{Z}_{\geq i_0} \longrightarrow \mathbb{Z}_{\geq i_0}$$

gibt, so dass

$$\forall i \in \mathbb{Z}_{\geq i_0}. b_{\varphi(i)} = a_i$$

gilt (vergl. [Wal92, S. 98]).

□

**Definition A.12** (Konvergenzverhalten von Reihen). Es sei  $\sum_{i=i_0}^{\infty} a_i$  eine Reihe.

- Die Reihe heißt *konvergent*, falls die Folge  $\left[ \sum_{i=i_0}^n a_i \right]_{n=i_0}^{\infty}$  ihrer Teilsummen konvergent ist.

Anstelle von  $S = \lim_{n \rightarrow \infty} \sum_{i=i_0}^n a_i$  wird vorwiegend die abgekürzte Schreibweise  $S = \sum_{i=i_0}^{\infty} a_i$  benutzt. Die Zahl  $S$  wird auch einfach die *Summe* der Reihe genannt.

- Eine nicht konvergente Reihe heißt *divergent*.
- Ist  $\sum_{i=i_0}^{\infty} a_i$  konvergent,  $\sum_{i=i_0}^{\infty} |a_i|$  aber nicht, so heißt die Reihe *bedingt konvergent*.
- Ist  $\sum_{i=i_0}^{\infty} |a_i|$  konvergent, dann heißt die Reihe *absolut konvergent*. Satz A.13 stellt sicher, dass eine absolut konvergente Reihe stets konvergiert.

□

**Satz A.13** (Eigenschaft der absolut konvergenten Reihen). [Vergl. Wal92, S. 94,98]. Ist eine Reihe  $\sum_{i=i_0}^{\infty} a_i$  absolut konvergent, so konvergieren sie und alle ihre Umordnungen gegen den gleichen Wert.

**Satz A.14** (Riemannscher Umordnungssatz). [Vergl. Wal92, S. 105]. Konvergiert eine Reihe  $\sum_{i=i_0}^{\infty} a_i$  nur bedingt, so existiert für jedes  $S \in \mathbb{R}^2$  eine Umordnung  $\sum_{i=i_0}^{\infty} b_i$  der Reihe mit  $S = \sum_{i=i_0}^{\infty} b_i$ .

<sup>2</sup>Die Eigenschaft gilt auch für  $S \in \mathbb{R} \cup \{\pm\infty\}$ , aber die Symbole  $\pm\infty$  werden in der vorliegenden Dissertation nicht formal eingeführt.



**Satz A.15** (Konvergenzkriterium für Reihen mit positiven Gliedern). [Vergl. Wal92, S. 92]. Eine unendliche Reihe mit nicht negativen Gliedern ist konvergent, wenn die Folge ihrer Teilsumme beschränkt ist.

**Satz A.16** (Majorantenkriterium für Konvergenz). [Vergl. Wal92, S. 92]. Ist  $0 \leq a_n \leq c_n$  für fast alle  $n$  und ist  $\sum c_n$  konvergent, so ist auch  $\sum a_n$  konvergent.

**Definition A.17** (Alternierende Reihen). Eine Reihe  $\sum_{i=i_0}^{\infty} a_i$  heißt *alternierend*, wenn die Glieder  $a_i$  abwechselnd positiv und negativ sind. □

**Satz A.18** (Konvergenzkriterium von Leibniz). [Vergl. Wal92, S. 93]. Ist die Reihe  $\sum_{i=i_0}^{\infty} a_i$  alternierend und konvergiert die Folge der Absolutbeträge der Glieder streng monoton gegen Null, dann ist die Reihe konvergent, und für jeden Reihenrest gilt

$$r_n = \sum_{k=n+1}^{\infty} a_k = \Theta a_{n+1} \quad \text{mit} \quad 0 < \Theta = \Theta_n < 1 .$$

**Satz A.19** (Konvergenzkriterium von Cauchy für Reihen). [Vergl. Wal92, S. 94]. Die Reihe  $\sum_{i=i_0}^{\infty} a_i$  ist dann und nur dann konvergent, wenn zu jedem  $\epsilon > 0$  ein  $N = N(\epsilon)$  existiert, so dass für alle  $n > m \geq N$  gilt

$$|s_n - s_m| = |a_{m+1} + a_{m+2} + \dots + a_n| < \epsilon .$$

**Satz A.20** (Wurzelkriterium für absolute Konvergenz). [Vergl. Wal92, S. 95]. Existiert eine Zahl  $q$  mit  $0 < q < 1$  derart, dass

$$\sqrt[n]{|a_n|} \leq q < 1 \quad \text{für fast alle}^3 n \tag{A.14}$$

gilt, so ist  $\sum a_n$  absolut konvergent. Ist dagegen

$$\sqrt[n]{|a_n|} \geq 1 \quad \text{für unendlich viele } n, \tag{A.15}$$

so ist  $\sum a_n$  divergent. Die prädikatenlogische Formulierung von (A.14) bzw. (A.15) lautet

$$\exists q < 1, \exists N(q) \in \mathbb{N}, \forall n. \left( (n > N(q)) \Rightarrow \left( \sqrt[n]{|a_n|} \leq q \right) \right) \tag{A.16}$$

bzw.

$$\forall n \in \mathbb{N}, \exists N. \left( (N > n) \wedge \left( \sqrt[N]{|a_N|} \geq 1 \right) \right) . \tag{A.17}$$

**Satz A.21** (Quotientenkriterium für absolute Konvergenz). [Vergl. Wal92, S. 95]. Ist  $a_n \neq 0$  und existiert ein  $q \in (0, 1)$  mit

$$\left| \frac{a_{n+1}}{a_n} \right| \leq q < 1 \quad \text{für fast alle } n, \tag{A.18}$$

so ist  $\sum a_n$  absolut konvergent. Ist dagegen

$$\left| \frac{a_{n+1}}{a_n} \right| \geq 1 \quad \text{für fast alle } n, \tag{A.19}$$

---

<sup>3</sup>Es existiert eine  $N_0 \in \mathbb{N}$ , so dass die Eigenschaft für alle  $n > N_0$  gilt.

so ist  $\sum a_n$  divergent. Die prädikatenlogische Formulierung von (A.18) bzw. (A.19) lautet

$$\exists q < 1, \exists N(q) \in \mathbb{N}, \forall n. \left( (n > N(q)) \Rightarrow \left( (a_n \neq 0) \wedge \left( \left| \frac{a_{n+1}}{a_n} \right| \leq q \right) \right) \right) \quad (\text{A.20})$$

bzw.

$$\exists N \in \mathbb{N}, \forall n > N. \left( (a_n \neq 0) \wedge \left( \left| \frac{a_{n+1}}{a_n} \right| \geq 1 \right) \right). \quad (\text{A.21})$$

**Satz A.22** (Produkt von absolut konvergenten Reihen). [Vergl. Wal92, S. 102]. Sind zwei Reihen  $\sum_{i=0}^{\infty} a_i$  und  $\sum_{i=0}^{\infty} b_i$  absolut konvergent, so kann ihr Produkt durch gliedweise Multiplikation gemäß der Gleichung

$$\left( \sum_{i=0}^{\infty} a_i \right) \left( \sum_{i=0}^{\infty} b_i \right) = \sum_{i,j=0}^{\infty} a_i b_j \quad (\text{A.22})$$

berechnet werden. Die Produktreihe ist absolut konvergent und ihre Summe ist von einer konkreten Anordnung der Glieder unabhängig.

Eine konkrete Indizierung/Anordnung für das Produkt in (A.22) ist das Cauchy-Produkt.

**Definition A.23** (Cauchy-Produkt von Reihen). [Vergl. Wal92, S. 103]. Das Cauchy-Produkt zweier Reihen  $\sum_{i=0}^{\infty} a_i$  und  $\sum_{i=0}^{\infty} b_i$  wird gemäß der Gleichung

$$\left( \sum_{i=0}^{\infty} a_i \right) \left( \sum_{i=0}^{\infty} b_i \right) = \sum_{n=0}^{\infty} \left( \sum_{i=0}^n a_i b_{n-i} \right) \quad (\text{A.23})$$

gebildet. □

### A.3 Konvergenz von Funktionenfolgen und Funktionenreihen

**Definition A.24** (Funktionenfolge, punktweise und gleichmäßige Konvergenz).

(Vergl. [Wal92, S. 139, 141] und [JH84, S. 574]). Eine (reelle) *Funktionenfolge*  $[f_i]_{i=i_0}^{\infty}$  versteht sich als eine Folge von reellwertigen Funktionen, die alle auf eine Menge  $D \subset \mathbb{R}$  definiert sind. Im Folgenden werden einige Begriffe in Bezug auf ihr Konvergenzverhalten definiert.

1. Ist  $\xi \in D$  und ist die Zahlenfolge  $[f_i(\xi)]_{i=i_0}^{\infty}$  konvergent, so heißt die Folge  $[f_i]_{i=i_0}^{\infty}$  *konvergent im Punkt  $\xi$* .
2. Konvergiert die Folge  $[f_i(x)]_{i=i_0}^{\infty}$  für jedes  $x \in D$ , so lässt sich auf  $D$

$$f(x) \stackrel{\text{def}}{=} \lim_{i \rightarrow \infty} f_i(x) \quad (\text{A.24a})$$

definieren. Man sagt dann, dass die Folge  $[f_i(x)]_{i=i_0}^{\infty}$  auf  $D$  **punktweise** gegen  $f(x)$  konvergiert. Die Funktion  $f(x)$  heißt die *Grenzfunktion* der Folge  $[f_i]_{i=i_0}^{\infty}$ . Die Eigenschaft (A.24a) lautet im Detail

$$\forall x \in D, \forall \epsilon > 0, \exists N(x, \epsilon) \in \mathbb{N}. \left( (i > N(x, \epsilon)) \Rightarrow (|f_i(x) - f(x)| < \epsilon) \right). \quad (\text{A.24b})$$

3. Gilt die im Vergleich zu (A.24b) stärkere Eigenschaft

$$\forall \epsilon > 0, \exists N(\epsilon) \in \mathbb{N}. ((i > N(\epsilon)) \Rightarrow (|f_i(x) - f(x)| < \epsilon)) \quad , \quad (\text{A.25a})$$

so heißt die Folge  $[f_i(x)]_{i=i_0}^{\infty}$  konvergiert auf  $D$  *gleichmäßig* gegen die Funktion  $f(x)$ . Der Sachverhalt wird mit

$$f(x) = \lim_{i \rightarrow \infty} f_i(x) \quad \text{gleichmäßig auf } D \quad (\text{A.25b})$$

bezeichnet.

□

**Definition A.25** (Funktionenreihe). [Vergl. JH84, S. 575]. Es sei  $[f_i]_{i=i_0}^{\infty}$  eine Funktionenfolge. Der formal gebildete Ausdruck

$$f_{i_0}(x) + f_{i_0+1}(x) + \dots + f_{i_0+n}(x) + \dots \quad (\text{A.26a})$$

bzw.

$$\sum_{i=i_0}^{\infty} f_i(x) \quad (\text{A.26b})$$

stellt die *Funktionenreihe* mit der Gliederfolge  $[f_i]_{i=i_0}^{\infty}$  dar. Die Elemente bzw. Unterstrukturen der Reihe werden analog zu Definition A.10 genannt. Auch der Begriff Indexverschiebung kann analog zu Definition A.11 definiert werden. Das gesamte Konvergenzverhalten der Reihe wird auf das Konvergenzverhalten der Folge  $\left[ \sum_{i=i_0}^n f_i(x) \right]_{n=i_0}^{\infty}$  ihrer Teilsummen zurückgeführt. □

## A.4 Konvergenzverhalten von Potenzreihen

**Definition A.26** (Potenzreihe). [Vergl. NS84, S. 371]. Eine *Potenzreihe* ist eine Funktionenreihe der Form

$$a_0 + a_1x + a_2x^2 + \dots + a_ix^i + \dots = \sum_{i=0}^{\infty} a_ix^i \quad (\text{A.27a})$$

bzw.

$$a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \dots + a_i(x - x_0)^i + \dots = \sum_{i=0}^{\infty} a_i(x - x_0)^i \quad , \quad (\text{A.27b})$$

wobei

- die  $a_i$  fest vorgegebene reelle Zahlen sind und *Koeffizienten* genannt werden,
- das Symbol  $x$  die *Variable* der Reihe — als Funktion gesehen — darstellt,
- $x_0$  eine fest vorgegebene reelle Zahl ist und der *Entwicklungspunkt* (vergl. [NS84, S. 371]) oder der *Mittelpunkt* (vergl. [Kno96, S. 158]) der Reihe genannt wird.

Unter Einbeziehung von Indexverschiebung kann als Anfangindex auch andre Werte als 0 gewählt werden. □

Mit der Substitution  $x_0 \xrightarrow{\text{sub}} 0$  geht (A.27b) in (A.27a) über. Dementsprechend wird (A.27b) eine Entwicklung um den Punkt  $x_0$  und (A.27a) eine Entwicklung um den Nullpunkt genannt. Für die vorliegende Dissertation sind hauptsächlich Potenzreihen um den Nullpunkt von Bedeutung.

**Satz A.27** (Konvergenzverhalten und Konvergenzradius von Potenzreihen). [Vergl. Wal92, S. 143]. Es sei  $\sum_{i=0}^{\infty} a_i x^i$  eine Potenzreihe.

1. Konvergiert die Reihe für jedes  $x$ , so heißt die Reihe *beständig konvergent*.
2. Konvergiert die Reihe für kein  $x \neq 0$ , so heißt die Reihe *nirgends konvergent*.
3. Ist die Reihe weder *beständig* noch *nirgends konvergent*, so besitzt sie einen **Konvergenzradius**  $r \in \mathbb{R} \mid r > 0^4$ , und zwar im folgenden Sinn:
  - (a) die Reihe ist für  $|x| < r$  *absolut konvergent*,
  - (b) die Reihe ist für  $|x| > r$  *divergent*,
  - (c) für  $0 < s < r$  ist die Reihe im Bereich  $|x| \leq s$  *gleichmäßig konvergent*.

Der Konvergenzradius hängt dabei nur von  $|a_i|$  ab und lässt sich mit Hilfe der Formel von Cauchy-Hadamard

$$e = \frac{1}{L} \quad \text{mit} \quad L = \limsup_{i \rightarrow \infty} \sqrt[i]{|a_i|} \quad (\text{A.28})$$

berechnen.<sup>5</sup> Das Intervall  $(-r, r)$  wird als **Konvergenzintervall** der Reihe bezeichnet.

Satz A.28 wird in [Kno96, S. 153] Hauptsatz genannt.

**Satz A.28** (Identitätssatz für Potenzreihen). [Vergl. Wal92, S. 147]. Es seien  $f(x) = \sum_{i=0}^{\infty} a_i x^i$  und  $g(x) = \sum_{i=0}^{\infty} b_i x^i$  zwei Potenzreihen mit jeweils positivem Konvergenzradius. Gilt dabei

- $f(x) = g(x)$  für  $|x| < \alpha$  ( $\alpha > 0$ )

oder auch nur

- $f(x_i) = g(x_i)$  für  $i \in \mathbb{Z}_{\geq 0}$ , wobei  $x_i \neq 0$  gilt und  $[x_i]_{i=0}^{\infty}$  eine Nullfolge ist,

so sind beide Reihen identisch, d. h. es gilt  $a_i = b_i$  für  $i \in \mathbb{Z}_{\geq 0}$ .

**Satz A.29** (Grenzwertsatz von Abel für Potenzreihen). [Vergl. Wal92, S. 149]. Die Potenzreihe  $\sum_{i=0}^{\infty} a_i x^i$  habe den Konvergenzradius  $r > 0$  und sei auch für  $x = r$  konvergent. Es gelten folgende Eigenschaften:

1. Die Reihe ist im abgeschlossenen Intervall  $[0, r]$  *gleichmäßig konvergent*.
2. Die Funktion  $f(x) = \sum_{i=0}^{\infty} a_i x^i$  ist *linksseitig stetig*, d. h.

$$\lim_{x \rightarrow r-0} f(x) = f(r) = \sum_{i=0}^{\infty} a_i (r)^i .$$

3. Es gelten auch die Analogien von 1. und 2., wenn die Reihe für  $x = -r$  konvergiert.

<sup>4</sup>Wir benutzen hier nicht die Klausel  $0 \leq r \leq \infty$  aus dem gleichen Grund wie im Satz A.14, Seite 146.

<sup>5</sup>Im Zusammenspiel mit  $\infty$  würden auch  $\frac{1}{0} = \infty$  und  $\frac{1}{\infty} = 0$  für die Fälle 1. und 2. gelten.

## A.5 Das Rechnen mit Potenzreihen

**Satz A.30** (Einfache Operation von Potenzreihen). *Es seien die beiden Potenzreihen*

$$f(x) = \sum_{n=0}^{\infty} a_n x^n$$

und

$$g(x) = \sum_{n=0}^{\infty} b_n x^n$$

mit den positiven Konvergenzradius  $r_a$  und  $r_b$  gegeben. So gelten

$$f(x) \pm g(x) = \sum_{n=0}^{\infty} (a_n \pm b_n) x^n$$

und

$$f(x)g(x) = \sum_{n=0}^{\infty} \left( \sum_{i=0}^n a_i b_{n-i} \right) x^n \quad (\text{vergl. [Wal92, S. 144]})$$

jeweils für  $x < \min\{r_a, r_b\}$ .

**Satz A.31** (Mittelpunktsverschiebung von Potenzreihen). [Vergl. Kno96, S. 175]. *Es sei*

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n,$$

eine Potenzreihe mit dem Konvergenzradius  $r$ . So lässt sich  $f(x)$  auch um jeden Punkt  $x_1 \in (x_0 - r, x_0 + r)$  als Mittelpunkt in eine Potenzreihe

$$f(x) = \sum_{k=0}^{\infty} b_k (x - x_1)^k$$

entwickeln, wobei

$$b_k = \sum_{n=0}^{\infty} \binom{n+k}{k} a_{n+k} (x_1 - x_0)^n$$

gilt. Der Konvergenzradius der neuen Reihe um den Mittelpunkt  $x_1$  ist mindestens  $r - |x_1 - x_0|$ .

**Satz A.32** (Gliederweise Differenziation von Potenzreihen). [Vergl. Kno96, S. 176]. *Eine durch eine Potenzreihe dargestellte Funktion*

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n$$

ist in jedem inneren Punkt  $x_1$  ihres Konvergenzintervalles differenzierbar. Außerdem kann die Ableitung  $f'(x)$  durch gliederweise Differenziation berechnet werden, d. h. es gilt

$$f'(x_1) = \sum_{n=1}^{\infty} n a_n (x_1 - x_0)^{n-1} = \sum_{n=0}^{\infty} (n+1) a_{n+1} (x_1 - x_0)^n.$$

**Satz A.33** (Gliederweise Mehrfache Differenziation von Potenzreihen). [Vergl. Kno96, S. 177]. Eine durch eine Potenzreihe dargestellte Funktion

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n$$

ist in jedem inneren Punkt  $x_1$  ihres Konvergenzintervalles beliebig oft differenzierbar, und es gilt

$$f^{(k)}(x_1) = \sum_{n=0}^{\infty} (n+1)(n+2)\dots(n+k)a_{n+k}(x_1 - x_0)^n .$$

**Satz A.34** (Gliederweise Integration von Potenzreihen). [Vergl. Kno96, S. 178]. Das Integral der durch  $f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n$  im Konvergenzintervall dargestellten (stetigen) Funktion  $f(x)$  lässt sich durch gliederweise Integration gemäß der Formel

$$\int_{x_1}^{x_2} f(t) dt = \sum_{n=0}^{\infty} \frac{a_n}{n+1} \left( (x_2 - x_0)^{n+1} - (x_1 - x_0)^{n+1} \right)$$

berechnen, sofern  $x_1$  und  $x_2$  zum Konvergenzintervall gehören.

**Satz A.35** (Hauptsatz der Differenzial- und Integralrechnung). (Vergl. [Kno96, S. 169] und [Wal92, S. 260]). Ist  $f(x)$  über  $[a, b]$  integrierbar, und besitzt  $f(x)$  dort ein unbestimmtes Integral  $F(x)$ , so gilt

$$\int_a^b f(x) dx = F(b) - F(a) \stackrel{\text{def}}{=} F(x) \Big|_a^b .$$

**Satz A.36** (Taylorsche Reihe). [Vergl. Wal92, S. 262]. Jede Potenzreihe  $\sum_{n=0}^{\infty} a_n (x - a)^n$  mit positivem Konvergenzradius lässt sich mit Hilfe der durch sie dargestellten Funktion  $f$  in der Form

$$f(x) = \sum_{n=0}^{\infty} \frac{f^n(a)}{n!} (x - a)^n$$

schreiben.

# Anhang B

## Verschiedenes

### B.1 Beispiel für den Beweis von Nebenbedingungen

**Beweis B.1** (Die Gültigkeit von (5.21)). Es gilt

$$\begin{aligned} F(n, k) &= \frac{k! n!}{(k+1)(n+k+1)!} \\ &= \frac{1 \cdot 2 \cdot 3 \cdot 4 \dots \underbrace{n}}{(k+1) \cdot (k+1) \cdot (k+2) \cdot (k+3) \cdot \dots \underbrace{(k+n-1)} \cdot (k+n) \cdot (k+n+1)} \\ &< \frac{1}{(k+n) \cdot (k+n+1)} < \frac{1}{n^2}, \end{aligned}$$

woraus

$$\sum_{k=0}^{n-1} F(n, k) < n \cdot \frac{1}{n^2} = \frac{1}{n}$$

folgt. Der Rest des Beweises ist offensichtlich. □

### B.2 Grundlegendes für die Benutzung des Gosper-Algorithmus unter Maple

**Quelle B.2** (Maple-Paket HYPERG). Das Maple-Paket HYPERG beinhaltet Werkzeuge zur Manipulation hypergeometrischer Terme und ist über

<http://www-igm.univ-mlv.fr/~gauthier/PROGRAMS/HYPERG/V1.0/HTML/download.html>

verfügbar. Ein Benutzerhandbuch ist unter

<http://www-igm.univ-mlv.fr/~gauthier/PROGRAMS/HYPERG/V1.0/HTML/manual.html>

zu holen. Für den Gegenstand der vorliegenden Dissertation ist die Funktion Gosper des Pakets vom Hauptinteresse. □

*Beispiel B.3* (Zur Anwendung des Gosper-Algorithmus). Im Folgenden ist die Maple-Sitzung für die Berechnung der linken Adjungierten  $F(n, k)$  aus dem rechten Adjungierten  $G(n, k)$  des Wz-Paares  $(F(n, k), G(n, k))$  in Beispiel 5.10 mit dem Paket HYPERG ausgeführt.

```
> # Einbindung des Pakets HYPERG)
> read 'HYPERG';
> # De-Quantifizierung des Paket-Namens
> with(HYPERG);
> # Definition von G(n,k)
> G:=(n,k) -> k!*n!/((n+1)*(n+k+1)!);
```

$$G := (n, k) \rightarrow \frac{k! n!}{(n+1)(n+k+1)!}$$

```
> # Berechnung/Definition der Differenz von G nach k
> difK:=(n,k) ->G(n,k+1)-G(n,k);
```

$$\text{difK} := (n, k) \rightarrow G(n, k+1) - G(n, k)$$

```
> # Aufrufen der Funktion Gosper, Berechnung der diskreten Stammfunktion von
> # difK nach n
> Gosper(difK(n,k),n);
```

$$\frac{(n+k+2)n!(-(k+1)!(n+k+1)!+k!(n+k+2)!)}{(k+1)(n+1)(n+k+2)!(n+k+1)!}$$

```
> simplify("");
```

$$\frac{\Gamma(k+1)\Gamma(n+1)}{(k+1)\Gamma(n+k+2)}$$

Die letzte Zeile ist die Formel (5.19) für  $F(n, k)$  im Ausdruck der  $\Gamma$ -Funktion.  $\square$

### B.3 Grundlegendes für die Benutzung des Zeilberger-Algorithmus unter Maple

**Quelle B.4** (Maple-Paket EKHAD). Das Paket ist eine Implementierung des in Exkurs 5.20 beschriebenen Zeilberger-Verfahren. Die Implementierung ist von Zeilberger selbst vorgenommen. Das Paket ist über

<http://www.cis.upenn.edu/~wilf/AeqB.html>

frei verfügbar. Eine Kurzbeschreibung ist in [PWZ96, S. 113-114 bzw. S.200] zu finden. Für den Gegenstand der vorliegenden Dissertation ist die Funktion `ct` des Pakets vom Hauptinteresse. Das Web-Site des Pakets ist zugleich die Home-Page von [PWZ96]. Hier kann man [PWZ96] als pdf-Datei herunterladen.  $\square$



### B.3 Grundlegendes für die Benutzung des Zeilberger-Algorithmus unter Maple 155

*Beispiel B.5* (Zur Anwendung des Zeilberger-Algorithmus unter Maple). Im Folgenden ist die Maple-Sitzung für die Konstruktion des  $(F(n, k), G(n, k))$ -Paares in Beispiel 5.14 mit dem Zeilberger-Algorithmus des Pakets EKHAD ausgeführt.

```
> # Das Paket EKHAD einlesen
> read 'EKHAD';
> # Definition von F(n,k)
> F:=(n,k)->((k!)^2)/((n+k+1)!)^2;
```

$$F := (n, k) \rightarrow \frac{(k!)^2}{((n+k+1)!)^2}$$

```
> # Suche nach einer Rekursionsgleichung für F(n,k)
> ct(F(n,k),1,k,n,N);
```

$$-4n - 2 + (n+1)^3 N, 3n + 3 + 2k$$

```
> # Explizite Definition (Bindung) von G(n,k) anhand des vorigen
> # Schrittes
> G:=(n,k)->(3*n+2*k+3)*F(n,k);
```

$$G := (n, k) \rightarrow (3n + 2k + 3)F(n, k)$$

Die Ergebnis-Struktur der Funktion ct ist nach [PWZ96, S. 140] zu interpretieren. Demnach haben wir die Rekursionsgleichung (5.43) für das Paar  $(F(n, k), G(n, k))$  und die Formel (5.44) für  $G(n, k)$ . Die Korrelation der Ergebnis-Struktur zu der Rekursionsgleichung und der Formel (5.44) für  $G(n, k)$  ist durch das Beispiel leicht zu erkennen.  $\square$

# Literaturverzeichnis

- [Abd70] ABDALI, S. K.: ALGORITHM 393 — SPECIAL SERIES SUMMATION WITH ARBITRARY PRECISION. In: *Communications of the ACM* 13 (1970), September, Nr. 9, S. 570
- [Apé79] APÉRY, R.: Irrationalité de  $\zeta(2)$  et  $\zeta(3)$ . In: *Asterisque* 61 (1979), S. 11–13
- [AS72] ABRAMNOWITZ ; STEGUN: *Handbook of Mathematical Functions*. 9. 1972
- [AZ97] AMDEBERHAN, Tewodros ; ZEILBERGER, Doron: Hypergeometric Series Acceleration via the WZ Method. In: *The Electronic Journal of Combinatorics* 4 (1997), Nr. 2, S. #R3
- [BB92] BLÄSIUS, K. H. (Hrsg.) ; BACKHOUSE, R. C. (Hrsg.): *Deduktionssysteme - Automatisierung des logischen Denkens*. 2. München : Oldenbourg Verlag, 1992
- [BBB89] BORWEIN, J. M. ; BORWEIN, P. B. ; BAILEY, D. H.: Ramanujan, Modular Equations, and Approximations to Pi, or How to Compute One Billion Digits of Pi. In: *Amer. Math. Monthly* 96 (1989), S. 201–219
- [BBP97] BAILEY, D. H. ; BORWEIN, P. ; PLOUFFE, S.: On the Rapid Computation of Various Polylogarithmic Constants. In: *Math. Comput.* 66 (1997), S. 903–913
- [Ber95] BERNSTEIN, Daniel J.: *Detecting perfect powers in essentially linear time, and other studies in computational number theory*, University of California at Berkeley, Diss., 1995
- [Ber98] BERNSTEIN, Daniel J.: Detecting perfect powers in essentially linear time. In: *Mathematics of Computation* 67 (1998), S. 1253–1283
- [Do94] DO, Dang-Khoa. *Experimentelle Anwendung von HOL in Isabelle*. Großer Beleg. TU-Dresden, Fakultät Informatik, Institut für theoretische Informatik. 1994
- [Do01] DO, Dang-Khoa: Spigot Algorithm and Root Computing. In: *Reliable Computing* (2001), July, S. 247–273
- [Do04] DO, Dang-Khoa: Spigot Algorithm and Reliable Computation of Natural Logarithm. In: *Reliable Computing* 10 (2004), S. 489–500
- [Fab10] FABRY, E.: *Theorie des series a termes constants*. Paris, 1910
- [Gau00] GAUTHIER, B. *HYPERG, Maple package to manipulate hypergeometric series*. <http://www-igm.univ-mlv.fr/~gauthier/HYPERG.html>. 2000

- [Gos74] GOSPER, R. W. *Acceleration of Series*. M.I.T. Artificial Intelligence Memo # 304. March 1974
- [Gos78] GOSPER, R. W.: Decision procedure for indefinite hypergeometric summation. In: *Proc. Natl. Acad. Sci. USA* Bd. 75, 1978, S. 4042
- [HS85] HOARE, C.A.R. ; SHEPHERDSON, J.C.: *Mathematical Logic and Programming Languages*. Prentice Hall, 1985
- [JH84] J. NAAS (Hrsg.) ; H. L. SCHMID (Hrsg.): *Mathematisches Wörterbuch*. Bd. 1. Akademie-Verlag Berlin B.G. Teubner Stuttgart, 1984
- [Kno96] KNOPP, Konrad: *Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen mit besonderen Berücksichtigung der Anwendungsgebiete*. Bd. 2: *Theorie und Anwendung der unendlichen Reihen*. 5. Berlin · Göttingen · New York : Springer-Verlag, 1996
- [Knu97] KNUTH, D. E.: *The Art of Computer Programming*. Bd. 2, Seminumerical Algorithms. 3. Addison-Wesley Reading, 1997
- [Kum37] KUMMER: Eine neue Methode die numerischen Summen langsam convergierender Reihen zu berechnen. In: *Crelle's Journal* (1837)
- [Mar90] MARKOFF: Memoire sur la transformation des series peu convergentes en series tres convergentes. In: *Mem. de l' Acad. Imp. des Sciences de St. Peterbourg* (1890)
- [Mar96] MARKOFF, A.A.: *Differenzenrechnung*. Leipzig : Teubner, 1896
- [NS84] NAAS, J. (Hrsg.) ; SCHMID, H. L. (Hrsg.): *Mathematisches Wörterbuch*. Bd. 2. Akademie-Verlag Berlin B.G. Teubner Stuttgart, 1984
- [Poo79] VAN DER POORTEN, A.: A Proof that Euler missed..., Apéry's proof of the irrationality of  $\zeta(3)$ . In: *Math. Intel.* 1 (1979), S. 195–203
- [PWZ96] PETKOVSEK, M. ; WILF, H. S. ; ZEILBERGER, D.: *A = B*. Peters, Wellesley, Massachusetts (programs are available on WWW at <http://www.cis.upenn.edu/~wilf/AeqB.html>), 1996
- [Rei93] REICHEL, Horst. *Grundlage des maschinellen Beweisens*. Vorlesungscript, TU-Dresden Fakultät Informatik, Wintersemester 1993/1994. 1993
- [RW95] RABINOWITZ, Stanley ; WAGON, Stan: A Spigot Algorithm for the Digits of  $\pi$ . In: *American Mathematical Monthly* 102 (1995), March, Nr. 3, S. 195–203
- [Sal68] SALE, A. H. J.: The calculation of  $e$  to many significant digits. In: *Computer Journal* 11 (1968), August, Nr. 2, S. 229–230
- [Stö96] STÖLTING, Gregor: *Algorithmische Berechnung von Summen*, FU Berlin, Diplomarbeit, 1996
- [Sto97] STOSCHEK, Erwin P.: *Abenteuer Algorithmus*. Bd. 2. DRESDEN UNIVERSITY PRESS, 1997. – ISBN 3–931828–1
- [Wal92] WALTER, Wolfgang: *Analysis 1*. 3. Springer, 1992

- [Wei] WEISSTEIN, Eric W. *Kummer's Series Transformation*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/KummersSeriesTransformation.html>
- [Wil99] WILF, Herbert S.: Accelerated series for universal constants, by the WZ method. In: *Discrete Mathematics and Theoretical Computer Science* 3 (1999), September, S. 155–158
- [Zei82] ZEILBERGER, Doron: Sister Celines technique and its generalizations. In: *J. Math. Anal. Appl.* 85 (1982), S. 114145
- [Zei90a] ZEILBERGER, Doron: A fast algorithm for proving terminating hypergeometric identities. In: *Discrete Math.* 80 (1990), S. 207–211
- [Zei90b] ZEILBERGER, Doron: A holonomic systems approach to special functions identities. In: *J. of Computational and Applied Math* 32 (1990), S. 321368
- [Zei91] ZEILBERGER, Doron: The method of creative telescoping. In: *Journal of symbolic computation* 11 (1991), S. 195–204
- [Zei93] ZEILBERGER, D.: Closed Form. In: *Contemporary Mathematics* 143 (1993), S. 579–607
- [ZSMD03] ZSCHUNKE, T. ; SENECHAL, U. ; MISCKE, M. ; DO, D.-K.: Laplace-Transformation der chemischen Kinetik: Chancen und Probleme. In: *21. Deutscher Flammentag*. Düsseldorf : VDI-Verlag GmbH, 9 2003 (VDI-Berichte 1705), S. 715–720

# Symbolverzeichnis

<b>Allgemeine Symbole</b> .....	1	$\text{RNE}\left(\left[\left(\frac{a_i}{b_i}, c_i\right)\right]_{i=k+1}^{k+n}\right)$ : Kompaktnotation für Restausdruck, Definition 2.1	22
: Trennung zwischen Ausdruck und zusätzlicher Information.....	1	$\text{FREE}^{\text{ne}}(NE)$ : freie-Koeffizient-Projektion von NE, Definition 2.5	23
<i>Vereinheitlichung</i> : Tabelle 1 .....	2	$\text{REM}^{\text{ne}}(NE)$ : Restausdruck-Projektion von NE, Definition 2.5	23
$\underbrace{(\square)}_{\text{EqE}}$ : Gleichungsausdruck.....	3	$\text{COE}^{\text{ne}}(NE, pos)$ : Koeffizient-Projektion von NE, Definition 2.5	23
$\mathbb{Z}_{\geq i_0}$ : Indexmenge .....	3	$\text{FAC}^{\text{ne}}(NE, pos)$ : Faktor-Projektion von NE, Definition 2.5	23
$\mathbb{Z}_{i_0 \rightarrow i_1}$ : Indexmenge .....	3	$\text{NUM}^{\text{ne}}(NE, pos)$ : Faktor-Zähler-Projektion von NE, Definition 2.5	23
<b>Gleichheit, Folgenotaion, Substitution</b>	4	$\text{DEN}^{\text{ne}}(NE, pos)$ : Faktor-Nenner-Projektion von NE, Definition 2.5	23
$\stackrel{=}{val}$ : Wertgleichheit, Bemerkung 0.11 .....	4	$\text{COE}^{\text{ne}}_n(NE, pos)$ : Koeffizient-Projektion von NE, Definition 2.5	23
$\stackrel{=}{expr}$ : mathematische Ausdrucksgleichheit, Bemerkung 0.11.....	4	$\text{FAC}^{\text{ne}}_n(NE, pos)$ : Faktor-Projektion von NE, Definition 2.5	23
$\equiv$ : Meta-Gleichheit, Bemerkung 0.11 .....	5	$\text{NUM}^{\text{ne}}_n(NE, pos)$ : Faktor-Zähler-Projektion von NE, Definition 2.5	23
$\stackrel{\text{def}}{=}$ : Definitionszeichen, Bemerkung 0.12 ...	5	$\text{DEN}^{\text{ne}}_n(NE, pos)$ : Faktor-Nenner-Projektion von NE, Definition 2.5	23
$[E]_{i=n_1}^{n_2}$ : Folge mit Indexgrenzen, Bemerkung 0.13.....	5	$\text{SUB}^{\text{ne}}(NE, pos)$ : Teilsumme-Projektion von NE, Definition 2.5	23
$\stackrel{+}{seq}$ : Folgenverkettung, Bemerkung 0.13 ...	6	$\text{SUB}^{\text{ne}}_n(NE, pos)$ : Teilsumme-Projektion von NE, Definition 2.5	23
$[E]_i$ : Folge ohne Indexgrenzen, Definition 0.15.....	6	<b>FNE-Manipulation, Normierung</b> ....	25
$\stackrel{\text{sub}}{\longrightarrow}$ : Substitution, Bemerkung 0.17 .....	7	$\stackrel{+}{head}$ : linke Erweiterung einer FNE, (2.14)	25
$\stackrel{\text{shft}}{\longrightarrow}$ : Verschiebung, Spezialfall von Substitution, Bemerkung 0.17 .....	7	$\stackrel{+}{tail}$ : rechte Erweiterung einer FNE, (2.15)	26
<b>FNE-Basis-Notation</b> .....	21	$\stackrel{+}{ne}$ : Verkettung zweier FNE, (2.16)	26
$\left[\frac{a_i}{b_i}\right]_{i=k+1}^{k+n}$ : Faktorsystem von FNE.....	21		
$[c_i]_{i=k}^{k+n}$ : Koeffizientenfolge von FNE .....	21		
$c_k + \text{RNE}\left(\left[\left(\frac{a_i}{b_i}, c_i\right)\right]_{i=k+1}^{k+n}\right)$ : Kompaktnotation für FNE, Definition 2.1.....	21		

<sup>reg-set</sup>  
**FNE** ( $FS$ ): Menge der normierten FNEs eines FS, (2.22) ..... 28

<sup>reg-set</sup>  
**RNE** ( $FS$ ): Menge der normierten Restausdrücke eines FS, (2.23) ..... 28

<sup>ne</sup>  
**CARRY** ( $NE, pos$ ): Versetzung des Übertrags, Definition 2.19 ..... 29

<sup>ne</sup>  
**NORM** ( $NE$ ): normierte Form einer FNE, Definition 2.21 ..... 30

<sup>norm</sup>  
**COE** ( $NE, i$ ): Normierungskoeffizient, Definition 2.21 ..... 30

<sup>norm</sup>  
**Carry** ( $NE, i$ ): Normierungsübertragszahl, Definition 2.21 ..... 30

<sup>ne</sup>  
 $\times_{coe}$ : Multiplikation einer FNE mit einer Konstanten, (2.41) ..... 35

**FNE und Spigot-Algorithmus** ..... 36

<sup>spigot</sup>  
<sup>fne</sup>  
**APR** ( $NE, g, i$ ):  
 Spigot-Approximierungssumme einer FNE, Definition 2.31 ..... 36

<sup>spigot</sup>  
<sup>fne</sup>  
**RNE** ( $NE, g, i$ ): Spigot-Resta Ausdruck, Definition 2.31 ..... 36

<sup>spigot</sup>  
<sup>fne</sup>  
**APN** ( $NE, g, i$ ): Spigot-Approximation, (2.60) 40

**INE und Spigot-Algorithmus** ..... 45

$\left[ \frac{a_i}{b_i} \right]_{i=k+1}^{\infty}$ : Faktorsystem einer INE, Definition 2.44 ..... 45

$[c_i]_{i=k}^{\infty}$ : Koeffizientenfolge einer INE, Definition 2.44 ..... 45

$c_k + \text{RNE} \left( \left[ \left( \frac{a_i}{b_i}, c_i \right) \right]_{i=k+1}^{\infty} \right)$ : Kompaktnotation einer INE, Definition 2.44 .. 45

<sup>fne</sup>  
<sup>ine</sup>  
**APR** ( $INE, n$ ): fne-Approximation einer INE, (2.73) ..... 45

<sup>+</sup>  
<sup>head</sup>: linke Erweiterung einer INE, (2.79) 46

<sup>+</sup>  
<sup>ine</sup>: Verkettung einer FNE mit einer INE, (2.80) ..... 47

<sup>spigot</sup>  
<sup>ine</sup>  
**APR** ( $INE, g, n, i$ ):

Spigot-Approximierungssumme einer INE, (2.83) ..... 47

<sup>spigot</sup>  
<sup>ine</sup>  
**RNE** ( $INE, g, n, i$ ): Spigot-Resta Ausdruck einer INE, (2.84) ..... 47

<sup>spigot</sup>  
<sup>ine</sup>  
**APN** ( $INE, g, n, i$ ): Spigot-Approximation einer INE, (2.85) ..... 47

<sup>spigot</sup>  
<sup>ine</sup>  
**ERR** ( $INE, g, n, i$ ): absoluter Fehler in Bezug auf eine INE, (2.86) ..... 48

<sup>cnv</sup>  
<sup>ine</sup>  
**ERR** ( $INE, g, n, i$ ): Konvertierungsfehler in Bezug auf eine INE, (2.87) ... 48

<sup>appr</sup>  
<sup>ine</sup>  
**ERR** ( $INE, n$ ): Approximationsfehler in Bezug auf eine INE, (2.88) ..... 48

**Schrankenfunktionen** ..... 49

$\square_{s\text{-ifs}}$   
 $\Delta$  ( $IFS$ ) ( $m, n$ ): funktionale Spigot-Schranke eines IFS, (2.90) ..... 49

$\min_{s\text{-ifs}}$   
 $\Delta$  ( $IFS$ ) ( $m, n$ ): minimale funktionale Spigot-Schranke eines IFS, (2.91) ..... 49

$\square_{r\text{-ifs}}$   
 $\Delta$  ( $IFS$ ) ( $n$ ): funktionale Restschranke eines IFS, (2.95) ..... 50

$\square_{r\text{-ine}}$   
 $\Delta$  ( $INE$ ) ( $n$ ): funktionale Restschranke einer INE, (2.96) ..... 50

**Charakteristische Funktionen** ..... 52

$\square_{spigot}$   
 $\chi$  ( $INE, g$ ) ( $l$ ): charakteristische Funktion der Spigot-Berechnung in Bezug auf eine INE, (2.101) ..... 52

$\square_{spigot}$   
 $\Delta$  ( $IFS, g$ ) ( $n$ ): charakteristische Schrankenfunktion der Spigot-Berechnung in Bezug auf ein IFS, Definition 2.63 52

**Ressourcenverbrauch** ..... 53

<sup>norm</sup>  
<sup>carry</sup>  
 $\epsilon_t$  ( $NE, g$ ): *normierter Rechenaufwand* der Spigot-Versetzung, (2.102),(2.103) 53

<sup>norm</sup>  
<sup>carry</sup>  
 $\epsilon_t$  ( $NE, 1$ ):  $\sim$  der Spigot-Normierung, (2.102),(2.103) ..... 53

$\epsilon_t^{\text{norm}}$ (g): $\sim$ der g-Normierung, g-carry (2.102).....	53	Schrankenfunktion der Spigot-Berechnung in Bezug auf $\overset{\text{sim}}{\text{INE}}$ zur Basis 10, (2.113) .....	60
$\epsilon_m^{\text{norm}}$ (NE): <i>normierter Speicherbedarf</i> für num Zähler,(2.104).....	54	$\overset{\text{sim}}{\chi}_{\text{spigot}} \left( \overset{\text{sim}}{\text{INE}}, 10^m \right) (l)$ : eine charakteristische Funktion der Spigot-Berechnung für $\sqrt{2}$ in Bezug auf $\overset{\text{sim}}{\text{INE}}$ zur Basis $10^m$ , (2.115).....	61
$\epsilon_m^{\text{norm}}$ (NE): $\sim$ für Nenner, (2.104) .....	54	$\overset{\text{sim}}{\Delta}_{\text{spigot}} \left( \overset{\text{sim}}{\text{IFS}}, 10^m \right) (n)$ : eine charakteristische Schrankenfunktion der Spigot-Berechnung in Bezug auf $\overset{\text{sim}}{\text{IFS}}$ zur Basis $10^m$ , (2.116) .....	61
$\epsilon_m^{\text{norm}}$ (NE): $\sim$ für Koeffizienten, (2.104) .	54	<b>Gosperscher <math>\mathcal{R}</math>-Ausdruck</b> .....	107
$\epsilon_m^{\text{norm}}$ (NE, g): $\sim$ für Ergebnisziffern, (2.104) result	54	$b_i \overset{i_2}{\mathcal{R}} r_i$ : allgemeiner endlicher $\mathcal{R}$ -Ausdruck, Definition 6.1.....	107
$\epsilon_t^{\text{total}}$ (NE, g, n, i): gesamter Rechenaufwand spigot für die Berechnung einer Spigot-Approximation, (2.103) ..	54	$\overset{i_2}{\mathcal{R}} r_i$ : endlicher Implizit-1- $\mathcal{R}$ -Ausdruck, Definition 6.1.....	107
$\epsilon_m^{\text{total}}$ (NE, g, n, i): gesamte Speicherbedarf spigot für die Berechnung einer Spigot-Approximation, (2.104) ..	54	$b_i \overset{\infty}{\mathcal{R}} r_i$ : allgemeiner unendlicher $\mathcal{R}$ -Ausdruck, Definition 6.1 .....	107
$\epsilon_t^{\text{norm}}$ (g, m): normierter Rechenaufwand der p-back Potenzrückführung.....	56	$\overset{\infty}{\mathcal{R}} r_i$ : unendlicher Implizit-1- $\mathcal{R}$ -Ausdruck, Definition 6.1.....	107
<b>einfache Spigot-Berechnung von <math>\sqrt{2}</math></b>	57		
$\overset{\text{sim}}{\text{INE}}$ : einfache INE für $\sqrt{2}$ , (2.106) .....	57		
$\overset{\text{sim}}{\text{IFS}}$ : einfache IFS für $\sqrt{2}$ , (2.108).....	57		
$\overset{\text{sim}}{\Delta}_{\text{r-ifs}} \left( \overset{\text{sim}}{\text{IFS}} \right) (n)$ : eine konstante Restschranke für $\overset{\text{sim}}{\text{IFS}}$ , (2.108).....	57		
$\overset{\text{sim}}{\Delta}_{\text{r-ine}} \left( \overset{\text{sim}}{\text{INE}} \right) (n)$ : eine Restschranke für $\overset{\text{sim}}{\text{INE}}$ , (2.109).....	58		
$\overset{\text{sim}}{\Delta}_{\text{s-ifs}} \left( \overset{\text{sim}}{\text{IFS}} \right) (m, n)$ : eine Spigot-Schranke für $\overset{\text{sim}}{\text{IFS}}$ , (2.110).....	58		
$\overset{\text{sim}}{\chi}_{\text{spigot}} \left( \overset{\text{sim}}{\text{INE}}, 10 \right) (l)$ : eine charakteristische Funktion der Spigot-Berechnung für $\sqrt{2}$ in Bezug auf $\overset{\text{sim}}{\text{INE}}$ zur Basis 10, (2.112).....	59		
$\overset{\text{sim}}{\Delta}_{\text{spigot}} \left( \overset{\text{sim}}{\text{IFS}}, 10 \right) (n)$ : eine charakteristische			

# Index

## A

- adisch,  $g$ - $\sim$  Darstellung einer Zahl.....27
- Adjungierte
  - linke und rechte  $\sim$  des Markoff-Paares  
92
- Algorithmus
  - Ergebnisnormierungs $\sim$ ..... 40
  - Extraktions $\sim$ ..... 37
  - Gosper- $\sim$ .....102
  - Potenzrückführungs $\sim$ ..... 55
  - Spigot- $\sim$ .....40
  - Zeilberger- $\sim$ ..... 103
- Anfangsindex
  - $\sim$  einer geschachtelten Summe ..... 21
  - Restreihen $\sim$  der Gosperschen Reihentransformation.....111
  - Term $\sim$  der Gosperschen Reihentransformation ..... 111
- anschauliche Form
  - $\sim$  einer geschachtelten Summe ..... 24
- Approximation
  - fne- $\sim$  vom Grad  $n$  einer INE..... 45
  - $\sim$ sfehler der Spigot-Berechnung ..... 48
  - $\sim$ sgrad der Spigot-Berechnung ..... 48
- atomare Speichereinheiten
  - $\sim$  des Spigot-Algorithmus..... 54
- atomarer Schritt
  - $\sim$  der Ergebnisnormierung ..... 53
  - $\sim$  der Potenzrückführung ..... 56
  - $\sim$  der Spigot-Versetzung ..... 53

## B

- Bernoullische Zahl ..... 72

## C

- charakteristisch
  - $\sim$ e Funktion der Spigot-Berechnung (in Bezug auf eine INE).....52
  - $\sim$ e Schrankenfunktion der Spigot-Berechnung in Bezug auf ein

- IFS..... 52

- Methode des Koeffizientenvergleiches in  $\sim$ en Funktionalgleichungen für die Reihenentwicklung..... 64

## D

- Differenz
  - $\sim$  von Folgen ..... 75
  - Mehrfache  $\sim$  von Folgen ..... 77
  - Reihentransformation durch  $\sim$  und Doppelreihe ..... 75–83
- Doppelreihe
  - Reihentransformation durch Differenz und  $\sim$ .....75–83

## E

- Ergebnisnormierung
  - $\sim$ salgorithmus ..... 40
  - Spezialfall des Spigot-Algorithmus ohne  $\sim$  ..... 55
- Erweiterung
  - linke  $\sim$  einer FNE ..... 25
  - linke  $\sim$  einer INE ..... 46
  - rechte  $\sim$  einer FNE ..... 26
- Euler
  - allgemeine Form der  $\sim$ schen Reihentransformation ..... 81
  - gewöhnliche Anwendungsform der  $\sim$ schen Reihentransformation ..... 83
- expandierte
  - $\sim$  Form einer FNE.....24
  - $\sim$  Reihe einer INE..... 45
- Extraktion
  - $\sim$ ssumme der Gosperschen Reihentransformation.....109
- Extraktionsalgorithmus ..... 37

## F

- Faktorsystem
  - endliches  $\sim$  ..... *siehe* FFS
  - homogenes  $\sim$  ..... 27



- kompatible  $\sim e$  ..... 131  
 unendliches  $\sim$  ..... *siehe* IFS
- FFS ..... 21  
 größter normierter Restausdruck eines  $\sim$   
 41
- FNE ..... **21**, 21–44  
 Anfangsindex einer  $\sim$  ..... 21  
 Faktor an der Position  $pos$  einer  $\sim$  .. 21  
 Faktorsystem einer  $\sim$  ..... 21  
 freier Koeffizient einer  $\sim$  ..... 21  
 Grad einer  $\sim$  ..... 21  
 Koeffizient an der Position  $pos$  einer  $\sim$   
 21  
 Koeffizientenfolge einer  $\sim$  ..... 21  
 Konstruktionskette einer  $\sim$  ..... 21  
 Nenner an der Position  $pos$  einer  $\sim$  . 21  
 Teilsumme an der Position  $pos$  einer  $\sim$   
 21  
 Zähler an der Position  $pos$  einer  $\sim$  .. 21
- fne-  
 einfache  $\sim$ -Form einer Summe ..... 27  
 einfache  $\sim$ -Transformation einer Summe  
 26
- fne –  
 $\sim$ -Approximation vom Grad  $n$  einer INE  
 45
- Folgen  
 $\sim$ -notation mit Indexgrenzen ..... 5  
 $\sim$ -notation ohne Indexgrenzen ..... 6  
 $\sim$ -verkettung ..... 6
- freier Koeffizient (einer NE) ..... 21
- G**
- $g$ -adisch  
 $\sim$ -er Ausdruck ..... 27  
 $\sim e$  Darstellung ..... 27  
 $\sim e$  Entwicklung ..... 27
- geschachtelt  
 einfache  $\sim e$  Form einer Summe ... *siehe*  
 fne-Form  
 endlich  $\sim e$  Summe ..... *siehe* FNE  
 unendlich  $\sim e$  Summe ..... *siehe* INE
- Gleichheit  
 mathematische Ausdrucks $\sim$  ..... 4  
 Meta- $\sim$  ..... 5  
 Wert $\sim$  ..... 4
- Gosper  
 $\sim$ -Algorithmus zur geschlossenen  
 Summation hypergeometrischer  
 Terme ..... 102
- $\sim$ -sche Reihentransformation ..... 105
- Grad  
 fne-Approximation vom  $\sim e$   $n$  einer INE  
 45  
 Approximations $\sim$  der Spigot-Berechnung  
 48  
 einer FNE ..... 21
- H**
- hypergeometrisch  
 $\sim$ -er Term ..... 101
- I**
- IFS ..... 45  
 charakteristische Schrankenfunktion der  
 Spigot-Berechnung in Bezug auf ein  
 $\sim$  ..... 52
- Index  
 Anfangs $\sim$  einer FNE ..... 21  
 $\sim$ -obergrenze in der Folgennotation ... 5  
 $\sim$ -untergrenze in der Folgennotation .. 5  
 $\sim$ -variable in der Folgennotation ..... 5
- INE ..... 45  
 fne-Approximation vom Grad  $n$  einer  $\sim$   
 45
- ine-  
 $\sim$ -Darstellung einer reellen Zahl ..... 45  
 einfache  $\sim$ -Transformation einer  
 Reihe ..... 46
- Invarianz, grundlegende  $\sim$   
 des Spigot-Algorithmus ..... 36
- Iterationszahl,  
 $\sim$  der Spigot-Berechnung ..... 48
- K**
- Koeffizientenfolge  
 $\sim$  einer FNE ..... 21  
 $\sim$  einer INE ..... 45
- Koeffizientenvergleich  
 Methode des  $\sim e$ s in charakteristischen  
 Funktionalgleichungen für die Rei-  
 henentwicklung ..... 64
- Kombinationsquotient,  $\sim$  der Gosperschen  
 Reihentransformation ..... 111
- Kompaktnotation  
 $\sim$  einer FNE ..... 21  
 $\sim$  einer INE ..... 45
- Konstruktionskette (einer FNE) ..... 21
- Konvertierungsfehler,  
 $\sim$  der Spigot-Berechnung ..... 48

- Kummer  
 ~sche Reihentransformation.....86  
 ~sche Transformationsfolge.....86  
 unnormierte ~sche  
 Transformationsfolge.....86
- L**
- Logarithmenberechnung.....131  
 Prozeduren der ~.....136  
 Restschranken für ~.....133–134  
 Spigot-Schranken für ~.....134  
 Zerlegungsgleichung der ~.....131
- M**
- Markoff  
 ~-Paar.....92  
 ~sche Rechtecktransformation.....91
- Multiplikation  
 einer FNE mit einer Konstanten....35
- N**
- normiert, im Sinne von regulär  
 ~e FNE.....28  
 ~er RNE.....28  
 ~e Form einer FNE.....30
- normiert, im Sinne von durchschnittlich  
 ~er Rechenaufwand eines elementaren  
 Schrittes  
 des Spigot-Algorithmus.....53  
 ~er Speicherbedarf einer Speicherzelle  
 des Spigot-Algorithmus.....54
- Normierung  
 ~salgorithmus.....31  
 ~sfolge einer FNE.....30  
 ~skoeffizient einer FNE.....30  
 ~übertragszahl einer FNE.....30
- P**
- Positionszahl, ~ in einer NE.....21
- Potenzüberbrückung.....56  
 Potenzrückführungsalgorithmus nach  
 der ~.....55  
 Spigot-Algorithmus mit ~.....56
- Q**
- Quotientenkriterium  
 Restschranke durch ~.....50
- R**
- Rabinowitz, Spigot-Berechnung von  $\pi$  nach  
 ~ und Wagong.....15
- Reihe  
 ~ für  $\frac{x}{\exp(x)-1}$ .....72  
 ~ für  $\tan x$ .....73  
 ~ mit durch Bernoullische Zahlen dar-  
 gestellten Koeffizienten.....72  
 Binomial~.....70  
 Exponential~.....65  
 Gregory-Halley-~.....68  
 Gregory-Leibniz-~ für  $\pi$ .....70  
 harmonische~ mit alternierenden Vor-  
 zeichen.....68  
 Leibniz-~ für  $\arctan x$ .....69  
 logarithmische ~.....66  
 Mercatorsche ~.....67
- Reihenentwicklung  
 Methode des Koeffizientenvergleiches in  
 charakteristischen  
 Funktionalgleichungen für die ~.64
- Reihentransformation  
 ~ durch Differenz und Doppelreihe.75–  
 83  
 ~ mit Markoffscher Rechtecktransfor-  
 mation.....91  
 ~ mit Wilfscher Methode.....97  
 ~ mit Zeilbergerscher Dreiecktransfor-  
 mation.....94  
 ~ mit zweistelliger Durchschnittsbildung  
 84  
 allgemeine Form der Eulerschen ~..81  
 gewöhnliche Anwendungsform der Eu-  
 lerschen ~.....83  
 Gospersche ~.....105  
 Kummersche ~.....86  
 Wertverschiebung als Grundbaustein  
 der ~.....83–86
- Restausdruck  
 ~ einer FNE.....21  
 größter normierte ~ eines FFS....41
- Restreihe  
 ~ssumme der Gosperschen Reihentrans-  
 formation.....109
- Restschranke  
 ~ durch Rundungslemma.....51  
 ~n für Logarithmenberechnung133–134  
 ~ für Wurzelberechnung.....121  
 eines FFS.....41  
 funktionale ~ einer INE.....50  
 funktionale ~ eines IFS.....50
- Restsumme (einer FNE).....21

- S**
- Sale, die Idee der Spigot-Berechnung von  $\sim$  11
- Spaltungsfunktion,  $\sim$  der Wertverschiebung 85
- Spigot
- $\sim$ -Ansatz ..... 19
- Spigot-Algorithmus ..... 40
- Grundgleichung des  $\sim$  ..... 40
  - grundlegende Invarianz des  $\sim$  ..... 36
  - normierter Rechenaufwand eines elementaren Schrittes des  $\sim$  ..... 53
  - normierter Speicherbedarf einer Speicherzelle des  $\sim$  ..... 54
- Spigot-Ansatz ..... 62
- Spigot-Approximation
- (der Iterationszahl  $i$  vom Grad  $n$  zur Basis  $g$ ) einer INE ..... 47
  - (der Iterationszahl  $i$  zur Basis  $g$ ) einer FNE ..... 39
- Spigot-Approximationssumme
- (der Iterationszahl  $i$  zur Basis  $g$ ) einer FNE ..... 36
- Spigot-Berechnung ..... 48
- $\sim$  von  $\pi$  nach Rabinowitz und Wagong 15
  - $\sim$  von Logarithmen ..... *siehe* Logarithmenberechnung
  - $\sim$  von Wurzeln *siehe* Wurzelberechnung
  - Approximationsfehler der  $\sim$  ..... 48
  - Approximationsgrad der  $\sim$  ..... 48
  - Besonderheit der  $\sim$  von  $\pi$  ..... 15
  - charakteristische Funktion einer  $\sim$  .. 52
  - die  $\sim$  von  $\sqrt{2}$  nach Stoschek ..... 57
  - die Idee der  $\sim$  von Sale ..... 11
  - Iterationszahl der  $\sim$  ..... 48
  - Konvertierungsfehler der  $\sim$  ..... 48
  - Schrankenfunktion einer  $\sim$  ..... 52
- Spigot-Resta Ausdruck
- (der Iterationszahl  $i$  zur Basis  $g$ ) einer FNE ..... 36
- Spigot-Schranke ..... 49
- $\sim$  für Wurzelberechnung ..... 121
  - $\sim$  für Logarithmenberechnung ... 134
  - funktionale  $\sim$  eines IFS ..... 49
  - minimale funktionale  $\sim$  eines IFS ... 49
- Stoschek, die Spigot-Berechnung von  $\sqrt{2}$  nach  $\sim$  ..... 57
- T**
- Teilsumme (an der Position  $pos$ ) einer FNE 21
- U**
- Übertrag
- $\sim$ szahl der Versetzung ..... 29
  - Versetzung des  $\sim$ s ..... 28
- V**
- Verkettung
- einer FNE mit einer INE ..... 47
  - zweier FNE ..... 26
- W**
- Wagong, Spigot-Berechnung von  $\pi$  nach Rabinowitz und  $\sim$  ..... 15
- Wertanteil an der Position  $pos$  (einer NE) 24
- Wilf,  $\sim$ sche Methode der Reihentransformation ..... 97
- Wurzelberechnung ..... 120
- BINE für die  $\sim$  ..... 121
  - einfaches Potenzkomplement für die  $\sim$  124
  - Grundgleichung für  $\sim$  ..... 120
  - Hilfsfaktor und Abweichungsradius der  $\sim$  ..... 120
  - Potenzkomplement für die  $\sim$  ..... 124
  - Restschranke für  $\sim$  ..... 121
  - Spigot-Schranke für  $\sim$  ..... 121
- WZ-Paar ..... 101
- Z**
- Zeilberger
- $\sim$ -Algorithmus ..... 103
  - $\sim$ sche Dreieckstransformation ..... 94

