Beiträge zur Algorithmik verallgemeinerter diskreter Fouriertransformationen

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Meinard Müller

aus

Böblingen

Bonn, 01. März 2001

Beiträge zur Algorithmik verallgemeinerter diskreter Fouriertransformationen

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Meinard Müller

aus

Böblingen

Bonn, 01. März 2001

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Referent: Prof. Dr. Michael Clausen
2. Referent: Priv.-Doz. Dr. Mohammad Amin Shokrollahi

Tag der Promotion: 04.07.2001

Inhaltsverzeichnis

Symbolverzeichnis

G	endliche Gruppe, 7
g	Gruppenelement in G, 7
D	Darstellung, 7
χ	Charakter, 7
$\langle \Delta \vert D \rangle$	Multiplizität von Δ in D, 8
$\mathrm{Int}(D, D')$	Verkettungsraum zweier Darstellungen D und D', 8
$D{\downarrow}H$	Einschränkung einer Darstellung D auf eine Untergruppe H, 8
$F{\uparrow}G$	Induktion einer Darstellung F nach G, 8
$N \lhd G$	Normale Untergruppe N von G, 9
F^g	g-konjugierte Darstellung, 9
$\mathbb{C}G$	Gruppenalgebra über dem Körper der komplexen Zahlen $\mathbb{C}$, 9
$\langle \cdot \vert \cdot \rangle$	Skalarprodukt auf $\mathbb{C}G$, 9
C_N	zyklische Gruppe der Ordnung N, 10
h	Anzahl der Konjugationsklassen der Gruppe G, 10
$D = \oplus_{k=1}^{h} D_k$	diskrete Fouriertransformation, 10
$\mathbf{D}$	DFT-Matrix, 10
$\mathrm{Irr}(G)$	Transversale irreduzibler Darstellungen, 10
$\mathcal{T}$	Kompositionsreihe, 11
n	Länge der Kompositionsreihe, 11
G_i	i-te Gruppe der Kompositionsreihe $\mathcal{T}$, 11
g_i	i-ter Erzeuger der Gruppe G, 11
$[g_i, g_j]$	Kommutator $(g_i^{-1} g_j^{-1} g_i g_j)$, 11
G_{128}	2-Gruppe der Ordnung 128, 11
$\mathcal{T}_i$	Kompositionsreihe für G_i, 13
$\mathrm{Irr}(G_i, \mathcal{T}_i)$	Transversale von $\mathcal{T}_i$-angepaßten irreduziblen Darstellungen von G_i, 13
$\mathbb{K}$	Körper, 15
S_N	symmetrische Gruppe der Ordnung N, 15
$\mathrm{diag}(c_1, \ldots, c_d)$	Diagonalmatrix mit Einträgen $c_1, \ldots, c_d$ auf der Hauptdiagonalen, 15
$\log$	Logarithmus zur Basis 2, 16
$d^r(G)$	Kurzschreibweise für $\sum_{k=1}^{h} d_k^r$, 16
e	Exponent der Gruppe G, 17
$\mathbb{Z}_e$	Restklassenring $\mathbb{Z}/e\mathbb{Z}$, 17
inn_h	durch h-Konjugation definierter innerer Automorphismus, 25
$\mathrm{Lib}_N(k)$	k-te Gruppe der Ordnung N der Bibliothek, 26

$e(\mathcal{T})$	Exponent der Kompositionsreihe $\mathcal{T}$, 30
$C(w)$	Komplexität des Wortes w, 34
$\mathbb{K}[X_n, \ldots, X_1]$	Polynomring in n Unbestimmten über dem Körper $\mathbb{K}$, 45
$\mathbb{K}[X]$	Multiindexschreibweise für $\mathbb{K}[X_n, \ldots, X_1]$, 45
$\deg(f)$	Grad des Polynoms f, 45
$\succ_{\text{lex}}$	lexikographische Ordnung, 45
$\text{multideg}(f)$	Multigrad des Polynoms f, 45
$\text{LC}(f)$	führender Koeffizient des Polynoms f, 45
$\text{LM}(f)$	führendes Monom des Polynoms f, 45
$\text{LT}(f)$	führender Term des Polynoms f, 45
$\mathcal{G}$	Gröbnerbasis, 46
X^*	das von der Menge X erzeugte freie Monoid, 50
$\succ_X \wr \succ_Y$	Kranzprodukt der Ordnungen $\succ_X$ und $\succ_Y$, 51
$L(G)$	lineare Komplexität der Gruppe G, 54
$L_2(G)$	2-lineare Komplexität der Gruppe G, 54
A	Ring mit Einselement 1, 68
ϵ	Idempotent, 68
$\mathcal{P}(A)$	Menge der Partitionen von A, 68
$\text{End}_A(M)$	Endomorphismenring des A-Moduls M, 69
$M(V)$	M-isotypische Komponente des A-Linksmoduls V, 70
$\text{irr}(G)$	Menge der irreduziblen Charaktere von G, 72
$\text{W}(\chi)$	Menge von in χ endenden Wegen im Charaktergraph von G, 72
$e(w)$	primitives Idempotent zum Weg w, 72
$(\cdot \mid \cdot)$	Standard-Skalarprodukt auf $\mathbb{C}G \simeq \mathbb{C}^{\lvert G\rvert}$, 76
$\hat{f}_{k\mu\nu}$	Fourierkoeffizient des Signals f, 78
D_p	Diedergruppe der Ordnung $2 \cdot p$, 83
$L^2(X)$	Hilbertraum der komplexen Signale über der Menge X, 86
$L^2(G, H)$	Raum der auf den Linksnebenklassen von H konstanten Signale, 87
$\pi_{(G,H)}$	Radonabbildung, 87
V_i	Approximationsraum der i-ten Stufe einer MRA, 88
W_i	Detailraum der i-ten Stufe einer MRA, 88
$\text{Irr}(G, G_i)$	Teilmenge der auf G_i trivialen Darstellungen von $\text{Irr}(G)$, 93
$\text{irr}(G, G_i)$	Menge der zu $\text{Irr}(G, G_i)$ gehörigen Charaktere, 93

Kapitel 1

Einleitung

Seit dem Einsatz von Computern in nahezu allen Bereichen des Lebens, insbesondere der Naturwissenschaften, gewinnt der Entwurf von Algorithmen zur Lösung von Berechungsproblemen immer mehr an Bedeutung. Neben der Frage der prinzipiellen Lösbarkeit der Probleme spielt dabei die *Effizienz* der Algorithmen eine immer wichtigere Rolle. Die *diskrete Fouriertransformation* (DFT) ist hierfür eine Paradebeispiel, insbesondere auch aufgrund ihrer interdisziplinären Stellung zwischen den Ingenieurwissenschaften, der Informatik und der reinen Mathematik. Zum einen hatte die Entdeckung der *schnellen Auswertbarkeit der DFT* (Fast Fourier Transform, FFT) einen entscheidenden Einfluß auf die Entwicklung der gesamten digitalen Signalverarbeitung. Zum anderen ermöglicht eine darstellungstheoretische Sichtweise eine Verallgemeinerung des DFT-Begriffs, was unmittelbar in den Bereich der algorithmischen Gruppentheorie und Computeralgebra führt. Die Generierung von DFTs ist dann gleichbedeutend mit der Konstruktion einer Transversalen irreduzibler Darstellungen.

Beide Sichtweisen der diskreten Fouriertransformation, die mathematische und die signaltheoretische, werden in der vorliegenden Arbeit eingenommen und dabei in eine für beide Disziplinen fruchtbare Wechselbeziehung gebracht. Zum einen geht es um den Entwurf neuer, effizienter Algorithmen, z. B. zur effizienten Konstruktion irreduzibler Darstellungen auflösbarer Gruppen (Kapitel 8) und zur schnellen Wortnormalisierung in überauflösbaren Gruppen (Kapitel 4). Zum anderen werden in der Theorie bereits bestehende Algorithmen zur Generierung verallgemeinerter DFTs und deren schnelle Auswertung erstmals in lauffähige Computerprogramme umgesetzt (Kapitel 3 und 5). Erst die Implementierung der scheinbar nur für die Mathematik interessanten Algorithmen ermöglicht wiederum Anwendungen in der digitalen Signalverarbeitung (Kapitel 7). Kapitel 6 stellt die Theorie der diskreten Fouriertransformationen systematisch in Form von Idempotenten dar und verbindet - über die verallgemeinerten Spektralzerlegungen - Algebra und Signalverarbeitung.

Hintergrund

Die *klassische schnelle Fouriertransformation* (Fast Fourier Transform, FFT) hat in einem Maße wie kaum ein anderer moderner Algorithmus die Entwicklung ganzer Industriezweige beeinflußt. „Whole industries are changed from slow to fast by this one idea - which is pure

mathematics", schreibt Gilbert Strang in [60], S. 290. So verwenden z. B. fast alle modernen Verfahren zur Bild- oder Audiokompression FFT-basierte Methoden.

Die Grundidee der FFT geht auf Carl Friedrich Gauß zurück, der bereits 1805 einen solchen Algorithmus benutzte, um Asteroidenorbits zu interpolieren [32]. Im Jahr 1965 wurde die FFT durch Cooley und Tukey [18] (wieder-)entdeckt und zur Analyse von Zeitreihen eingesetzt. Durch eine geschickte Faktorisierung der DFT-Matrix

$$\mathbf{D}_N = (\omega^{jk})_{0 \le j,k < N},$$

für *gewisse* $N \in \mathbb{N}$ und einer primitiven N-ten Einheitswurzel ω, konnte die Auswertung der DFT auf einem Vektor der Länge N mit $O(N \log N)$ arithmetischen Operationen durchgeführt werden - im Vergleich zur naiven Matrix-Vektor-Multiplikation, die für diese Aufgabe $O(N^2)$ Operationen benötigt. In Arbeiten von Rader (1968), [49], und Bluestein (1970), [9], wurde die FFT auf *beliebige* $N \in \mathbb{N}$ verallgemeinert und damit das Problem der klassischen Fouriertransformation weitgehend gelöst.

Aus *algebraischer Sicht* ist die Berechnung der DFT der Länge N gleichbedeutend mit der Auswertung eines vollen Satzes an paarweise inäquivalenten irreduziblen Darstellungen der zyklischen Gruppe C_N der Ordnung N. Der Satz von Wedderburn zur Zerlegung von halbeinfachen Algebren über Zerfällungskörpern ermöglicht eine Verallgemeinerung der DFT auf beliebige endliche Gruppen G: Nach diesem Satz ist die komplexe Gruppenalgebra $\mathbb{C}G := \{a \,|\, a : G \to \mathbb{C}\}$ (Signalraum) isomorph zu einer Algebra von Blockdiagonalmatrizen (Spektralbereich),

$$D = \oplus_{k=1}^{h} D_k : \mathbb{C}G \longrightarrow \oplus_{k=1}^{h} \mathbb{C}^{d_k \times d_k}.$$

Die Anzahl h der Blöcke entspricht dabei der Anzahl der Konjugationsklassen von G, und die Projektionen $D_1, \ldots, D_h$ bilden einen vollständigen Satz (Transversale) an paarweise inäquivalenten irreduziblen Darstellungen von $\mathbb{C}G$. Jeder solche Isomorphismus D wird DFT von G genannt. Bezüglich dieser verallgemeinerten diskreten Fouriertransformationen einer endlichen Gruppe G gibt es zwei grundlegende Berechnungsprobleme:

(1) Wie kann eine DFT für G *effizient erzeugt* werden? Ist G nicht abelsch, dann gibt es unendlich viele DFTs. Da wir in der vorliegenden Arbeit an einer schnellen *Generierung* eines Isomorphismus $D = \oplus D_k$ interessiert sind, müssen die Vertreter D_k der Äquivalenzklassen von Darstellungen sorgfältig gewählt werden.

(2) Gibt es eine *geeignete* DFT von G, die eine *effiziente Auswertung* derselben auf einem Vektor der Länge $N = |G|$ erlaubt? Mit anderen Worten, es müssen, falls möglich, die Darstellungen in (1) so gewählt werden, daß ein Algorithmus zur DFT-Auswertung angegeben werden kann, der mit weniger als $O(N^2)$ Operationen, der trivialen oberen Schranke, auskommt. Jeder Algorithmus, der dies in $\Theta(N \log N)$ - welches in einem geeigneten Komplexitätsmodell eine untere Schranke für das Auswertungsproblem darstellt - bewerkstelligt, kann dann mit Recht eine FFT genannt werden.

Besitzt die endliche Gruppe G eine Normalreihe der Form

$$\mathcal{T} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_1 \rhd G_0 = \{1\}),$$

dann liefert das Konzept der sogenannten *Symmetrieanpassung*, d. h. der Anpassung der DFT an die Kette $\mathcal{T}$, einen Lösungsansatz für beide algorithmischen Probleme. Wir besprechen dieses wichtige Konzept ausführlich in den nachfolgenden Kapiteln und geben zunächst einen kurzen Überblick über die relevante Literatur.

Verallgemeinerte FFTs (siehe Problem (2)) wurden 1987 für auflösbare Gruppen von Beth [8], 1989 für allgemeine endliche und symmetrische Gruppen von Clausen [14] und 1991 für überauflösbare Gruppen von Baum [2] entwickelt. Einige neuere Resultate und weitere Hinweise auf die Literatur findet man in [42]. Das Konzept der Symmetrieanpassung hat ihren Ursprung in Youngs „seminormal form“ der irreduziblen Darstellungen der symmetrischen Gruppen (siehe [40], S. 124 ff., und [12], S. 343).

Für Problem (1) haben Baum und Clausen (1994) in [4] eine im wesentlichen optimale Lösung für den Fall der endlichen überauflösbaren Gruppen angegeben. Püschel (1998) beschreibt in [47] einen Algorithmus, der die reguläre Darstellung einer auflösbaren Gruppe G zerlegt, was gleichbedeutend mit der Konstruktion einer symmetrieangepaßten DFT für D ist.

Die Anwendungen verallgemeinerter DFTs sind noch vergleichsweise dünn gesät und liegen hauptsächlich im Bereich der Signalverarbeitung und Statistik (siehe u. a. [15, 24, 25]). Einen Überblick hierfür liefert auch der Artikel [51] von Rockmore (1995). In dem im letzten Jahr erschienenen Artikel [29] wird die Anwendung verallgemeinerter DFTs nicht-abelscher Gruppen (Kranzprodukte zyklischer Gruppen) im Bereich der Bildverarbeitung beschrieben, die viele bekannte Transformationen wie die Haar-Wavelet-Transformation und verschiedene Mehrkanal-Filterbänke verallgemeinern.

Beitrag und Gliederung dieser Arbeit

Sei G eine endliche überauflösbare Gruppe der Ordnung $N = |G|$, die in pc-präsentierter Form vorliegt. Für solche Gruppen haben Baum und Clausen Algorithmen angegeben, die sowohl die DFT-Generierung (Problem (1)) als auch die DFT-Auswertung (Problem (2)) in einer im wesentlichen optimalen Laufzeit realisieren. In der vorliegenden Dissertation geht es um die Fortsetzung dieser Arbeiten, und zwar sowohl in theoretischer als auch in praktischer Sicht. Darüber hinaus werden neuartige, auf diesen verallgemeinerten DFTs basierende Anwendungen im Bereich der Computeralgebra und im Bereich der Signalverarbeitung beschrieben. Zum einen wird ein Algorithmus zur DFT-basierten Wortnormalisierung in pc-präsentierten Gruppen entwickelt, der im Vergleich zu herkömmlichen „Collection“-Verfahren von anderen Gruppeninvarianten abhängt und ein in vielen Fällen besseres Laufzeitverhalten aufweist. Zum anderen wird durch die erstmalige Implementierung der schnellen Algorithmen zur DFT-Generierung und DFT-Auswertung der Signalverarbeitung eine bisher nicht bekannte Familie von Transformationen zugänglich gemacht, und es werden mittels der verallgemeinerten Spektraltransformationen erste Experimente zur randomisierten Datenkompression durchgeführt.

Im folgenden gehen wir auf die Gliederung dieser Arbeit ein und fassen die Inhalte der einzelnen Kapitel zusammen. Jedes Kapitel beginnt mit einer kurzen Übersicht, und der jeweilige Stand der Forschung wird erläutert. In den Schlußabschnitten werden dann an gegebener Stelle weiterführende Fragestellungen und offene Probleme formuliert und diskutiert.

Kapitel 2 stellt die in dieser Arbeit benötigten algebraischen Grundlagen dar und legt damit auch die Bezeichnungskonventionen fest. Unter anderem werden auch die Grundideen der auf dem Satz von Clifford beruhenden DFT-Generierung für endliche auflösbare Gruppen dargestellt.

Kapitel 3 behandelt den von Baum und Clausen in [4] vorgestellten Algorithmus (BC-Algorithmus), der im überauflösbaren Fall eine an eine Kompositionreihe angepaßte DFT mit $O(|G| \log^2 |G|)$ arithmetischen Operationen generiert. Darüber hinaus können alle Berechnungen rein symbolisch in $\mathbb{Z}_e$, wobei e den Exponenten von G bezeichnet, durchgeführt werden. Bei den Matrixeinträgen der Darstellungen handelt es sich um e-te Einheitswurzeln, und alle durchzuführenden Matrixmanipulationen sind entweder Multiplikationen oder Inversionen. Dieser Algorithmus wurde im Rahmen der vorliegenden Dissertation erstmals implementiert. Hierfür wurde eine auf dem Charaktergraphen basierende DFT-Datenstruktur entwickelt, die zur Speicherung der DFT-Daten nur $O(|G|)$ ganzen Zahlen benötigt, die durch den Exponenten der Gruppe beschränkt sind. Der BC-Algorithmus wurde dahingehend modifiziert, daß auch während der Laufzeit die obere Schranke $O(|G|)$ für den benötigten Speicherplatz nicht überschritten wird. Es wurden zahlreiche Tests zur Bestimmung der tatsächlichen Laufzeiten und des Speicherplatzbedarfs durchgeführt. Zu diesem Zweck wurde mit Hilfe des Computeralgebrasystems GAP eine Bibliothek von pc-Präsentationen erzeugt. Eine ausführliche Diskussion der Ergebnisse und einige Beispiele runden das Kapitel ab.

Kapitel 4 befaßt sich mit einer ersten DFT-basierten Anwendung im Bereich der algorithmischen Gruppentheorie. Hierbei geht es um einen im Rahmen dieser Arbeit entwickelten Algorithmus (WN-Algorithmus) zur effizienten Normalisierung von Wörtern in endlichen pc-präsentierten Gruppen. Eine Implementierung des WN-Algorithmus zeigt die Praktikabilität dieses Algorithmus. Als Alternative zu herkömmlichen „Collection"-Verfahren [55] werden hierbei Teile der zuvor berechneten DFT benutzt, um das Normalisierungsproblem in den Spektralbereich zu übertragen, wo es effizient gelöst werden kann. Eine Analyse des Algorithmus ergibt als Spezialfall (siehe Korollar 4.1.3), daß in pc-präsentierten p-Gruppen der Ordnung p^n mit Exponent e die Normalisierung des Produkts von zwei in Normalform gegebenen Wörtern mit maximal $5 \cdot p \cdot n^2$ Additionen in $\mathbb{Z}_e$ durchgeführt werden kann. Anders als bei herkömmlichen Verfahren hängt das Laufzeitverhalten nicht wesentlich von der Komplexität der pc-Präsentation, sondern vor allem von der Länge n der Kompositionsreihe ab. Die theoretischen Ergebnisse werden durch praktische Tests zur Analyse des Laufzeitverhaltens untermauert. In den anschließenden Abschnitten wird besprochen, wie die Wortnormalisierung zur multivariaten Polynomdivision in kommutativen wie auch nicht-kommutativen Polynomringen bezüglich spezieller Gröbnerbasen verwendet werden kann.

Kapitel 5 kommt zur Auswertung der in Kapitel 3 generierten DFT auf einem komplexwertigen Signal $f \in \mathbb{C}G$. Auch dieses Berechnungsproblem läßt sich im überauflösbaren Fall, wie Baum in [2] gezeigt hat, im wesentlichen optimal lösen. Sein rekursiver Algorithmus benötigt $O(|G| \log |G|)$ komplexe Operationen. Im Hinblick auf eine insbesondere auch bezüglich des Speicherplatzes effiziente Implementierung benötigen wir eine iterative Variante des Baum-Algorithmus. Es stellt sich heraus, daß diese Variante einer Faktorisierung der DFT-Matrix in n dünnbesetzte Matrizen entspricht, wobei n die Länge der Kompositionsreihe von G bezeichnet und die Anzahl der von Null verschiedenen Einträge dieser Matrizen von der Primfaktorzerlegung von G abhängt. Auch hier wurden zahlreiche Tests zum Laufzeitverhalten und zur Fehleranalyse, die beim Rechnen mit Gleitpunktzahlen entstehen, durchgeführt.

Kapitel 6 behandelt die Zerlegung der Gruppenalgebra $\mathbb{C}G$ in G-invariante Unterräume, was signaltheoretisch als Zerlegung eines Signals in seine verallgemeinerten Spektralkomponenten gedeutet werden kann. Mit Hilfe von Idempotenten lassen sich beliebige Ringe mit 1 in eine direkte Summe von Linksidealen zerlegen. Je mehr Struktur diese Ringe aufweisen, desto stärkere Aussagen lassen sich bezüglich der durch die Idempotenten definierten Summenzerlegungen treffen. In diesem Kapitel untersuchen wir durch sukzessive Hinzunahme von Struktur den Fall von Algebren, halbeinfachen Algebren, Gruppenalgebren, bis hin zu Gruppenalgebren endlicher überauflösbarer Gruppen. Berücksichtigt man schließlich noch die Hilbertraumstruktur von $\mathbb{C}G$ bez. des Standardskalarprodukts, so landet man schließlich bei der verallgemeinerten Spektraldarstellung einer Funktion $f \in \mathbb{C}G$, die sich mit einem schnellen FFT-Algorithmus berechnen läßt.

Kapitel 7 geht auf mögliche Anwendungen der verallgemeinerten Spektraltransformationen im Bereich der digitalen Signalverarbeitung zur Signalanalyse und Datenkompression ein. Es beschreibt, wie sich die Menge der verallgemeinerten Spektralkoeffizienten mit Hilfe der diskreten Multiskalenanalyse in Teilmengen zerlegen läßt, so daß die Koeffizienten jeder Teilmenge grob einem Frequenzband zugeordnet werden können. Eine Multiskalenanalyse entspricht dabei einer Folge von Partitionen der Eins der Gruppenalgebra $\mathbb{C}G$. In [15], Kapitel 11, deuten Clausen und Baum einen Algorithmus zur effizienten Datenkompression mittels verallmeinerter DFTs an. Hier wird nun ein Gesamtsystem vorgestellt, das diese Ideen erstmalig realisiert, und es werden einige der Experimente, die im Bereich der randomisierten Datenkompression durchgeführt wurden, beschrieben.

Das abschließende **Kapitel 8** wendet sich noch einmal der DFT-Generierung zu. Setzt man die endliche Gruppe G nicht mehr als *überauflösbar*, sondern nur noch als *auflösbar* voraus, wird die DFT-Generierung wesentlich aufwendiger. Da in diesem Fall die Untergruppen einer Kompositionsreihe von G im allgemeinen nicht normal in G sind, lassen sich die Verkettungsräume nicht mehr wie im überauflösbaren Fall konstruieren. Darüber hinaus sind auflösbare Gruppen im allgemeinen keine M-Gruppen, d. h. es existieren in diesen Fällen keine monomialen Darstellungen. Ähnlich effiziente Algorithmen wie der BC-Algorithmus sind daher nicht zu erwarten. Wir beschreiben einen Algorithmus (M-Algorithmus), der die DFT-Generierung für auflösbare Gruppen mit $O(p \cdot |G|^2 \log(|G|))$ Körperoperationen realisiert, wobei p den größten Primteiler von $|G|$ bezeichnet. Hierbei muß die auflösbare Gruppe in Form einer pc-Präsentation vorliegen, die mit einer Kompositionsreihe korrespondiert, die wiederum eine Hauptreihe verfeinert. Unseres Wissens ist dies die erste „worst-case“-obere Komplexitätsschranke für den Fall auflösbarer Gruppen.

Danksagung

Diese Arbeit entstand in der Abteilung V des Instituts für Informatik an der Rheinischen Friedrich-Wilhelms-Universität. An dieser Stelle bedanke ich mich herzlich bei den zahlreichen am Gelingen dieser Arbeit direkt oder indirekt beteiligten Personen.

Hierzu zählen in erster Linie meine Eltern und Freunde, bei denen ich immer ein offenes Ohr und die notwendige menschliche Unterstützung fand.

Ein Stipendium der Studienstiftung des deutschen Volkes ermöglichte es mir, mich voll auf

meine Studien zu konzentrieren und mich, über den Tellerrand hinwegblickend, in viele für mich neue und spannende Themen einzuarbeiten. Stellvertretend möchte ich mich dafür bei meinem Ansprechpartner Herrn Dr. N. Weidtmann bedanken.

Viele Anregungen bekam ich durch Diskussionen, die ich unter anderem mit Prof. Dr. C.-F. Bödigheimer, Dietmar König, Thomas Rieband und Prof. Dr. B. Sturmfels (speziell über die Gröbnerbasen) führen durfte.

Außerdem danke ich allen Mitarbeitern unserer Arbeitsgruppe für die offene und freundschaftliche Atmosphäre, insbesondere Vlora Arifi, Roland Engelbrecht, Heiko Goeman, Frank Kurth und Dirk Meyer.

Bei Herrn Priv.-Doz. Dr. M.A. Shokrollahi bedanke ich mich für die Übernahme des Korreferats.

Zu besonders großem Dank bin ich meinem Doktorvater Prof. Dr. M. Clausen verpflichtet, bei dem ich immer ein offenes Ohr für meine Forschungsprobleme fand. Ohne seine Diskussionsbereitschaft, seine konstruktiven Ideen und seine Anregungen wäre diese Arbeit nicht entstanden. Ich habe die drei Jahre in seiner Arbeitsgruppe sehr genossen und bin von ihm in dieser Zeit in vielerlei Hinsicht geprägt worden.

Meinard Müller, März 2001.

Kapitel 2

Grundlagen

In diesem Kapitel fassen wir die für alle Teile dieser Arbeit relevanten mathematischen Grundlagen zusammen und führen damit gleichzeitig die benötigten Symbole und Bezeichnungen ein, die einheitlich in der ganzen Arbeit verwendet werden. In Abschnitt 2.1 erinnern wir an die Grundlagen der (gewöhnlichen) Darstellungstheorie endlicher Gruppen. In Abschnitt 2.2 wird die diskrete Fouriertransformation (DFT) auf beliebige endliche Gruppen verallgemeinert. Für das Rechnen in endlichen auflösbaren Gruppen haben sich insbesondere die pc-Präsentationen, die wir in Abschnitt 2.3 beschreiben, als eine sehr geeignete und effiziente Präsentationsform erwiesen. Die Grundideen zur DFT-Generierung aus den als pc-Präsentation vorliegenden Gruppen fassen wir in Abschnitt 2.4 zusammen und beschließen dieses Kapitel mit einigen Bemerkungen über dünn besetzte Matrizen und Matrixdarstellungen in Abschnitt 2.5.

2.1 Grundlagen zur Darstellungstheorie

In diesem Abschnitt wird kurz an einige Grundlagen der Darstellungstheorie erinnert und die dafür benötigte Notation eingeführt. Für Einzelheiten verweisen wir auf die Standardliteratur, z. B. [52].

Sei G eine endliche Gruppe. Eine (gewöhnliche) *Darstellung* von G vom *Grad* (oder der *Dimension*) d ist ein Gruppenhomomorphismus $D : G \to \mathrm{GL}(d, \mathbb{C})$. Der zugehörige Charakter $\chi : G \to \mathbb{C}$ ist durch $\chi(g) := \mathrm{Spur}(D(g))$ definiert. Zwei Darstellungen D und D' heißen *äquivalent*, $D \sim D'$, falls für eine invertierbare Matrix X und alle $g \in G$ die Gleichung $D'(g) = D^X(g)$ gilt, wobei D^X durch $D^X(g) := XD(g)X^{-1}$, $g \in G$, definiert ist. Zwei Darstellungen sind äquivalent genau dann, wenn ihre Charaktere übereinstimmen.

Die *direkte Summe* $D \oplus D'$ zweier Darstellungen D und D' von G ist für $g \in G$ definiert durch $(D \oplus D')(g) := D(g) \oplus D'(g)$. Mit mD, $m \in \mathbb{N}$, bezeichnen wir die m-fache direkte Summe von D. Eine Darstellung D heißt *irreduzibel*, falls D nicht äquivalent ist zu einer direkten Summe zweier Darstellungen. Charaktere, die zu irreduziblen Darstellungen gehören, heißen *irreduzible Charaktere*. Die Anzahl irreduzibler Charaktere, d. h. die Anzahl von Äquivalenzklassen irreduzibler Darstellungen von G, ist gleich der Anzahl der Konjugationsklassen von G.

Nach dem Satz von Maschke ist jede Darstellung D äquivalent zu einer direkten Summe irreduzibler Darstellungen: $D \sim D_1 \oplus \ldots \oplus D_r$. Sei Δ eine irreduzible Darstellung von G mit Charakter δ, dann hängt die *Multiplizität* $\langle \Delta | D \rangle := |\{i : D_i \sim \Delta\}|$ von Δ in D nur von den Charakteren ab. Bezeichne χ den Charakter von D, dann gilt genauer

$$\langle \Delta | D \rangle = \langle \delta | \chi \rangle := |G|^{-1} \sum_{g \in G} \delta(g^{-1}) \chi(g). \tag{2.1}$$

Verkettungsräume (intertwining spaces) sind ein weiteres Hilfsmittel zur Untersuchung von Multiplizitäten oder allgemeiner von direkten Summenzerlegungen von Darstellungen. Der *Verkettungsraum* zweier Darstellungen D und D' von G ist definiert durch

$$\mathrm{Int}(D, D') := \{X \in \mathbb{C}^{d' \times d} | X D(g) = D'(g) X \quad \text{für alle} \quad g \in G\}, \tag{2.2}$$

wobei d bzw. d' den jeweiligen Grad der Darstellungen bezeichnen. Nach dem Lemma von Schur ist $\mathrm{Int}(D, D)$ eindimensional genau dann, wenn D irreduzibel ist. In diesem Fall besteht der Verkettungsraum aus allen skalaren Vielfachen der Einheitsmatrix Id_d. Die folgenden Aussagen ergeben sich unmittelbar aus dem Lemma von Schur.

Lemma 2.1.1 *Seien $D_1, \ldots, D_h$ paarweise inäquivalente Darstellungen von G und seien $D_i \sim F_i$. Dann gilt für beliebige nicht-negative ganze Zahlen n_i, m_i*

$$\mathrm{Int}\left(\oplus_{i=1}^h m_i D_i, \oplus_{i=1}^h n_i F_i\right) = \bigoplus_{i=1}^h \mathrm{Int}\left(m_i D_i, n_i F_i\right) = \bigoplus_{i=1}^h \mathbb{C}^{n_i \times m_i} \otimes \mathrm{Int}\left(D_i, F_i\right).$$

Darüber hinaus gilt für Darstellungen D, F von G und invertierbare Matrizen X, Y geeigneten Formats

$$\mathrm{Int}\left(F^X, D^Y\right) = Y \,\mathrm{Int}\left(F, D\right) X^{-1}.$$

Sei H Untergruppe von G, D eine Darstellung von G und F eine Darstellung von H. Falls die *Einschränkung* $D{\downarrow}H$ von D auf H gleich der Darstellung F ist, dann heißt D *Erweiterung* von F. Ausgehend von F vom Grad f und einer vollständigen Liste $T = (g_1, \ldots, g_t)$ von Repräsentanten der Linksnebenklassen von H in G erhält man eine Darstellung G vom Grad $f \cdot t$, die sogenannte *induzierte Darstellung* $F{\uparrow}_T G$, auf die folgende Weise:

$$(F{\uparrow}_T G)(g) := \left(\dot{F}(g_i^{-1} g g_j)\right)_{1 \le i,j \le t}.$$

Hierbei ist $\dot{F}(x)$ definiert als $F(x)$ für $x \in H$ und als $f \times f$ Nullmatrix außerhalb von H. Nach dem Reziprozitätssatz von Frobenius sind Induktion und Einschränkung im folgenden Sinn dual zueinander: Ist D eine irreduzible Darstellung von G und F eine irreduzible Darstellung von H, dann ist die Multiplizität von F in $D{\downarrow}H$ gleich der Multiplizität von D in $F{\uparrow}_T G$. Wir bezeichnen diese gemeinsame Multiplizität mit $\langle D | F \rangle$. Ein entsprechendes Ergebnis gilt für die Charaktere.

Sei nun $\mathcal{C} = (G = G_n > G_{n-1} > \ldots > G_1 > G_0 = \{1\})$ eine Kette von Untergruppen von G. Zu dieser Kette assoziieren wir einen Graph, den $\mathcal{C}$-*Charaktergraph* of G. Die Menge der Knoten ist aufgeteilt auf $n+1$ Stufen. Die Knoten der Stufe i entsprechen den irreduziblen

Charakteren von G_i. Nur die Knoten aufeinanderfolgender Stufen sind durch gewichtete Kanten verbunden. Falls χ und ψ irreduzible Charaktere von G_i bzw. G_{i-1} sind, dann sind die zwei Knoten durch eine gewichtete Kante verbunden, falls das Gewicht $\langle\chi|\psi\rangle > 0$ ist. Dieser Graph wird als grundlegende Datenstruktur zur Konstruktion und Speicherung irreduzibler Darstellungen dienen. Ein Beispiel ist in Abbildung 2.1 gegeben.

Es gibt eine enge Verbindung zwischen den Darstellungen von G und einer normalen Untergruppe N. Grundlegend hierfür ist die Operation von G auf der Menge der irreduziblen Charaktere von N mittels $(g\cdot\psi)(n) := \psi^g(n) := \psi(g^{-1}ng)$. Für eine Darstellung F von N und jedes $g \in G$ definiert $F^g(n) := F(g^{-1}ng)$, $n \in N$, eine Darstellung von N, eine *g-Konjugierte* von F. Die folgende Version des Satzes von Clifford wird eine wichtige Rolle spielen.

Satz 2.1.2 *Sei $N \lhd G$ und χ ein irreduzibler Charakter von G. Sei ψ eine irreduzible Komponente von $\chi{\downarrow}N$ mit Multiplizität $m > 0$ und seien $\psi = \psi_1, \ldots, \psi_q$ die verschiedenen g-Konjugierten von ψ, $g \in G$. Dann gilt*

$$\chi{\downarrow}N = m\sum_{k=1}^{q}\psi_k.$$

Für einen Beweis des Satzes verweisen wir auf [36], Theorem (6.2). Ein analoges Ergebnis gilt für die entsprechenden Darstellungen.

2.2 Gruppenalgebren und DFT

In diesem Abschnitt wird der Begriff der diskreten Fouriertransformation (DFT) auf beliebige endliche Gruppen verallgemeinert. Wir folgen dabei [15], wo man auch die hier nicht aufgeführten Beweise der Aussagen findet.

Sei also G eine endliche Gruppe. Die Menge $\mathbb{C}G := \{a|a : G \to \mathbb{C}\}$ aller $\mathbb{C}$-wertigen Funktionen (Signale) auf G wird durch punktweise Addition und Skalarmultiplikation zu einem Vektorraum. Eine natürliche Basis ist durch die Indikatorfunktionen $(G \ni h \mapsto \delta_{gh})$ der Gruppenelemente $g \in G$ gegeben. Identifiziert man jedes Gruppenelement mit seiner Indikatorfunktion, dann kann $\mathbb{C}G$ als $\mathbb{C}$-lineare Hülle von G aufgefaßt werden. Weiterhin definiert

$$\langle a|b\rangle := \frac{1}{|G|}\sum_{g\in G} a_g\overline{b_g} \tag{2.3}$$

für $a = (a_g)_{g\in G}, b = (b_g)_{g\in G} \in \mathbb{C}G$ ein Skalarprodukt auf $\mathbb{C}G$ und macht diesen zum Hilbertraum. Die Gruppenmultiplikation in G erweitert sich zur sogenannten *Faltung* in $\mathbb{C}G$:

$$\Big(\sum_{g\in G} a_g g\Big)\cdot\Big(\sum_{h\in G} b_h h\Big) = \sum_{k\in G}\Big(\sum_{g\in G} a_g b_{g^{-1}k}\Big)k.$$

Auf diese Weise wird $\mathbb{C}G$ zu einer $\mathbb{C}$-Algebra, der sogenannten *Gruppenalgebra* von G über $\mathbb{C}$. Darstellungen von G erweitern sich auf kanonische Weise zu Algebrenhomomorphismen von $\mathbb{C}G$ und werden ebenfalls als Darstellung bezeichnet. Da sich Darstellungen von G und $\mathbb{C}G$ eineindeutig entsprechen, werden diese im folgenden identifiziert.

Ist zum Beipiel $G = C_N = \langle X \mid X^N = 1 \rangle$ die zyklische Gruppe der Ordnung N, dann kann $\mathbb{C}G$ mit dem Polynomring $\mathbb{C}[X]$ modulo dem von $X^N - 1$ erzeugten Ideal identifiziert werden. In diesem Fall ist die Faltung in $\mathbb{C}G$ nichts anderes als gewöhnliche Polynommultiplikation modulo der Relation $X^N = 1$. Sei ω eine primitive N-te Einheitswurzel, dann folgt aus der Faktorisierung $X^n - 1 = \prod_{j=0}^{n-1}(X - \omega^j)$ zusammen mit dem Chinesischen Restesatz, daß die kanonische Abbildung

$$\mathbb{C}C_N = \mathbb{C}[X]/(X^N - 1) \longrightarrow \oplus_{j=0}^{N-1}\mathbb{C}[X]/(X - \omega^j) = \oplus_{j=0}^{N-1}\mathbb{C}^{1\times 1}$$

ein Algebrenisomorphismus von der Gruppenalgebra in die Algebra der $N \times N$-Diagonalmatrizen definiert. Bezüglich der kanonischen Basen in beiden Räumen wird dieser Isomorphismus durch die klassische DFT-Matrix $\mathbf{D} = (\omega^{jk})_{0 \le j,k < N}$ beschrieben.

Wedderburns Satz für halbeinfache Algebren über Zerfällungskörpern verallgemeinert diese Situation. Im Spezialfall von komplexen Gruppenalgebren gilt die folgende Version.

Satz 2.2.1 (Wedderburn) *Die Gruppenalgebra $\mathbb{C}G$ einer endlichen Gruppe G ist isomorph zu einer Algebra von Blockdiagonalmatrizen:*

$$D = \oplus_{k=1}^{h} D_k : \mathbb{C}G \xrightarrow{\simeq} \oplus_{k=1}^{h}\mathbb{C}^{d_k \times d_k}.$$

Die Anzahl h der Blöcke ist gleich der Anzahl der Konjugationsklassen von G und die Projektionen $D_1, \ldots, D_h$ bilden eine vollständige Menge von paarweise inäquivalenten irreduziblen Darstellungen von $\mathbb{C}G$.

Man nennt eine solche Liste $D_1, \ldots, D_h$ auch *Transversale* von irreduziblen Darstellungen, die wir im folgenden mit $\mathrm{Irr}(G)$ bezeichnen.

Der Begriff der klassischen diskreten Fouriertransformation für zyklische Gruppen wird nun in folgender Weise auf beliebige endliche Gruppen G verallgemeinert.

Definition 2.2.2 *Jeder Isomorphismus $D : \mathbb{C}G \longrightarrow \oplus_{k=1}^{h}\mathbb{C}^{d_k \times d_k}$ von $\mathbb{C}$-Algebren heißt* diskrete Fouriertransformation (DFT) *für $\mathbb{C}G$ (oder einfach für G).*

Bezüglich der kanonischen Basen in $\mathbb{C}G$ und $\oplus_{k=1}^{h}\mathbb{C}^{d_k \times d_k}$ ist jede DFT durch eine $|G| \times |G|$-Matrix $\mathbf{D}$ gegeben. Dabei korrespondieren die Spalten von $\mathbf{D}$ zu den Elementen von G. Die Reihen werden durch

$$\bigcup_{1 \le k \le h} \{(k, \mu, \nu) | 1 \le \mu, \nu \le d_k\} \tag{2.4}$$

parametrisiert, wobei (k, μ, ν) die Position (μ, ν) in D_k beschreibt. Mit anderen Worten gilt

$$\mathbf{D}_{(k,\mu,\nu),g} := D_k(g)_{\mu,\nu}. \tag{2.5}$$

Wir verweisen auf Abschnitt 5.4 für ein konkretes Beispiel einer DFT-Matrix $\mathbf{D}$. Wedderburns Satz garantiert für jede endliche Gruppe G die Existenz einer DFT.

2.3 PC-Präsentationen

In der algorithmischen Gruppentheorie sind die relevanten Gruppen oft durch eine endliche Präsentation mit Erzeugern und Relationen gegeben. Solche Präsentationen dienen auch als Eingabe für unsere Algorithmen und werden im folgenden kurz beschrieben.

Eine endliche Gruppe G heißt *auflösbar* , wenn eine Kompositionsreihe

$$\mathcal{T} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_1 \rhd G_0 = \{1\})$$

existiert, deren Kompositionsfaktoren G_i/G_{i-1} von Primzahlordnung p_i sind. Für $1 \leq i \leq n$ sei g_i ein Element in G_i, welches nicht in G_{i-1} ist. In Bezug auf $(g_1, \ldots, g_n)$ kann jedes Element $g \in G$ eindeutig ausgedrückt werden in *Normalform*

$$g = g_n^{e_n} \cdot g_{n-1}^{e_{n-1}} \cdot \ldots \cdot g_1^{e_1} \quad (0 \leq e_i < p_i). \tag{2.6}$$

Die Multiplikation in G ist dann vollständig beschrieben, wenn die Normalformen von allen Potenzen $g_i^{p_i}$ und allen Kommutatoren $[g_i, g_j] := g_i^{-1} g_j^{-1} g_i g_j$ bekannt sind. Jede auflösbare Gruppe besitzt demnach eine *pc-Präsentation* (power-commutator presentation), die aus Relationen für die Potenzen und Kommutatoren besteht und von der folgenden Form ist:

$$G = \langle g_1, \ldots, g_n \mid g_i^{p_i} = u_i \ (1 \leq i \leq n), [g_i, g_j] = w_{ij} \ (1 \leq i < j \leq n) \rangle, \tag{2.7}$$

mit Worten $u_i \in G_{i-1}$ und $w_{ij} \in G_{j-1}$ gegeben in Normalform

$$u_i = g_{i-1}^{a_{i,i-1}} \cdot \ldots \cdot g_1^{a_{i,1}} \qquad \text{bzw.} \qquad w_{ij} = g_{j-1}^{b_{ij,j-1}} \cdot \ldots \cdot g_1^{b_{ij,1}}. \tag{2.8}$$

Darüber hinaus verlangen wir, daß die Präsentationen *konsistent* sind, d. h. jedes Wort in den Erzeugern hat eine eindeutig bestimmte Normalform. Konsistente pc-Präsentationen dieser Art beschreiben exakt die Klasse der auflösbaren Gruppen.

Eine auflösbare endliche Gruppe G heißt *überauflösbar*, wenn es eine Kompositionsreihe gibt, bei der jede Gruppe G_i nicht nur normal in G_{i+1} sondern sogar normal in der ganzen Gruppe G ist. Derartige Kompositionsreihen sind *Hauptreihen*. In diesem Fall gilt für eine zugehörige pc-Präsentation $w_{ij} \in G_i$, also

$$w_{ij} = g_i^{b_{ij,i}} \cdot \ldots \cdot g_1^{b_{ij,1}}.$$

Als Beispiel geben wir eine konsistente pc-Präsentation einer überauflösbaren Gruppe mit 128 Elementen an, die wir auch mit G_{128} bezeichnen. In der Präsentation sind, wie dies im folgenden aus Gründen der Übersichtlichkeit meist geschehen wird, triviale Kommutator-Relationen nicht aufgeführt.

$$\begin{aligned} G_{128} \quad = \quad & \langle g_7, g_6, g_5, g_4, g_3, g_2, g_1 \mid g_1^2 = g_2^2 = g_4^2 = g_5^2 = g_6^2 = 1, g_3^2 = g_1, g_7^2 = g_4, \\ & [g_2, g_6] = [g_2, g_7] = [g_3, g_4] = [g_3, g_5] = [g_3, g_6] = g_1, [g_3, g_7] = g_2, \\ & [g_4, g_5] = g_2 \cdot g_1, [g_4, g_6] = g_3 \cdot g_1, [g_5, g_7] = g_3, [g_6, g_7] = g_5 \rangle \end{aligned} \tag{2.9}$$

Ist eine pc-Präsentation der Form (2.7) gegeben, so definiert diese kanonisch eine Kompositionsreihe $\mathcal{T}$. Die Abbildung 2.1 zeigt den $\mathcal{T}$-Charaktergraph der pc-präsentierten Gruppe G_{128}. Jeder Knoten entspricht einem irreduziblen Charakter und der entsprechenden Äquivalenzklasse von irreduziblen Darstellungen. Die Kantengewichte haben alle den Wert 1, d. h.

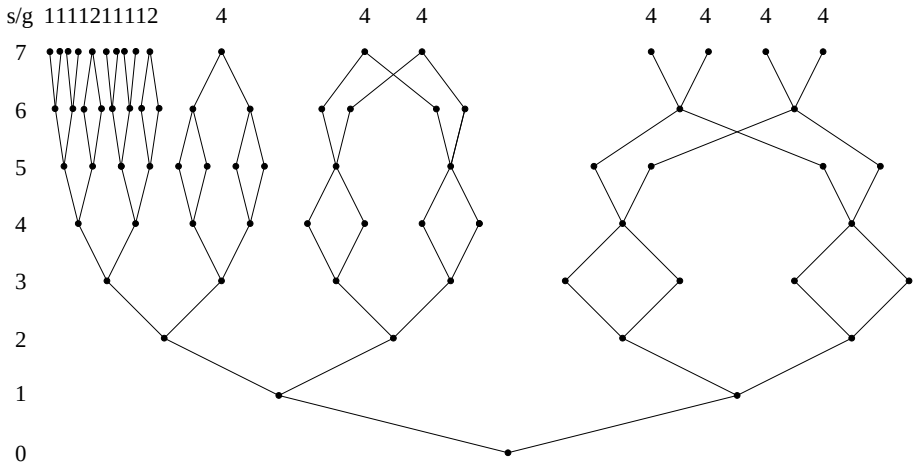


Abbildung 2.1: Charaktergraph von G_{128}.

alle nicht-trivialen Multiplizitäten sind 1. Die Zahlen der linken Spalte beschreiben die Stufe und die der obersten Zeile den Grad der entsprechenden Darstellung der obersten Stufe.

Es ist kein Zufall, daß alle Kantengewichte den Wert 1 haben. Dies gilt allgemein für $\mathcal{T}$-Charaktergraphen auflösbarer Gruppen mit Kompositionsreihe $\mathcal{T}$ und folgt unmittelbar aus der Version des Satzes von Clifford in Abschnitt 2.4.

Bemerkung 2.3.1 In der Notation haben wir uns eine kleine Nachlässigkeit erlaubt, da nicht zwischen Wörtern und Elementen unterschieden wurde. Ein *Wort* über dem Alphabet $A := \{g_1, \ldots, g_n\}$ ist eine endliche Folge von Elementen in A. Die Menge aller Wörter über diesem Alphabet wird auch mit A^* bezeichnet. Ein *Element* einer pc-präsentierten Gruppe G ist eine Äquivalenzklasse von Wörtern modulo des durch die pc-Relationen erzeugten Normalteilers. Um die Notation nicht unnötig zu verkomplizieren, schreiben wir für ein Gruppenelement ein dieses Gruppenelement repräsentierendes Wort, wie z. B. geschehen in (2.8). Wenn wir also im folgenden von einem *Wort* $g_n^{\alpha_n} \cdot \ldots \cdot g_1^{\alpha_1}$ sprechen, meinen wir ein Element in $\{g_1, \ldots, g_n\}^*$. Sprechen wir von einem *Element* $g_n^{\alpha_n} \cdot \ldots \cdot g_1^{\alpha_1}$, dann meinen wir das durch dieses Wort repräsentierte Gruppenelement in G.

Bemerkung 2.3.2 Im allgemeinen muß man bei Präsentationen zwischen sogenannten Monoid- und Gruppenpräsentationen unterscheiden. Bei Gruppenpräsentationen werden neben den angegebenen Erzeugern g_i deren formale Inverse g_i^{-1} als zusätzliche Erzeuger und die zwischen Erzeuger und inversem Erzeuger geltenden Relationen $g_i g_i^{-1} = 1$ und $g_i^{-1} g_i = 1$ zu den angegebenen Relationen hinzugenommen. Damit definieren Gruppenpräsentationen immer Gruppen, während Monoidpräsentationen im allgemeinen nur Monoide definieren. Präsentationen der Form (2.7) definieren allerdings schon als Monoidpräsentation eine Gruppe. Dies folgt daraus, daß aus den angegebenen Relationen für jeden Erzeuger g_i eine Relation der Form $g_i^N = 1$ für ein $N \in \mathbb{N}$ ableitbar ist, woraus die Invertierbarkeit von g_i folgt. In diesem Fall führen Monoid- und Gruppenpräsentation zu isomorphen Gruppen und zwischen diesen beiden Präsentationsformen muß nicht unterschieden werden. Für weitere Einzelheiten zu dieser Problematik verweisen wir auf [55].

2.4 Grundidee zur DFT-Generierung

In diesem Abschnitt fassen wir die allgemeinen Ideen für einen Algorithmus zusammen, der zu einer endlichen pc-präsentierten auflösbaren Gruppe eine DFT konstruiert, die bezüglich einer Kompositionsreihe von G angepaßt ist. In Bezug auf eine solche pc-Präsentation ist eine d-dimensionale Darstellung D von G vollständig durch die darstellenden Matrizen $D(g_1), \ldots, D(g_n)$ auf den Erzeugern beschrieben.

Das Konzept der *Symmetrieanpassung* einer Darstellung D ist entscheidend im Hinblick auf einen effizienten Algorithmus. D heißt $\mathcal{T}$*-angepaßt*, falls für alle $0 \leq i \leq n$ die folgenden Bedingungen gelten:

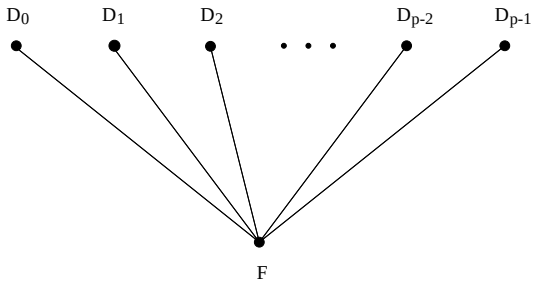
(1) Die Einschränkung $D{\downarrow}G_i$ ist *gleich* der direkten Summe irreduzibler Darstellungen von G_i, d. h. $D{\downarrow}G_i = \oplus_q F_{iq}$ mit irreduziblen Darstellungen F_{iq}.

(2) Äquivalente irreduzible Konstituenten von $D{\downarrow}G_i$ sind *gleich*, d. h. falls $F_{iq} \sim F_{it}$ dann $F_{iq} = F_{it}$ (aber nicht notwendigerweise $q = t$).

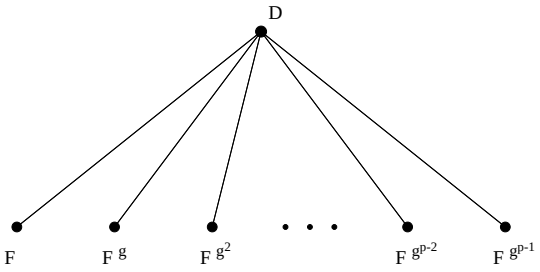
Ist D eine $\mathcal{T}$-angepaßte Darstellung, dann ist auch die Einschränkung $D{\downarrow}G_i$, $0 \leq i \leq n$, $\mathcal{T}_i$-angepaßt, wobei $\mathcal{T}_i$ die Kette $(G_i \supset \ldots \supset G_0)$ bezeichnet. Wir schreiben im folgenden auch $\mathrm{Irr}(G_i, \mathcal{T}_i)$ für eine Transversale von $\mathcal{T}_i$-angepaßten irreduziblen Darstellungen von G_i.

Die zentrale Idee des Algorithmus basiert auf dem Satz von Clifford. In unserem speziellen Fall besagt er, daß für eine irreduzible Darstellung F von G_{i-1}, $0 < i \leq n$, genau einer der beiden folgenden Fälle zutrifft.

Fall 1. Alle $F^{g_i^k}$, $0 \leq k < p_i =: p$, sind äquivalent. Dann läßt sich F zu p paarweise inäquivalenten irreduziblen Darstellungen $D_0, \ldots, D_{p-1}$ von G_i gleichen Grades erweitern. Bezeichnen $\chi^0, \ldots, \chi^{p-1}$ die linearen Charaktere der zyklischen Gruppe G_i/G_{i-1} in geeigneter Reihenfolge, dann gilt darüber hinaus $D_k = \chi^k \otimes D_0$ für alle k und $F{\uparrow}G_i \sim D_0 \oplus \ldots \oplus D_{p-1}$.



Fall 2. Alle $F^{g_i^k}$, $0 \leq k < p_i =: p$, sind paarweise inäquivalent. Dann ist die Induktion $F{\uparrow}G_i$ eine irreduzible Darstellung von G_i vom Grad $p \cdot \deg(F)$. Darüber hinaus sind alle Darstellungen $F^{g_i^k}{\uparrow}G_i$, $0 \leq k < p$, äquivalent und $(F{\uparrow}G_i){\downarrow}G_{i-1} = \bigoplus_{k=0}^{p-1} F^{g_i^k}$.



Für einen Beweis verweisen wir auf Theorem (6.20) in [15]. Bis auf Äquivalenz können alle irreduziblen Darstellungen von G_i auf diese Weise erhalten werden. So können die irreduziblen Darstellungen von G iterativ von unten nach oben entlang der Hauptreihe $\mathcal{T}$ konstruiert werden. Für eine effiziente Konstruktion von $\mathrm{Irr}(G_i)$ aus $\mathrm{Irr}(G_{i-1})$ sollten dabei möglichst viele der Daten von Stufe $i-1$ verwendet werden. Man sollte daher $D \in \mathrm{Irr}(G_i)$ auf solche Weise definieren, daß $D{\downarrow}G_{i-1}$ nicht nur *äquivalent*, sondern sogar *gleich* der direkten Summe von irreduziblen Darstellungen aus $\mathrm{Irr}(G_{i-1})$ ist. Dies entspricht genau der Philosophie der Symmetrieanpassung. Folglich muß eine neue Darstellung $D \in \mathrm{Irr}(G_i, \mathcal{T}_i)$ in Schritt i nur auf dem Erzeuger g_i definiert werden; die Werte von D auf den Erzeugern $g_1, \ldots, g_{i-1}$ können von Schritt $i-1$ ohne zusätzlichen Aufwand übernommen werden.

Allerdings muß man für den Äquivalenztest und die Symmetrieanpassung für jedes $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ die Beziehung zwischen den konjugierten Darstellungen F^{g_i} und den entsprechenden $F' \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ mit $F^{g_i} \sim F'$ kennen. Das ist der Grund dafür, daß die Verkettungsräume $\mathrm{Int}(F^{g_i}, F')$ benötigt werden. Es stellt sich heraus, daß die Berechnung der Verkettungsräume der aufwendigste Teil der Konstruktion ist und den Berechnungsaufwand des Algorithmus bestimmt. Wir wollen zunächst annehmen, daß sowohl die Äquivalenz von Darstellungen entschieden als auch Verkettungsräume berechnet werden können. Dann kann der Algorithmus zur Berechnung von $\mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ folgendermaßen zusammengefaßt werden:

Eingabe: Eine konsistente pc-Präsentation einer endlichen auflösbaren Gruppe G, die eine Kompositionsreihe $\mathcal{T}$ definiert. $\mathrm{Irr}(G_0, \mathcal{T}_0)$ ist trivial.

Schritt i. $\mathrm{Irr}(G_i, \mathcal{T}_i)$ wird berechnet aus $\mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$, $1 \le i \le n$. Nach Clifford müssen für jedes $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ zwei Fälle berücksichtigt werden.

Fall 1. $F \sim F^{g_i}$. F hat p_i Erweiterungen $D_0, \ldots, D_{p_i-1}$.

- Sei ω primitive p_i-te Einheitswurzel und $X \in \mathrm{Int}(F^{g_i}, F) \setminus \{0\}$.
- Bestimme eine Lösung c_0 der Gleichung $c^{p_i} X^{p_i} = F(g_i^{p_i})$. ($g_i^{p_i}$ ist ein Wort in G_{i-1} gegeben durch die pc-Präsentation).
- Definiere $D_k(g_i) := c_0 \cdot \omega^k \cdot X$, $k = 0, \ldots, p_i - 1$.
- Definiere $D_k{\downarrow}G_{i-1} := F$ (Daten schon bekannt aus Schritt $i-1$) und erhalte eine $\mathcal{T}_i$-angepaßte Erweiterung D von F.

Fall 2. $F \not\sim F^{g_i}$. Dann ist $F \uparrow G_i$ irreduzibel und $(F \uparrow G_i) \downarrow G_{i-1} = \bigoplus_{k=0}^{p_i-1} F^{g_i^k}$. Nun muß $F \uparrow G_i$ adaptiert werden.

- Finde $F_k \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ mit $F_k \sim F^{g_i^k}$ für $k = 0, \ldots, p_i - 1$.
- Berechne $X_k \in \mathrm{Int}(F^{g_i^k}, F_k) \setminus \{0\}$ and setze $X := \bigoplus_{k=0}^{p_i-1} X_k$.

- Definiere $D(g_i) := X^{-1}(F \uparrow G_i)(g_i)X$.
- Definiere $D(g_j) := \bigoplus_{k=0}^{p_i-1} F_k(g_j)$ für $j = 0, ..., i-1$ (Daten schon bekannt aus Schritt $i-1$) und erhalte eine $\mathcal{T}_i$-angepaßte Darstellung D.

Ausgabe: Eine Transversale von irreduziblen $\mathcal{T}$-angepaßten Darstellungen $\mathrm{Irr}(G, \mathcal{T})$, wobei jedes $D \in \mathrm{Irr}(G, \mathcal{T})$ durch die Matrizen $D(g_1), \ldots, D(g_n)$ gegeben ist.

Weitere Einzelheiten und die Verifikation des Algorithmus findet man in [15].

2.5 Dünn besetzte Matrizen und Darstellungen

Wir benötigen im folgenden einige grundlegende Komplexitätsschranken hinsichtlich dünn besetzter Matrizen und Darstellungen. Eine Permutation $\sigma \in S_r := \mathrm{Sym}(\{1, \ldots, r\})$ werde dabei mit der Permutationsmatrix P_σ identifiziert, die nach Definition in der i-ten Spalte, $1 \leq i \leq r$, genau einen von Null verschiedenen Eintrag mit Wert 1 an der $\sigma(i)$-ten Position hat. Sei $\mathbb{K}$ ein beliebiger Körper und $d = f \cdot r$ mit $f, d, r \in \mathbb{N}$. Eine Matrix M heißt nach Definition *f-blockmonomial* genau dann, wenn

$$\exists \sigma \in S_r \, \exists A_1, \ldots, A_r \in \mathrm{GL}(f, \mathbb{K}) : M = (\sigma \otimes \mathrm{Id}_f) \cdot (A_1 \oplus \ldots \oplus A_r).$$

Eine Darstellung D von G heißt f-blockmonomial, falls $D(g)$ eine f-blockmonomiale Matrix für jedes $g \in G$ ist. In diesem Abschnitt sei eine Operation entweder eine Multiplikation, Addition, Subtraktion oder Inversion in $\mathbb{K}$ und alle Matrizen seien invertierbare $d \times d$-Matrizen über $\mathbb{K}$. Standardalgorithmen zur Multiplikation und Inversion von Matrizen benötigen $O(d^3)$ Operationen über $\mathbb{K}$ (asymptotisch effizientere Algorithmen zur Matrixmultiplikation wie z. B. Strassens Algorithmus [59] werden im folgenden nicht berücksichtigt). Falls f ein Teiler von d ist und alle Matrizen f-blockmonomial sind, können die Multiplikation und Inversion derartiger Matrizen offensichtlich mit

$$O\left(\frac{d}{f} f^3\right) = O\left(d \cdot f^2\right). \tag{2.10}$$

Operationen durchgeführt werden. Multiplikation einer f-blockmonomialen Matrix mit einer vollen Matrix benötigt

$$O\left(\frac{d^2}{f^2} f^3\right) = O\left(d^2 \cdot f\right) \tag{2.11}$$

Operationen. Der Fall $f = 1$ ist für uns von besonderem Interesse. Abkürzend bezeichnen wir eine 1-blockmonomiale Matrix auch als *monomiale Matrix* und analog eine 1-blockmonomiale Darstellung als *monomiale Darstellung*. Eine monomiale invertierbare Matrix $M \in \mathbb{K}^{d \times d}$ kann in der Form

$$M = \pi \cdot \mathrm{diag}(c_1, \ldots, c_d) \tag{2.12}$$

mit einer Permutation $\pi \in S_d$ und von Null verschiedenen Koeffizienten $c_1, \ldots, c_d$ geschrieben werden. Multiplikation und Inversion sind dann linear in der Dimension d. Unter dem m-ten Eintrag, $1 \leq m \leq d$, der monomialen Matrix M verstehen wir im folgenden den Koeffizienten c_m.

Beim Rechnen mit Darstellungen ergeben sich ähnliche Komplexitätsschranken. Seien im folgenden die Bezeichnungen wie in Abschnitt 2.3. In Bezug auf eine pc-Präsentation wird eine Darstellung D vom Grad d von G vollständig beschrieben durch die darstellenden Matrizen $D(g_1), \ldots, D(g_n)$ auf den Erzeugern. Für jedes $g \in G$ in Normalform kann mit Hilfe der binären Methode das Produkt $D(g) = D(g_n)^{e_n} \cdot \ldots \cdot D(g_1)^{e_1}$ mit

$$O(d^3 \log(|G|)) \tag{2.13}$$

arithmetischen Operationen berechnet werden. In dieser Arbeit bezeichne dabei log immer den Logarithmus zur Basis 2. Im Falle von f-blockmonomialen Darstellungen reduziert sich dieser Aufwand nach (2.10) zu

$$O(d \cdot f^2 \log(|G|)). \tag{2.14}$$

Abschließend stellen wir noch ein paar einfache Abschätzungen zusammen, die im folgenden hilfreich sein werden. Sei G eine endliche Gruppe, $\mathbb{C}G \simeq \oplus_{k=1}^{h} \mathbb{C}^{d_k \times d_k}$. Dann gilt

$$\sum_{k=1}^{h} d_k^2 = |G| \tag{2.15}$$

und

$$d := \max(d_1, \ldots, d_h) \leq |G|^{\frac{1}{2}}. \tag{2.16}$$

Für beliebige reelle Zahlen $r \geq 2$ gilt damit

$$d^r(G) := \sum_{k=1}^{h} d_k^r \leq d^{r-2} \sum_{k=1}^{h} d_k^2 \leq |G|^{\frac{r}{2}}. \tag{2.17}$$

Weiterhin sei $d^1(G) := \sum_{k=1}^{h} d_k$.

Kapitel 3

DFT-Generierung überauflösbarer Gruppen

Für endliche überauflösbare Gruppen G wurde von Baum und Clausen in [4] ein Algorithmus zur DFT-Generierung angegeben, der nur $O(|G| \log^2 |G|)$ rein symbolische Operationen über der additiven Gruppe $\mathbb{Z}_e := \mathbb{Z}/e\mathbb{Z}$ benötigt, wobei e den Exponenten der Gruppe bezeichnet. Im Rahmen der vorliegenden Dissertation konnte dieser Algorithmus erstmals in ein auf dem Computer lauffähiges Programm umgesetzt werden, so daß nun die irreduziblen Darstellungen in Form von Matrixdarstellungen konkret ausgerechnet werden können. In diesem Kapitel fassen wir in Abschnitt 3.1 die Hauptideen des Baum-Clausen-Algorithmus zusammen. Für die Implementierung wurde eine Datenstruktur entworfen, die für die DFT-Daten einen Speicherplatz benötigt, der maximal linear in der Gruppenordnung $|G|$ ist (siehe Abschnitt 3.2). Dabei wurde erreicht, daß auch während der Laufzeit des Algorithmus der benötigte Speicherplatz diese obere Schranke nicht übersteigt. Zur Illustration der Theorie sind in Abschnitt 3.3 einige Beispiele angegeben. Die Implementierung wurde hinsichtlich ihres Laufzeitverhaltens ausführlich analysiert und getestet. Teile der Ergebnisse werden in Abschnitt 3.4 zusammengefaßt und diskutiert. Wir schließen das Kapitel mit einem Ausblick auf weiterführende Fragestellungen in Abschnitt 3.5 ab.

3.1 Baum-Clausen-Algorithmus

In diesem Abschnitt fassen wir die wichtigsten Eigenschaften des Baum-Clausen-Algorithmus zusammen, der eine monomiale DFT einer überauflösbaren pc-präsentierten endlichen Gruppe G mit $O(|G| \log^2 |G|)$ Operationen konstruiert. Im folgenden bezeichnen wir diesen Algorithmus auch einfach als BC-Algorithmus. Für eine ausführliche Darstellung und eine Analyse des Algorithmus verweisen wir auf [4].

Die Effizienz des BC-Algorithmus beruht auf der Tatsache, daß jede endliche überauflösbare Gruppe G eine *monomiale* DFT besitzt. Es gibt also eine DFT $D = \oplus_{k=1}^{h} D_k$, so daß jede irreduzible Darstellung D_k monomial ist. Sei $\mathcal{T} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_0 = \{1\})$ eine Hauptreihe von G. Dann kann man zeigen, daß jede $\mathcal{T}$-adaptierte DFT automatisch monomial ist [2, 12, 16].

Es gilt sogar noch mehr: Es bezeichne e den Exponenten von G. Dann besitzt G eine *e-monomiale* DFT D, d. h. alle nicht-trivialen Einträge von $D(g)$, $g \in G$, sind e-te Einheitswurzeln. Obwohl die irreduziblen Darstellungen über dem Körper $\mathbb{C}$ berechnet werden, kommt man bei der Konstruktion einer e-monomialen DFT mit Ganzzahl-Arithmetik aus. Mit anderen Worten, es entstehen keine numerischen Probleme, da die Gleitkomma-Arithmetik nicht benötigt wird. Der tiefere Grund hierfür ist, daß der Algorithmus nur e-monomiale Matrizen verarbeitet und sich die Matrixmanipulationen auf Matrixmultiplikationen und Matrixinversionen beschränken. Daher kann in der additiven Gruppe $\mathbb{Z}_e$ gerechnet werden, welche isomorph zur Gruppe der e-ten Einheitswurzeln in $\mathbb{C}$ ist. (Man kann zeigen, daß der BC-Algorithmus über jedem Körper $\mathbb{K}$ arbeitet, der eine primitive e-te Einheitswurzel enthält. Wir betrachten aber im folgenden nur den Fall $\mathbb{K} = \mathbb{C}$. Siehe Abschnitt 3.5.1.)

Der BC-Algorithmus basiert auf dem Theorem von Clifford und verfährt nach dem in Abschnitt 2.4 beschriebenen iterativen Verfahren, das wir im folgenden auch als Hauptroutine bezeichnen. In Schritt i wird dabei $\mathrm{Irr}(G_i, \mathcal{T}_i)$ und der $\mathcal{T}_i$-Charaktergraph zu G_i aus $\mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ und dem $\mathcal{T}_{i-1}$-Charaktergraphen von G_{i-1}, die aus Schritt $i-1$ bekannt sind, berechnet. Hierbei werden zusätzlich die folgenden Daten benötigt:

(1) Die durch Konjugation definierte g_i-Operation auf $\mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$, die in Form einer Permutation π_{i-1} von $\mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ mit $F^{g_i} \sim \pi_{i-1}F$, $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$, gegeben ist.

(2) Die zugehörigen Verkettungsmatrizen $X_F \in \mathrm{Int}(F^{g_i}, \pi_{i-1}F)$, $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$.

Bis auf die Konstruktion der Daten (1) und (2) sind alle weiteren Konstruktionsschritte schon in der in Abschnitt 2.4 beschriebenen Hauptroutine angegeben. Der entscheidende Trick besteht nun darin, daß für überauflösbare Gruppen sowohl die g_i-Operation als auch die Verkettungsmatrizen ebenfalls iterativ entlang der Hauptreihe konstruiert werden können. Der Grund hierfür liegt darin, daß im überauflösbaren Fall die Gruppen G_m, $m = 1, \ldots, i-1$, normal in G und damit auch in G_i sind, so daß der Erzeuger g_i durch Konjugation auf $\mathrm{Irr}(G_m, \mathcal{T}_m)$ für $m = 1, \ldots, i-1$ operiert. (Dies ist für auflösbare Gruppen im allgemeinen falsch!) Darüber hinaus folgt aus dieser iterativen Konstruktion, die wir im folgenden auch als Unterroutine bezeichnen, daß die Verkettungsmatrizen e-monomial sind.

Unterroutine: Iterative Konstruktion von π_{i-1} und X_F

Nach Induktionsvoraussetzung der Hauptroutine sind sowohl $\mathrm{Irr}(G_m, \mathcal{T}_m)$ für $m = 0, \ldots, i-1$ als auch der $\mathcal{T}_{i-1}$-Charaktergraph von G_{i-1} bekannt. Die g_i-Operation auf $\mathrm{Irr}(G_0, \mathcal{T}_0)$ ist trivial und die zugehörige 1-dimensionale Verkettungsmatrix ist die Identität. Als Induktionsvoraussetzung der Unterroutine nehmen wir an, daß die g_i-Operation auf $\mathrm{Irr}(G_{m-1}, \mathcal{T}_{m-1})$ für $0 < m < i$ in Form der Permutation $\pi_{i,m-1}$ von $\mathrm{Irr}(G_{m-1}, \mathcal{T}_{m-1})$ und die zugehörigen Verkettungsmatrizen $X_F \in \mathrm{Int}(F^{g_i}, \pi_{i,m-1}F)$, $F \in \mathrm{Irr}(G_{m-1}, \mathcal{T}_{m-1})$, bekannt sind. Mit dieser Notation gilt $\pi_{i,i-1} = \pi_{i-1}$. Im folgenden Schritt ist nun die Permutation $\pi_{i,m}$ von $\mathrm{Irr}(G_m, \mathcal{T}_m)$ zu bestimmen. Wir fixieren ein $F \in \mathrm{Irr}(G_{m-1}, \mathcal{T}_{m-1})$. Dann müssen in Schritt m (analog zur Hauptroutine) zwei Fälle unterschieden werden:

Fall 1. $F \sim F^{g_m}$. In diesem Fall hat F genau $p = p_m$ Erweiterungen $D_0, \ldots, D_{p-1} \in \mathrm{Irr}(G_m, \mathcal{T}_m)$, die wegen der Induktionsvoraussetzung der Hauptroutine bekannt sind. Da D_k, $0 \leq k < p$, eine Erweiterung von F ist, folgt leicht, daß $\pi_{i,m} D_k$ eine Erweiterung von $\pi_{i,m-1} F$ sein muß. Seien $\Delta_0, \ldots, \Delta_{p-1}$ die Erweiterungen von $\pi_{i,m-1} F$. Dann kann man zeigen, daß $X_{D_k} := X_F$ für alle k gesetzt werden muß und $\pi_{i,m}(\{D_0, \ldots, D_{p-1}\}) = \{\Delta_0, \ldots, \Delta_{p-1}\}$ gilt. Mit Hilfe der X_{D_k} kann dann $\pi_{i,m}$ bestimmt werden (siehe [4]).

Fall 2. $F \not\sim F^{g_m}$. In diesem Fall induziert F eine Darstellung $D \in \mathrm{Irr}(G_m, \mathcal{T}_m)$ und $\pi_{i,m}(D)$ kann unmittelbar angegeben werden: Es ist die eindeutig bestimmte Darstellung $\Delta \in \mathrm{Irr}(G_m, \mathcal{T}_m)$, deren Einschränkung $\Delta \downarrow G_{m-1}$ die Darstellung $\pi_{i,m-1} F$ enthält. (Diese Information kann vom $\mathcal{T}_{i-1}$-Charaktergraph abgelesen werden.) Wir wollen hier nicht auf die Konstruktion von X_D eingehen, sondern verweisen auf [4].

Eine Analyse des BC-Algorithmus führt zu folgendem Ergebnis:

Satz 3.1.1 (Baum, Clausen, [4]) *Eine Transversale* $\mathrm{Irr}(G, \mathcal{T})$ *von irreduziblen e-monomialen Darstellungen einer überauflösbaren Gruppe* G *über* $\mathbb{C}$ *kann in*

$$O(|G| \log^2 |G|)$$

Operationen[1] *aus der pc-Präsentation von* G *berechnet werden. Dabei ist eine Operation eine Addition oder Subtraktion in* $\mathbb{Z}_e$.

Wir beschließen diesen Abschnitt mit den folgenden wichtigen Bemerkungen zum BC-Algorithmus:

(1) Die durch die O-Notation unterdrückte Konstante in der Abschätzung von Satz 3.1.1 ist klein (< 20), so daß diese Schranke auch für kleine Gruppen aussagekräftig ist.

(2) Die pc-Präsentation von G enthält bereits alle Informationen bezüglich der Gruppe, die der BC-Algorithmus benötigt, so daß keine Gruppenoperationen ausgeführt werden müssen.

(3) Der BC-Algorithmus ist in folgendem Sinne im wesentlichen optimal: Wir werden in Abschnitt 3.2 sehen, daß bei einer geeigneten Datenstruktur die Länge der Ausgabedaten des BC-Algorithmus proportional zur Gruppenordnung $|G|$ ist. Damit ist die Laufzeit des BC-Algorithmus bis auf logarithmische Faktoren proportional zur Länge der Ausgabedaten und kann in diesem Sinne als im wesentlichen optimal bezeichnet werden.

(4) Im Unterschied zur Originalversion in [4] wird in unserer Version der BC-Algorithmus in Teilen reorganisiert. Dadurch wird sichergestellt, daß der benötigte Speicherplatzbedarf auch während der Laufzeit im „worst case“ maximal linear in der Gruppenordnung $|G|$ ist. Diese Reorganisation hat keine Auswirkungen auf die Laufzeit.

[1] Bei der Implementierung dieses Algorithmus haben wir einen kleinen Fehler in der Komplexitätsanalyse in [4] entdeckt, der zu einem zusätzlichen Faktor $\log(|G|)$ und damit zur hier angegebenen Abschätzung führt.

(5) Die Kenntnis des Exponenten e ist für den BC-Algorithmus nicht notwendig. Die für die Matrixeinträge der e-monomialen Darstellungen minimal notwendige Ordnung der primitiven Einheitswurzel wird während der Laufzeit bestimmt: Angefangen mit $e = 1$ wird e um den Faktor p_i vergrößert, falls im i-ten Schritt der „bottom-up"-Konstruktion eine p_i-te Wurzel nicht berechnet werden kann. Wir verweisen auf Abschnitt 3.5.2 für eine weitere Diskussion.

3.2 DFT-Datenstruktur

Bei überauflösbaren Gruppen lassen sich die Ausgabedaten durch eine kompakte Datenstruktur realisieren, da wir es in diesem Fall nicht mit vollen Matrizen, sondern mit e-monomialen Matrizen zu tun haben. Da sich jede e-monomiale Matrix $M \in \mathbb{C}^{N\times N}$ in der Form

$$M = \pi \cdot \mathrm{diag}(\omega^{a_1}, \ldots, \omega^{a_N}) \tag{3.1}$$

schreiben läßt mit einer Permutation $\pi \in S_N$ und nicht-trivialen Koeffizienten $\omega^{a_1}, \ldots, \omega^{a_N}$, benötigt man nur die $2N$ ganzen Zahlen $\pi(1), \ldots, \pi(N)$ und $a_1, \ldots, a_N$, um M zu speichern.

Im folgenden wollen wir die für reale Anwendungen unserer Algorithmen vollkommen ausreichende Annahme machen, daß ganze Zahlen durch ein 32-Bit-Wort repräsentiert werden können. Darüber hinaus benötigt man für Zeiger ebenfalls ein 32-Bit-Wort. In der folgenden Diskussion ist der Speicherplatzbedarf also in 32-Bit-Wörtern angegeben (siehe auch Abschnitt 3.4).

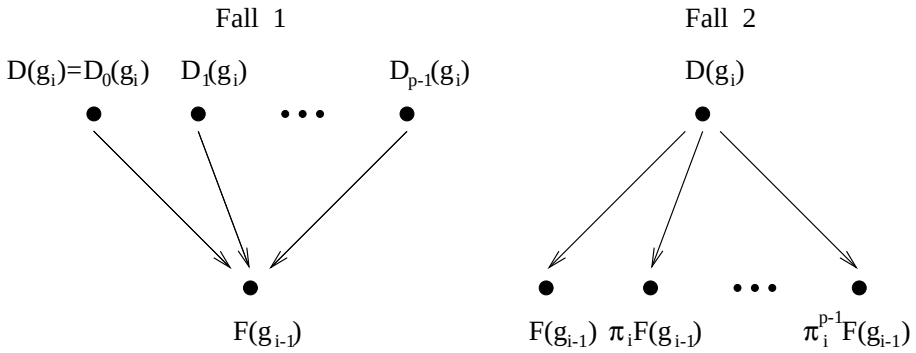
Der BC-Algorithmus berechnet $\mathrm{Irr}(G, \mathcal{T})$, wobei jede Darstellung $D \in \mathrm{Irr}(G, \mathcal{T})$ durch die darstellenden e-monomialen Matrizen $D(g_1), \ldots, D(g_n)$ auf den Erzeugern der pc-präsentierten Gruppe G gegeben ist. Speichert man diese Matrizen direkt ab, so ergibt sich mit (3.1) der Speicherbedarf von

$$2n \sum_{D\in\mathrm{Irr}(G,\mathcal{T})} \deg(D) = 2nd^1(G) \leq 2nd^2(G) \overset{(2.17)}{\leq} 2\log(|G|)|G|. \tag{3.2}$$

Die obere Schranke $2\log(|G|)|G|$ kann im schlimmsten Fall erreicht werden, was in der Praxis zu Speicherplatzproblemen führt. Wir geben eine Datenstruktur an, im folgenden auch als DFT-Datenstruktur bezeichnet, bei der garantiert ist, daß der maximale Speicherplatzbedarf linear in der Gruppenordnung $|G|$ ist. Als Grundgerüst für die DFT-Datenstruktur dient der $\mathcal{T}$-Charaktergraph von G. Jeder Knoten des Graphen entspricht eineindeutig einer Darstellung $D \in \mathrm{Irr}(G_i, \mathcal{T}_i)$ für ein i, $0 \leq i \leq n$. Daher sprechen wir im folgenden nicht nur von der Darstellung D, sondern auch vom Knoten D. Zu jedem Knoten assoziieren wir die folgenden Daten:

- Die e-monomiale Matrix $D(g_i)$ in Form von $2\deg(D)$ ganzen Zahlen.
- Einen Zeiger vom Knoten D zum Knoten F, falls $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ in $D{\downarrow}G_{i-1}$ vorkommt, also $\langle D|F\rangle = 1$ gilt.

Die Darstellung $D \in \mathrm{Irr}(G_i, \mathcal{T}_i)$ ist iterativ entweder durch Erweiterung oder durch Induktion einer Darstellung $F \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ entstanden. Damit hat das Datenformat lokal um D die folgende Form mit $p = p_i$:



Die Bezeichnungen sind dabei wie in Abschnitt 2.4. Im Fall 1 gilt zudem $D_k(g_i) = \chi^k(g_i) \cdot D(g_i)$, so daß anstelle der p Matrizen $D(g_i) = D_0(g_i), \ldots, D_{p-1}(g_i)$ nur die Matrix $D_0(g_i)$ und $p-1$ ganze Zahlen für die Exponenten der Einheitswurzeln $\chi^k(g_i)$, $1 \leq k < p$, abgespeichert werden müssen. Wir verweisen auf Abschnitt 3.3 für Beispiele zum Datenformat. Für den Speicherplatz gilt die folgende Abschätzung:

Lemma 3.2.1 *Die DFT-Datenstruktur benötigt Speicherplatz für maximal* $12|G|$ *Zeiger oder ganze Zahlen (32-Bit-Wörtern) und ist somit auch im schlimmsten Fall linear in der Gruppenordnung.*

Beweis: Es sei h_i die Anzahl der Konjugationsklassen von G_i. Dann gilt $h_i = |\mathrm{Irr}(G_i, \mathcal{T}_i)|$, und es gelten die offensichtlichen Abschätzungen

$$h_i \leq p_i \cdot h_{i-1} \quad \text{und} \quad h_i \leq |G_i|. \tag{3.3}$$

Weiterhin sei $\mathcal{E}_i \subset \mathrm{Irr}(G_i, \mathcal{T}_i)$ die Menge der Darstellungen auf Stufe i, die durch Erweiterungen entstanden sind (Fall 1), und $\mathcal{I}_i = \mathrm{Irr}(G_i, \mathcal{T}_i) \setminus \mathcal{E}_i$ die Menge der Darstellungen, die durch Induktion entstanden sind (Fall 2), also $|\mathcal{E}_i| + |\mathcal{I}_i| = h_i$. Für die Anzahl $P(i)$ an Zeigern, die den Kanten des Charaktergraphen zwischen Stufe i und Stufe $i-1$ entsprechen, gilt dann mit $|\mathcal{E}_i| = p_i \cdot (h_{i-1} - p_i \cdot |\mathcal{I}_i|)$:

$$P(i) = |\mathcal{E}_i| + p_i \cdot |\mathcal{I}_i| \leq p_i \cdot h_{i-1} \leq p_i \cdot G_{i-1} \leq |G_i|. \tag{3.4}$$

Für die Anzahl an ganzen Zahlen $M(i)$, die für die Matrixdaten $D(g_i)$ und die Einheitswurzeln $\chi^k(g_i)$ benötigt werden, gilt

$$\begin{aligned} M(i) &\leq 2 \cdot \frac{1}{p_i} \sum_{D \in \mathcal{E}_i} \deg(D) + |\mathcal{E}_i| + 2 \cdot \sum_{D \in \mathcal{I}_i} \deg(D) \\ &\leq 2 \cdot d^1(G_i) + h_i \;\leq\; 2 \cdot d^2(G_i) + h_i \\ &\overset{(2.17)}{\leq} 3|G_i|. \end{aligned} \tag{3.5}$$

Der Speicherbedarf $Z(i)$ auf Stufe i, der für Zusatzdaten (Zeiger auf die Matrizen $D(g_i)$ und Flags) benötigt wird, kann abgeschätzt werden durch

$$Z(i) \leq 2 \cdot h_i \leq 2 \cdot |G_i|. \tag{3.6}$$

Zusammen mit $|G_i| \leq 2^{i-n}|G_n|$ kann damit der Speicherbedarf SB für die Datenstruktur folgendermaßen abgeschätzt werden:

$$\text{SB} \leq \sum_{i=1}^{n}(P(i) + M(i) + Z(i)) \leq 6 \cdot \sum_{i=1}^{n}|G_i| \leq 12 \cdot |G_n|.$$

□

Die Abschätzungen sind im allgemeinen sehr grob. Aus (3.4), (3.5) und (3.6) folgt insbesondere die Abschätzung

$$\text{SB} \leq \sum_{i=1}^{n}\left(p_i \cdot h_{i-1} + 3 \cdot h_i + 2 \cdot d^1(G_i)\right). \tag{3.7}$$

Der Speicherbedarf hängt also von den Größen h_i und $d^1(G_i)$ ab. Hat man also viele Induktionen, so sind h_i und $d^1(G_i)$ klein im Verhältnis zu $|G_i|$ und der Speicherbedarf ist wesentlich geringer als die angegebene obere Schranke in Lemma 3.2.1. Der Speicherplatzbedarf hängt insbesondere von der Anzahl der Knoten $\kappa := \sum_{i=0}^{n} h_i$ im Charaktergraph von G ab. Es sei angemerkt, daß diese Zahl keine Gruppeninvariante der Gruppe G ist, sondern von der gewählten pc-Präsentation und der zugehörigen Hauptreihe $\mathcal{T}$ abhängt. Mit anderen Worten, die Wahl der pc-Präsentation für G hat einen entscheidenden Einfluß auf die Größe der DFT-Datenstruktur. Wir geben ein kleines Beispiel, um diesen Sachverhalt zu illustrieren. Sei $G = C_2 \times C_5$ das Produkt von zyklischen Gruppen. Dann besitzt G offensichtlich die beiden pc-Präsentationen

$$G \simeq \langle g_1, g_2 \,|\, g_1^5 = 1, g_2^2 = 1\rangle \qquad \text{und} \qquad G \simeq \langle g_1, g_2 \,|\, g_1^2 = 1, g_2^5 = 1\rangle \tag{3.8}$$

mit jeweils trivialen Kommutatorrelationen $[g_1, g_2] = 1$. Während für die erste pc-Präsentation $\kappa = 16$ gilt, hat man für die zweite den Wert $\kappa = 13$ (siehe Abbildung 3.1). In diesem Zusammenhang sei auch auf die Gruppen $G = \text{Lib}_{44100}(13)$ und $G = \text{Lib}_{44100}(21)$ aus Abschnitt 3.4.3 verwiesen.

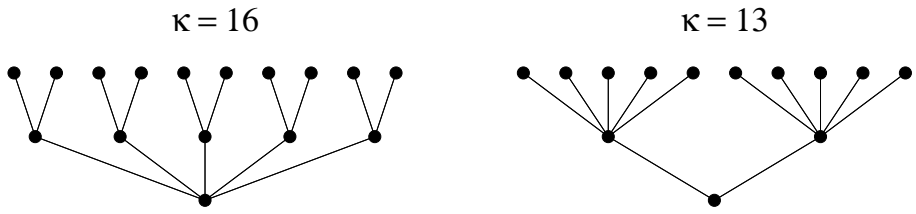


Abbildung 3.1: Charaktergraphen der zwei pc-Präsentationen (3.8) von $G = C_2 \times C_5$.

Abschließend sei bemerkt, daß die darstellenden Matrizen $D(g_i)$, $D \in \text{Irr}(G, \mathcal{T})$, für alle Erzeuger g_i, $1 \leq i \leq n$, aufgrund der $\mathcal{T}$-Adaptivität aus der Datenstruktur ablesbar sind. Außerdem eignet sich diese Datenstruktur sehr gut für eine schnelle Auswertung (FFT) der DFT, wie wir in Kapitel 5 sehen werden.

3.3 Beispiele

In diesem Abschnitt behandeln wir zwei einfache Beispiele: Die symmetrische Gruppe S_3 und den klassischen Fall der zyklischen Gruppe C_{2^n}. Im folgenden bezeichne $D_{i,k}$, $0 \leq i \leq n$,

$0 \leq k < h_i$, die k-te Darstellung von $\mathrm{Irr}(G_i, \mathcal{T}_i)$, wobei $D_{i,0}$ immer die triviale Darstellung sei und im Erweiterungsfall die p_i Erweiterungen hintereinander durchnumeriert seien. In den angegebenen pc-Präsentationen sind die trivialen Kommutatorrelationen nicht explizit aufgeführt.

3.3.1 Symmetrische Gruppe S_3

Die symmetrische Gruppe S_3 der Ordnung 6 hat die pc-Präsentation

$$S_3 \simeq \langle g_1, g_2 \,|\, g_1^3 = 1, g_2^2 = 1, [g_1, g_2] = g_1 \rangle, \tag{3.9}$$

wobei $(123) \mapsto g_1$ und $(12) \mapsto g_2$ eine mögliche Isomorphie definiert. Diese pc-Präsentation definiert die Hauptreihe

$$\mathcal{T} = (G = G_2 = S_3 \rhd G_1 = A_3 \rhd G_0 = S_1).$$

Der Charaktergraph und die Ausgabedaten des BC-Algorithmus in der DFT-Datenstruktur sind in Abbildung 3.2 dargestellt. Der Exponent von S_3 ist $e = 6$. Die Ziffern an den Knoten sind die Exponenten, in die die fest gewählte primitive e-te Einheitswurzel ω erhoben wird.

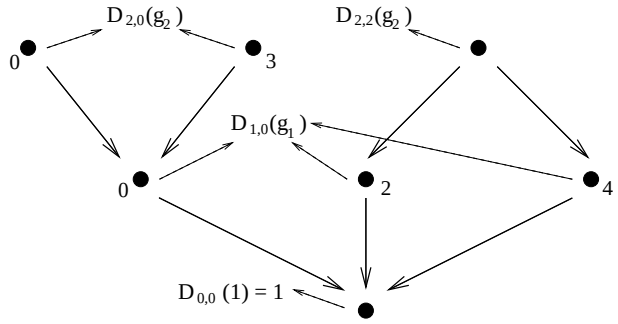


Abbildung 3.2: DFT-Datenstruktur und $\mathcal{T}$-Charaktergraph der S_3.

Die Darstellungen der S_3 sind durch die darstellenden Matrizen auf den Erzeugern g_1 und g_2 gegeben. Diese lassen sich aufgrund der $\mathcal{T}$-Adaptivität unmittelbar aus der Datenstruktur ablesen:

$$\begin{aligned} D_{2,0}(g_1) &= D_{1,0}(g_1) = 1, & D_{2,0}(g_2) &= 1, \\ D_{2,1}(g_1) &= D_{1,0}(g_1) = 1, & D_{2,1}(g_2) &= \omega^3 \cdot D_{2,0}(g_2) = \omega^3, \end{aligned}$$

$$\begin{aligned} D_{2,2}(g_1) &= D_{1,1}(g_1) \oplus D_{1,2}(g_1) \\ &= \begin{bmatrix} \omega^2 & 0 \\ 0 & \omega^4 \end{bmatrix}, & D_{2,2}(g_2) &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned} \tag{3.10}$$

Der Wedderburn-Isomorphismus nimmt damit qualitativ die folgende Gestalt an:

$$\mathbb{C}S_3 \quad \simeq \quad \begin{pmatrix} * & & \\ & * & \\ & & \begin{matrix} * & * \\ * & * \end{matrix} \end{pmatrix}.$$

3.3.2 Zyklische Gruppe C_{2^n}

Die zyklische Gruppe C_{2^n}, $n \in \mathbb{N}$, der Ordnung 2^n hat die pc-Präsentation

$$C_{2^n} \simeq \langle g_1, \ldots, g_n \,|\, g_1^2 = 1, g_2^2 = g_1, \ldots, g_n^2 = g_{n-1} \rangle. \tag{3.11}$$

Damit hat der Erzeuger g_n die Ordnung 2^n und erzeugt die gesamte Gruppe. In der zugehörigen Hauptreihe haben alle Hauptfaktoren die Ordnung 2. Der Charaktergraph von C_{2^n}, in Abbildung 3.3 für den Fall $n = 3$ dargestellt, ist ein vollständiger Binärbaum (wie allgemein für jede abelsche 2-Gruppe). In diesem Beispiel ist u. a. $D_{2,2}(g_2)$ eine primitive 4-te Einheitswurzel und $D_{3,4}(g_3)$ eine primitive 8-te Einheitswurzel.

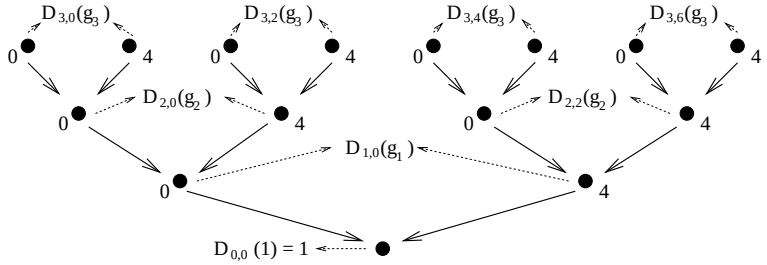


Abbildung 3.3: DFT-Datenstruktur und $\mathcal{T}$-Charaktergraph der C_8.

Die Darstellungen $D_{i,k}$ sind, wie bei allen abelschen Gruppen, 1-dimensional und im Fall der zyklischen Gruppe C_{2^n} eindeutig durch den Wert $D_{i,k}(g_i) \in \mathbb{C}$ bestimmt. Es sei

$$\beta_i : [0 : 2^i - 1] \longrightarrow [0 : 2^i - 1] \tag{3.12}$$

die Permutation, die einer Zahl $k \in [0 : 2^i - 1]$ mit Binärdarstellung $(b_{i-1} \ldots b_1 b_0) \in \{0,1\}^i$ die Zahl mit der Binärdarstellung $(b_0 b_1 \ldots b_{i-1})$ zuordnet. Aus dem BC-Algorithmus folgt dann induktiv

$$D_{i,k} : g_i \longmapsto \omega^{2^{n-i}\beta_i(k)} \tag{3.13}$$

für $1 \leq i \leq n$, $1 \leq k \leq 2^i - 1$ und einer primitiven 2^n-ten Einheitswurzel ω. Die Darstellung $D_{i-1,k}$ hat hierbei die zwei Erweiterungen $D_{i,2k}$ und $D_{i,2k+1}$.

Wir wollen abschließend bemerken, daß abelsche Gruppen im Hinblick auf den Speicherplatzbedarf der DFT-Datenstruktur (siehe (3.7)) und auf die Laufzeit des BC-Algorithmus den „worst case“ darstellen, bei dem $h_i = d^1(G_i) = |G_i|$ gilt.

3.4 Implementierung

Der BC-Algorithmus wurde in der Programmiersprache C/C++ implementiert. Es wurden zahlreiche Tests zur Bestimmung des tatsächlichen Speicherplatzbedarfs und der konkreten Laufzeiten der Implementierung durchgeführt. Einige der Ergebnisse sind in Abschnitt 3.4.3 zusammengefaßt. Für die Tests wurden als Eingabedaten des BC-Algorithmus pc-Präsentationen überauflösbarer Gruppen benötigt. Es wurde daher ein ganze Bibliothek solcher Präsentationen erzeugt, die wir in Abschnitt 3.4.1 beschreiben. In den Abschnitten 3.4.2 und 3.4.4 werden weitere technische Details bzw. Testalgorithmen für die Ergebnisse der Implementierung diskutiert.

3.4.1 Bibliothek von pc-Präsentationen

Die Eingabe des BC-Algorithmus besteht aus konsistenten pc-Präsentationen von endlichen überauflösbaren Gruppen. Um genügend Eingabedaten für die Experimente mit der Implementierung des BC-Algorithmus zu erhalten, wurden mit Hilfe des Computeralgebra-Systems GAP (Groups, Algorithms and Programming, siehe [31]) durch sukzessive zyklische Erweiterungen solche konsistenten pc-Präsentationen für überauflösbare und auch auflösbare Gruppen erzeugt. Im folgenden fassen wir kurz das Konstruktionsprinzip zusammen und verweisen für Einzelheiten auf Kapitel I, §14 von [35].

Sei G eine Gruppe mit normaler Untergruppe N vom Primindex $p = [G : N]$ und $g \in G \setminus N$, so daß gN ein Erzeuger der zyklische Gruppe G/N ist. Wegen der Normalität von N in G definiert g durch Konjugation einen Automorphismus $\alpha_g : N \to N$, $h \mapsto h^g := g^{-1}hg$. Ist speziell $h := g^p \in N$, dann folgt leicht

(i) $\alpha_g(h) = h^g = h$,

(ii) $(\alpha_g)^p = \mathrm{inn}_h$, wobei inn_h den inneren Automorphismus von N bezeichne, der durch Konjugation mit h induziert wird.

Diese Eigenschaften sind auch hinreichend für die Erweiterung einer Gruppe. Sei also p eine Primzahl, N eine Gruppe, $h \in N$ und $\alpha \in \mathrm{Aut}(N)$ mit den Eigenschaften

(i) $\alpha(h) = h$,

(ii) $(\alpha)^p = \mathrm{inn}_h$.

Dann erhält man eine Gruppenerweiterung durch Hinzunahme eines Elements g mit den Relationen $g^p := h$ und $g^{-1}hg := \alpha(h)$.

$$G := \mathrm{Ext}(N, p, h, \alpha) := \langle g, N \mid g^p = h,\ g^{-1}hg = \alpha(h) \rangle$$

definiert eine Gruppe der Ordnung $p \times |N|$, und N ist normal in G. Elemente $h \in N$ und $\alpha \in \mathrm{Aut}(N)$ mit den Eigenschaften (i) und (ii) bezeichnen wir im folgenden auch als *zulässige* Paare (h, α).

Mit Hilfe dieser Vorgehensweise können konsistente pc-Präsentationen endlicher auflösbarer Gruppen sukzessive in einer „bottom-up"-Konstruktion erzeugt werden. Hierzu müssen im Erweiterungsschritt die Automorphismengruppe von N berechnet und dann zulässige Elemente $h \in N$ und $\alpha \in \mathrm{Aut}(N)$ ausgewählt werden. Dieses Verfahren wurde in GAP implementiert, und durch randomisierte Auswahl von zulässigen Paaren (h, α) wurden in jedem Erweiterungsschritt pc-Präsentationen auflösbarer Gruppen erzeugt. Aus diesen wurden dann diejenigen pc-Präsentationen ausgewählt, die überauflösbare Gruppen definieren.

Auf diese Weise wurden über 1000 konsistente pc-Präsentationen erzeugt, die überauflösbare Gruppen unterschiedlichster Ordnung zwischen 10^3 und 10^7 definieren. Zusätzlich konnte auf eine GAP-Bibliothek zurückgegriffen werden, die aus Präsentationen für Vertreter aller Isomorphieklassen von Gruppen bis zur Ordnung 511 besteht (siehe auch [6]). Mit Hilfe dieser Präsentationen wurden für die überauflösbaren Gruppen bis zur Ordnung 511

konsistente pc-Präsentationen berechnet. Damit stehen z. B. 2328 konsistente pc-Präsentationen, die paarweise nicht-isomorphe 2-Gruppen der Ordnung 128 definieren, und analog 56092 pc-Präsentationen für Gruppen der Ordnung 256 zur Verfügung. Auf diese Weise wurde eine große Bibliothek von konsistenten pc-Präsentationen überauflösbarer Gruppen aufgebaut.

Im folgenden bezeichnen wir diese durch die pc-Präsentationen definierten Gruppen mit $G = \mathrm{Lib}_N(k)$, wobei N die Gruppengröße angibt und k die Nummer der Gruppe unter allen in der Bibliothek befindlichen Gruppen derselben Gruppenordnung N. Die Numerierung der Gruppen dient dabei nur zur Identifikation der Gruppen und hat keine tiefere Bedeutung.

3.4.2 Technische Details

Der BC-Algorithmus wurde in der Programmiersprache C/C++ implementiert. Die Tests wurden auf einem Intel Pentium III, 700 MHz mit 128 MByte Hauptspeicher, durchgeführt.

Wie schon erwähnt, wird beim BC-Algorithmus keine Körperarithmetik benötigt, sondern alle Manipulationen mit Matrixkoeffizienten sind Additionen oder Subtraktionen in der additiven Gruppe $\mathbb{Z}_e$. Da der Exponent e durch die Gruppenordnung $|G|$ abgeschätzt werden kann, bleiben alle Zwischenergebnisse, die bei Addition modulo e entstehen, unter der Schranke $2|G|$. Ebenso sind die ganzen Zahlen, die die Permutationen der darstellenden e-monomialen Matrizen beschreiben, sowie alle weiteren im BC-Algorithmus auftauchenden ganzen Zahlen und Zwischenergebnisse betragsmäßig durch $2|G|$ beschränkt. Bei unserer Implementierung des BC-Algorithmus wurde angenommen, daß alle ganzen Zahlen und Zeiger durch ein 32-Bit-Wort darstellbar sind. Damit kann also unsere Implementierung bis zu einer Gruppenordnung von $|G| \leq 2^{30} \approx 10^9$ eingesetzt werden, ohne daß es zu einem Überlauf bei den 32-Bit-Wörtern kommt.

Da der Speicherbedarf der DFT-Datenstruktur nach Lemma 3.2.1 durch $12|G|$ 32-Bit-Wörter abgeschätzt werden kann, ist bei einer Hauptspeicherkapazität von 128 MByte mit problemlosem Rechnen bis mindestens zu einer Gruppengröße von $|G| \approx 10^7$ zu rechnen. Dies wurde auch durch unsere Experimente bestätigt, wobei es erst ab Gruppen der Ordnung über diesem Wert zu Auslagerungen auf die Festplatte kam.

3.4.3 Laufzeiten und Speicherplatzbedarf

Wie schon erwähnt stellt die angegebene Größenordnung $O(|G| \log^2 |G|)$ aus Satz 3.1.1 eine obere Schranke für die Laufzeit im „worst case“ dar. Für die meisten Gruppen, insbesondere für nicht-abelsche Gruppen mit irreduziblen Darstellungen hohen Grades, ist das tatsächliche Laufzeitverhalten wesentlich günstiger. Theoretische Überlegungen wie auch die folgenden praktischen Ergebnisse zeigen, daß sich die Laufzeit des BC-Algorithmus (bis auf logarithmische Faktoren der Form $\log |G|$) im wesentlichen linear in der Größe $d^1(G)$ verhält, wobei $d^1(G)$ als die Summe der Grade über die Darstellungen von $\mathrm{Irr}(G)$ definiert ist (siehe (2.17)). Ähnliches läßt sich auch für den tatsächlichen Speicherplatzbedarf der DFT-Datenstruktur sagen, für den in Lemma 3.2.1 eine „worst case“-obere Schranke angegeben ist.

Die folgenden Tabellen zeigen die Laufzeiten und den Speicherplatzbedarf der Implementierung des BC-Algorithmus für einige überauflösbare Gruppen G der in Abschnitt 3.4.1

beschriebenen Bibliothek. Wie üblich bezeichne $|G|$ die Gruppenordnung von G, n die Länge der Kompositionsreihe und h die Anzahl der Konjugationsklassen. In der Spalte „SB“ ist der Speicherplatzbedarf in Byte für die DFT-Datenstruktur der jeweiligen Gruppe angegeben, während man in der Spalte „t“ die Laufzeit in Millisekunden ablesen kann, die zur Berechnung der DFT benötigt wurde. (Hierbei wurden für die Laufzeitbestimmung alle Berechnungen zehnmal durchgeführt und die durchschnittliche Laufzeit bestimmt.) Da sowohl der Speicherplatzbedarf SB als auch die Laufzeit t sich in etwa linear in $d^1(G)$ verhalten, wurden zur besseren Vergleichbarkeit von SB und t für verschiedene Gruppen G die jeweiligen Quotienten mit $d^1(G)$ berechnet.

Bei den Gruppen in Tabelle 3.1 handelt es sich um m-fache direkte Produkte der symmetrischen Gruppe S_3 für $m = 3, \ldots, 10$. Natürlich sind diese Gruppen von sehr einfacher Natur, deren DFT sich unmittelbar als m-fache Tensorprodukte der DFT der S_3 angeben lassen. Dennoch liefern diese Gruppen ein repräsentatives Bild zum Speicherplatzbedarf und Laufzeitverhalten der Implementierung des BC-Algorithmus, da diese nicht wesentlich von der Komplexität der pc-Präsentation, sondern hauptsächlich von der Anzahl und dem Grad der irreduziblen Darstellungen abhängen. Bei steigender Gruppengröße explodieren weder der Speicherplatzbedarf noch die Laufzeit, sondern beide Größen verhalten sich in etwa linear in $|G|$ bzw. $d^1(G)$.

G	$\lvert G\rvert$	n	h	$d^1(G)$	SB (byte)	SB/$d^1(G)$	t (ms)	$t/d^1(G)$
$(S_3)^3$	216	6	27	64	4728	73.9	<1	0
$(S_3)^4$	1296	8	81	256	14476	56.5	2	0.008
$(S_3)^5$	7776	10	243	1024	45704	44.6	18	0.018
$(S_3)^6$	46656	12	729	4096	147516	36.0	74	0.018
$(S_3)^7$	279936	14	2187	16384	485656	29.6	270	0.017
$(S_3)^8$	679616	16	16561	65536	1631084	24.9	1078	0.016
$(S_3)^9$	10077696	18	19683	262144	5591592	21.3	4844	0.019
$(S_3)^{10}$	60466176	20	59049	1048576	19570204	18.7	22105	0.021

Tabelle 3.1: SB und Laufzeiten in Abhängigkeit von der Gruppengröße.

In Tabelle 3.2 wurde der Algorithmus für den elementaren Fall der zyklischen Gruppen von Primzahlordnung getestet. Die Hauptreihen dieser Gruppen haben die Länge $n = 1$. Diese Beispiele zeigen, daß sich der Speicherplatzbedarf und das Laufzeitverhalten für solche einfachen Gruppen nicht von dem für komplexere Gruppen, wie z. B. die in den nächsten beiden Tabellen aufgeführten Gruppen, unterscheiden.

G	$\lvert G\rvert$	n	h	$d^1(G)$	SB (byte)	SB/$d^1(G)$	t (ms)	$t/d^1(G)$
C_{97}	97	1	97	97	3804	39.2	<1	0
C_{997}	997	1	997	997	36204	36.3	<1	0
C_{9973}	9973	1	9973	9973	359340	36.0	13	0.0013
C_{99991}	99991	1	99991	99991	3599988	36.0	104	0.0010
C_{999983}	999983	1	999983	999983	35999700	36.0	848	0.0008

Tabelle 3.2: SB und Laufzeiten für zyklische Gruppen von Primzahlordnung.

Die Gruppen aus Tabelle 3.3 haben alle die Ordnung $|G| = 44100$, unterscheiden sich jedoch wesentlich in der Anzahl der Konjugationsklassen und dem Grad ihrer irreduziblen

Darstellungen. Die ersten beiden Gruppen sind abelsch und stellen sowohl hinsichtlich des Speicherplatzbedarfs als auch hinsichtlich der Laufzeit ungünstige Fälle dar. Ist die Anzahl der Konjugationsklassen im Verhältnis zur Gruppengröße klein, so nehmen der Speicherplatzbedarf und die Laufzeit drastisch ab.

Interessant ist auch ein Vergleich der Gruppen $\mathrm{Lib}_{44100}(21)$ und $\mathrm{Lib}_{44100}(13)$, die bei gleichem h und $d^1(G)$ ein völlig unterschiedliches Speicherplatz- und Laufzeitverhalten aufweisen. Der Grund hierfür liegt darin, daß sich trotz gleicher Anzahl h der Blätter im Charaktergraphen die Anzahl κ der Knoten der jeweiligen Charaktergraphen stark unterscheiden. So hat die Gruppe $\mathrm{Lib}_{44100}(13)$ den Wert $\kappa = 15833$, während die Gruppe $\mathrm{Lib}_{44100}(21)$ den Wert $\kappa = 45681$ besitzt. Wie schon am Ende von Abschnitt 3.2 diskutiert und aus Formel (3.7) ersichtlich, hängt der tatsächliche Speicherplatzbedarf von dieser Zahl κ ab, die im Gegensatz zu h keine Gruppeninvariante ist, sondern von der gewählten pc-Präsentation und der zugehörigen Hauptreihe abhängt. Ebenso hängt die Laufzeit von dieser Zahl κ ab.

G	$\lvert G\rvert$	n	h	$d^1(G)$	SB (byte)	SB/$d^1(G)$	t (ms)	$t/d^1(G)$
$\mathrm{Lib}_{44100}(6)$	44100	8	44100	44100	2548088	57.8	244	0.0055
$\mathrm{Lib}_{44100}(10)$	44100	8	44100	44100	2333176	52.9	189	0.0043
$\mathrm{Lib}_{44100}(1)$	44100	8	22050	29400	2780712	94.6	441	0.0150
$\mathrm{Lib}_{44100}(20)$	44100	8	15750	25200	1048268	41.6	184	0.0073
$\mathrm{Lib}_{44100}(21)$	44100	8	13230	23520	2430024	103.3	437	0.0186
$\mathrm{Lib}_{44100}(13)$	44100	8	13230	23520	657932	28.0	73	0.0031
$\mathrm{Lib}_{44100}(2)$	44100	8	11025	19600	636716	32.5	102	0.0052
$\mathrm{Lib}_{44100}(3)$	44100	8	3750	10800	595336	55.1	138	0.0128
$\mathrm{Lib}_{44100}(19)$	44100	8	3675	8400	226808	27.0	50	0.0060

Tabelle 3.3: SB und Laufzeiten bei fester Gruppengröße aber variierendem h.

Zur Abrundung dieses Abschnitts ist in Tabelle 3.4 das Laufzeitverhalten unterschiedlichster Gruppen dargestellt. Die relativ langsame Laufzeit bei der 2-Sylowgruppe $\mathrm{Syl}_2(S_{16})$ der symmetrischen Gruppe S_{16} läßt sich mit dem vergleichsweise großen Wert $n = 15$ der Länge der Hauptreihe begründen. Das Vorliegen vieler kleiner Primteiler der Gruppenordnung wirkt sich negativ auf das tatsächliche Laufzeitverhalten des BC-Algorithmus aus.

G	$\lvert G\rvert$	n	h	$d^1(G)$	SB (byte)	SB/$d^1(G)$	t (ms)	$t/d^1(G)$
$\mathrm{Lib}_{1024}(1)$	1024	10	175	288	23028	80.0	4	0.0139
$\mathrm{Lib}_{2187}(1)$	2187	7	155	351	22612	64.4	3	0.0086
$\mathrm{Lib}_{7560}(1)$	7560	8	2295	4050	434272	107.0	111	0.0274
$\mathrm{Lib}_{7560}(54)$	7560	8	945	2016	58084	28.8	10	0.0050
$\mathrm{Lib}_{7776}(1)$	7776	10	333	1152	42936	37.3	15	0.0130
$\mathrm{Lib}_{15625}(35)$	15625	6	265	1225	32396	26.4	8	0.0065
$\mathrm{Lib}_{16000}(1)$	16000	10	424	2112	100408	47.5	67	0.0317
$\mathrm{Lib}_{22287}(1)$	22287	4	22287	22287	1304644	58.5	96	0.0043
$\mathrm{Lib}_{23940}(1)$	23940	7	3990	6840	483464	70.7	116	0.0170
$\mathrm{Syl}_2(S_{16})$	32768	15	230	2064	134760	65.3	144	0.0698
$\mathrm{Lib}_{42875}(1)$	42875	6	1595	4095	223812	54.7	35	0.0086
$\mathrm{Lib}_{179894}(3)$	179894	5	179894	179894	7441040	41.4	253	0.0014

Tabelle 3.4: SB und Laufzeiten für verschiedene Gruppen.

3.4.4 Testalgorithmen zur Implementierung

Um die Ergebnisse der Implementierung des BC-Algorithmus zu verifizieren, wurden diese weiteren Tests unterzogen. Auf diese Weise konnten Fehler in der Implementierung gefunden und behoben werden. Die jetzige Implementierung liefert (durch diese Testalgorithmen) nachweisbar für alle pc-Präsentationen der Bibliothek korrekte Ausgabedaten. Im einzelnen wurden die folgenden Tests durchgeführt:

(1) Jede Darstellung $D \in \mathrm{Irr}(G, \mathcal{T})$ ist durch ihre darstellenden Matrizen $D(g_1), \ldots, D(g_n)$ gegeben, die auch die Ausgabedaten des BC-Algorithmus ausmachen. Diese Matrizen müssen dieselben Potenz- und Kommutatorrelationen erfüllen wie die Erzeuger $g_1, \ldots, g_n$. Die durch die Implementierung berechneten darstellenden Matrizen wurden im Hinblick auf diese Relationen getestet, so daß damit nachgewiesen wurde, daß diese tatsächlich Darstellungen von G definieren.

(2) Es gilt $\sum_{D \in \mathrm{Irr}(G,\mathcal{T})} \deg(D)^2 = |G|$. Diese Bedingung wurde für die Dimensionen der darstellenden Matrizen überprüft.

(3) Nach dem Satz 2.2.1 (Wedderburn) ist $\oplus_{D \in \mathrm{Irr}(G,\mathcal{T})} D$ ein Isomorphismus von $\mathbb{C}G$ in eine Algebra von Blockdiagonalmatrizen. Die direkte Summe der durch die Implementierung berechneten Darstellungen wurde auf ihre Invertierbarkeit hin getestet. Hierzu wurden die in Kapitel 5 beschriebenen schnellen Fouriertransformationen verwendet.

Man kann zeigen, daß die in (1),(2) und (3) formulierten notwendigen Bedingungen für die Richtigkeit der berechneten darstellenden Matrizen auch hinreichend sind. Aufgrund dieser Tests konnte die Korrektheit der Implementierung des BC-Algorithmus auf der Testmenge der pc-Präsentationen unserer Bibliothek nachgewiesen werden.

3.5 Zusammenfassung und Ausblick

3.5.1 Zerfällungskörper

In diesem Kapitel wurde der BC-Algorithmus besprochen, der für jede endliche überauflösbare pc-präsentierte Gruppe G eine Transversale an paarweise inäquivalenten irreduziblen e-monomialen Darstellungen von G über dem Körper $\mathbb{C}$ berechnet. Wie schon in [15], S. 124, angemerkt, ist es dabei nicht notwendig, über $\mathbb{C}$ zu arbeiten. Es sei e der Exponent von G. Nach einem klassischen Resultat von Brauer ist jeder Körper $\mathbb{K}$, der eine primitive e-te Einheitswurzel enthält, ein *Zerfällungskörper* von G (d. h. der Gruppenring $\mathbb{K}[G]$ ist eine direkte Summe von vollen Matrixringen über $\mathbb{K}$, siehe [35], S.520 ff.). Die durch den BC-Algorithmus berechneten Exponenten in $\mathbb{Z}_e$ können dann als Exponenten dieser Einheitswurzel in $\mathbb{K}$ aufgefaßt werden. Damit ist der BC-Algorithmus eine universelle Konstruktion über allen Körpern mit einer primitiven e-ten Einheitswurzel.

3.5.2 Abhängigkeiten von der Hauptreihe

Für die Komplexität des BC-Algorithmus und den Speicherplatzbedarf der DFT-Datenstruktur können mit $O(|G|\log^2|G|)$ bzw. $O(|G|)$ obere Schranken angegeben werden, die nur von der Gruppenordnung, also einer Gruppeninvarianten, abhängen. Diese oberen Schranken sind im allgemeinen sehr grob und werden nur für abelsche Gruppen angenommen. Wie sich schon mehrfach in diesem Kapitel angedeutet hat, hängt die Laufzeit und der Speicherplatzbedarf viel mehr von der Struktur des Charaktergraphen ab, der durch die Hauptreihe $\mathcal{T}$ bestimmt ist. Für eine feste Gruppe G gibt es im allgemeinen viele Möglichkeiten zur Wahl einer Hauptreihe, die zu verschiedenen Charaktergraphen von G führen können. In Abbildung 3.1 sind z. B. zwei Charaktergraphen zu $C_2 \times C_5$ angegeben, deren Knotenzahl κ sich unterscheiden. Es wäre daher interessant, die folgenden Fragestellungen weiter zu untersuchen:

(1) Wie sieht ein geeignetes Komplexitätsmaß für eine Hauptreihe $\mathcal{T}$ aus? Dieses Komplexitätsmaß sollte die Anzahl der Knoten κ zum zugehörigen Charaktergraphen und die Grade der zu diesen Knoten gehörigen Darstellungen berücksichtigen.

(2) Es sollte eine Analyse des BC-Algorithmus und der DFT-Datenstruktur bezüglich dieses Komplexitätsmaßes durchgeführt werden. Obere Schranken bez. dieses Komplexitätsmaßes reflektieren wesentlich besser das tatsächliche Laufzeitverhalten.

(3) Folgende Fragestellung ist wesentlich schwieriger: Wie kann man für eine feste Gruppe G eine Hauptreihe $\mathcal{T}$ finden, bei der die Anzahl der benötigten Operationen und der Speicherplatzbedarf zur Berechnung der DFT minimal unter allen möglichen Hauptreihen von G sind?

Die Anzahl der Knoten κ und die Grade der zugehörigen Darstellungen sind nicht der einzige interessante Aspekt bei der Wahl der Hauptreihe. Die durch den BC-Algorithmus berechneten Matrixdarstellungen sind e-monomial, d. h. insbesondere sind alle Matrixeinträge e-te Einheitswurzeln, wobei e den Exponenten der Gruppe G bezeichnet. Diese Aussage ist unabhängig von der Wahl der Hauptreihe $\mathcal{T}$ von G. Im allgemeinen sind die Matrixeinträge sogar f-te Einheitswurzeln, wobei $f \in \mathbb{N}$ ein Teiler von e ist. Die kleinste solche Zahl f, für die alle Einträge der Matrixdarstellungen der DFT f-te Einheitswurzeln sind, ist von der Wahl der Hauptreihe $\mathcal{T}$ abhängig. Wir nennen diese Zahl im folgenden *Exponenten der Hauptreihe* $\mathcal{T}$ und bezeichnen sie mit $e(\mathcal{T})$. Es gilt also $e(\mathcal{T}) \mid e$. Wie schon in (5) der abschließenden Bemerkung von Abschnitt 3.1 erwähnt wurde, wird die Zahl $e(\mathcal{T})$ während der Laufzeit des BC-Algorithmus konstruiert. Aus der Konstruktion folgt $e(\mathcal{T}_i) \in \{e(\mathcal{T}_{i-1}), p_i \cdot e(\mathcal{T}_i)\}$. (Mit den Bezeichnungen von [15], S. 119, gilt, daß der zweite Fall genau dann eintritt, wenn die Gleichung $c^{p_i} \cdot X^{p_i} = F(g_i^p)$ über den $e(\mathcal{T}_{i-1})$-ten Einheitswurzeln nicht lösbar ist.)

Clausen und Baum geben für den Fall $e \neq e(\mathcal{T})$ schon auf dem Titelbild ihres Buches [15] folgendes Beispiel: Es bezeichne D_4 die Diedergruppe der Ordnung 8, die als Symmetriegruppe eines Quadrats durch die Spieglung S, die 90°-Drehung D_{90} und die 180°-Drehung D_{180} erzeugt wird (siehe Abbildung 3.4). Auf der rechten Seite dieser Abbildung findet man den Untergruppenverband der D_4, wobei Normalteiler den Quadraten und die übrigen Untergruppen den Kreisen entsprechen. Durch waagrechte Kanten verbundene Untergruppen sind zueinander konjugiert.

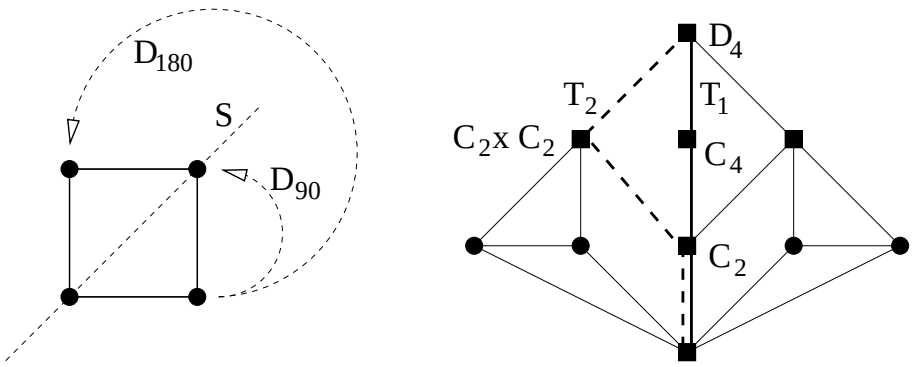


Abbildung 3.4: Diedergruppe D_4 der Ordnung 8 mit Untergruppenverband.

Für D_4 lassen sich die durch die folgenden pc-Präsentationen definierten Hauptreihen $\mathcal{T}_1$ und $\mathcal{T}_2$ angeben:

$$\mathcal{T}_1: \quad \langle g_1 = D_{180},\, g_2 = D_{90},\, g_3 = S \,|\, g_1^2 = 1,\, g_2^2 = g_2,\, g_3^2 = 1,\, [g_2, g_3] = g_1 \rangle,$$
$$\mathcal{T}_2: \quad \langle g_1 = D_{180},\, g_2 = S,\, g_3 = D_{90} \,|\, g_1^2 = 1,\, g_2^2 = 1,\, g_3^2 = g_1,\, [g_2, g_3] = g_1 \rangle.$$

Diese Hauptreihen definieren die in Abbildung 3.5 dargestellten Charaktergraphen, deren Struktur übereinstimmt. An den Knoten sind jeweils die Werte der eindimensionalen Darstellungen an den entsprechenden Erzeugern dieser Stufe angegeben. Die zweidimensionalen Darstellungen D_1 bzw. D_2, die bezüglich $\mathcal{T}_1$ bzw. $\mathcal{T}_2$ ausgerechnet wurden, sehen wie folgt aus:

$$D_1(g_1) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad D_1(g_2) = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}, \quad D_1(g_3) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$
$$D_2(g_1) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad D_2(g_2) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad D_2(g_3) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Damit gilt also $e(\mathcal{T}_1) = e = 4$ und $e(\mathcal{T}_2) = 2$.

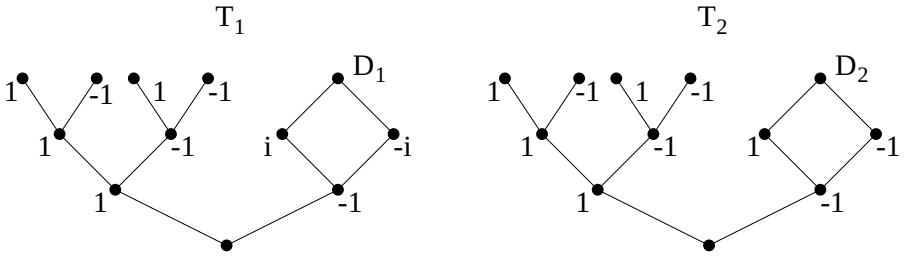


Abbildung 3.5: Charaktergraphen für die zwei Hauptreihen $\mathcal{T}_1$ und $\mathcal{T}_2$ von D_4.

Eine weitere interessante Fragestellung ist, wie für eine feste Gruppe G die Hauptreihe $\mathcal{T}$ gefunden werden kann, deren Exponent $e(\mathcal{T})$ minimal unter allen möglichen Hauptreihen ist. Dies kann im Hinblick auf eine effiziente Speicherung (je kleiner $e(\mathcal{T})$, desto kleiner die abzuspeichernden Exponenten für die Matrixeinträge der DFT) und im Hinblick auf eine numerisch stabile Auswertung der DFT (siehe Abschitt 5.5.2 und Tabelle 5.8) von Interesse sein.

3.5.3 Inkonsistente Präsentationen

Wir wollen abschließend noch einmal auf die Eingabedaten des BC-Algorithmus zu sprechen kommen, die aus den in Abschnitt 2.3 beschriebenen pc-Präsentationen endlicher überauflösbarer Gruppen bestehen. Die vorausgesetzte *Konsistenz* dieser Präsentationen ist dabei ganz entscheidend für den BC-Algorithmus. Wir formalisieren unser Problem. Wir bezeichnen im folgenden ein Tupel der Form

$$\Big(n \in \mathbb{N},\ (p_i \in \mathbb{N})_{1\le i\le n},\ (u_i \in G_{i-1})_{1\le i\le n},\ (w_{ij} \in G_{j-1})_{1\le i<j\le n}\Big), \tag{3.14}$$

mit Primzahlen p_i und in Normalform gegebenen Wörtern u_i und w_{ij} (siehe (2.8)) kurz als *pc-Daten*. Solche pc-Daten definieren durch

$$G := \langle g_1, \ldots, g_n \mid g_i^{p_i} = u_i\ (1 \le i \le n), [g_i, g_j] = w_{ij}\ (1 \le i < j \le n)\rangle, \tag{3.15}$$

immer eine auflösbare Gruppe, deren Ordnung $|G|$ ein Teiler der Zahl $N := p_1 \cdot \ldots \cdot p_n$ ist. Gilt zusätzlich $w_{ij} \in G_i$, so ist G immer überauflösbar. Wir nennen daher solche pc-Daten auch *überauflösbar*. Die pc-Daten definieren genau dann eine konsistente pc-Präsentation für G, falls $|G| = N$ gilt (siehe [55]). In diesem Fall sprechen wir dann auch von *konsistenten pc-Daten*. Zum Beispiel ist die durch

$$G := \langle g_1, g_2 \mid g_1^3 = 1, g_2^2 = g_1, [g_1, g_2] = g_1\rangle, \tag{3.16}$$

definierte Gruppe isomorph zur zyklischen Gruppe C_2 (es läßt sich die Relation $g_1 = 1$ ableiten). Die zugehörigen pc-Daten sind damit - im Gegensatz zu denen in (3.9) - nicht konsistent. Die Erzeugung konsistenter Präsentationen ist aus Sicht der Komplexitätstheorie ein schwieriges Problem. Für weitere Referenzen hierzu verweisen wir auf [11], S. 35.

Allgemein übliche Verfahren zum Test auf Konsistenz und zur Konstruktion von konsistenten Präsentationen beruhen auf dem *Knuth-Bendix Verfahren* [37]. Für den Spezialfall polyzyklischer Gruppen lassen sich diese Verfahren vereinfachen (siehe [55], Kapitel 9).

Der BC-Algorithmus wurde durch Hinzufügen einiger kleiner Routinen erweitert, so daß er bei Eingabe von überauflösbaren pc-Daten genau dann eine DFT von G konstruiert, wenn die pc-Daten konsistent sind. Im Fall von nicht-konsistenten Daten bricht der Algorithmus ab und gibt eine Fehlermeldung aus. Eine interessante Aufgabenstellung ist es nun, einen effizienten, auf DFT-Methoden basierenden Algorithmus zu entwerfen, der aus beliebigen pc-Daten konsistente pc-Präsentationen konstruiert. Im Spezialfall von pc-Daten könnten DFT-Methoden effizienter sein als die Vervollständigungsmethode des Knuth-Bendix-Verfahrens, und sie erlauben möglicherweise explizite obere Schranken für die Zahl der benötigten arithmetischen Operationen.

Kapitel 4

DFT-basierte Wortnormalisierung

Wie bereits erwähnt, kann jedes Element a einer pc-präsentierten überauflösbaren Gruppe G in Normalform

$$a = g^{\alpha} := g_n^{\alpha_n} \cdot g_{n-1}^{\alpha_{n-1}} \cdot \ldots \cdot g_1^{\alpha_1}$$

ausgedrückt werden. Sind zwei Elemente $a = g^{\alpha}$ und $b = g^{\beta}$ in Normalform gegeben, so besteht das Problem der *Wortnormalisierung* darin, die eindeutig bestimmte Normalform für das Produkt

$$g^{\alpha} \cdot g^{\beta} = g^{\gamma}$$

zu bestimmen. Klassische Techniken für die Lösung des Normalformenproblems beruhen auf verschiedenen „Collection"-Verfahren (siehe z. B. [55]) oder auf Hall-Polynomen in Verbindung mit Interpolationstechniken (siehe z. B. [56]). Unseres Wissens gibt es kein „bestes" Verfahren, welches allen anderen Verfahren überlegen wäre. In diesem Kapitel stellen wir als Alternative ein DFT-basiertes Verfahren zur Wortnormalisierung vor. Abkürzend bezeichnen wir den entsprechenden Algorithmus als WN-Algorithmus, der in den ersten beiden Abschnitten angegeben und analysiert wird. Der WN-Algorithmus wurde implementiert und getestet. Die Versuchsergebnisse sind in Abschnitt 4.3 zusammengefaßt. In Abschnitt 4.4 wird eine Brücke zur algorithmischen algebraischen Geometrie geschlagen. Wir zeigen, wie sich der WN-Algorithmus für die multivariate Polynomdivision bez. Gitterideale mit endlichem Index einsetzen läßt und verallgemeinern diese Ideen in Abschnitt 4.5 auf den Fall nicht-kommutativer Polynomringe.

4.1 WN-Algorithmus

Vor einer detaillierten Beschreibung des WN-Algorithmus wollen wir kurz auf die Hauptidee eingehen. Wie oben seien a und b Elemente einer überauflösbaren pc-präsentierten Gruppe G mit den üblichen Bezeichnungen. In einer Vorverarbeitungsphase wird eine e-monomiale DFT von G berechnet, die wir mit $\mathbf{D}$ bezeichnen. Anstelle a und b direkt (im „Zeitbereich") zu multiplizieren, berechnen wir zuerst die Fouriertransformationen $\mathbf{D}(a)$ und $\mathbf{D}(b)$ (Transformation in den „Spektralbereich"). Im neuen Transformationsbereich hat man es nun mit Multiplikationen von Matrizen sehr einfacher Struktur zu tun. Da $\mathbf{D}(a) \cdot \mathbf{D}(b) = \mathbf{D}(ab)$ gilt,

kann nun die Normalform für ab aus $\mathbf{D}(ab)$ extrahiert werden. Dies entspricht der inversen Fouriertransformation und kann in unserem Fall sehr effizient berechnet werden.

Wir wollen im folgenden das Problem der Wortnormalisierung allgemeiner betrachten. Es sei an dieser Stelle an die in Bemerkung 2.3.1 getroffenen Konventionen erinnert. Sei w ein beliebiges Wort in den Erzeugern $g_1, \ldots, g_n$ gegeben durch

$$w \;=\; {g_{\varphi(1)}}^{\psi(1)} \cdot {g_{\varphi(2)}}^{\psi(2)} \cdot \ldots \cdot {g_{\varphi(L)}}^{\psi(L)} \tag{4.1}$$

mit einem $L \in \mathbb{N}$ und zwei Funktionen $\varphi : \{1, \ldots, L\} \to \{1, \ldots, n\}$ und $\psi : \{1, \ldots, L\} \to \mathbb{Z} \setminus \{0\}$. Durch Zusammenfassen von Exponenten kann $\varphi(\ell) \neq \varphi(\ell + 1)$ für $1 \leq \ell < L$ angenommen werden. Sind darüber hinaus die Exponenten $\exp(G_i)$ der Untergruppen G_i bekannt, so kann $\psi(\ell)$ durch $\psi(\ell) \bmod \exp(G_{\varphi(\ell)})$, $1 \leq \ell \leq L$, ersetzt werden. Wir definieren die *Komplexität* $C(w)$ des Wortes w durch

$$C(w) := L + \sum_{\ell=1}^{L} \lfloor \log(|\psi(\ell)|) \rfloor, \tag{4.2}$$

welches gerade die Summe der Länge L und der Binärlängen $\lfloor \log(|\psi(\ell)|) \rfloor$ der Exponenten $\psi(\ell)$ ist. Diese Definition wird dadurch gerechtfertigt, daß in den späteren Berechnungen im Fall der Potenzen ${g_{\varphi(\ell)}}^{\psi(\ell)}$ die binären Methode verwendet werden kann. Das Problem der Wortnormalisierung besteht nun darin, die eindeutig bestimmte Normalform

$$g_n^{e_n} \cdot \ldots \cdot g_1^{e_1} \;\sim\; w \tag{4.3}$$

zu berechnen, also die Exponenten $0 \leq e_k < p_k$ für $k = 1, \ldots, n$.

Beispiel 4.1.1 Sei $G := S_3 \simeq \langle g_1, g_2 \,|\, g_1^3 = 1, g_2^2 = 1, [g_1, g_2] = g_1 \rangle$ die symmetrische Gruppe mit den Bezeichnungen von Abschnitt 3.3.1. Durch

$$w := g_1 \cdot g_2^2 \cdot g_1^3 \cdot g_2^4 \cdot g_1^5 \cdot g_2 \cdot g_1^2 \cdot g_2^3 \cdot g_1^4 \cdot g_2^5$$

ist ein Wort in den Erzeugern g_1 und g_2 der Länge $L = 10$ und der Komplexität $C(w) = 22$ definiert. Allein aus der Kenntnis der Exponenten $\exp(G_1) = 3$ und $\exp(G) = 6$ läßt sich unter Verwendung der dadurch implizierten Relationen $g_1^3 = 1$ und $g_2^6 = 1$ das Wort w zu

$$g_2 \cdot g_1^2 \cdot g_2^3 \cdot g_1 \cdot g_2^5 \;\sim\; w$$

vereinfachen. Unter Verwendung der pc-Relationen erhählt man schließlich die Normalform

$$g_2 \cdot g_1 \;\sim\; w,$$

es gilt also mit den obigen Bezeichnungen $e_1 = 1$ und $e_2 = 1$. □

Wir verwenden im folgenden dieselben Bezeichnungen wie in Kapitel 3. Es seien also G eine endliche überauflösbare pc-präsentierte Gruppe und $\mathrm{Irr}(G_i, \mathcal{T}_i)$, $i = 1, \ldots, n$, Transversalen $\mathcal{T}_i$-angepaßter e-monomialer irreduzibler Darstellungen. In einer Vorverarbeitungsphase, die unabhängig vom Problem der Wortnormalisierung ist, wird durch den BC-Algorithmus eine $\mathcal{T}$-adaptierte DFT von G berechnet, die in Form der DFT-Datenstruktur gespeichert

wird. Darüber hinaus werden alle Matrizen der DFT-Datenstruktur invertiert und abgespeichert. Insgesamt benötigt die Vorverarbeitungsphase eine Laufzeit der Größenordnung $O(|G|\log^2(|G|))$ und einen Speicherplatzbedarf von $O(|G|)$ ganzen Zahlen und Zeigern (32-Bit-Wörtern).

Zur Illustration der Idee des WN-Algorithmus erklären wir zuerst, wie der Koeffzient e_n von w bestimmt werden kann. Es sei $D_{n-1,0} \in \mathrm{Irr}(G_{n-1}, \mathcal{T}_{n-1})$ die triviale Darstellung, dann hat $D_{n-1,0}$ genau $p = p_n$ eindimensionale Erweiterungen, die wir mit $D_{n,0}, \ldots, D_{n,p-1} \in \mathrm{Irr}(G_n, \mathcal{T}_n)$ bezeichnen. Dabei sei die Numerierung so gewählt, daß $D_{n,k}(g_n) = \varrho^k$ für $k = 0, \ldots, p-1$ gelte, wobei $\varrho := \omega^{e/p}$ eine primitive p-te Einheitswurzel ist (siehe Abschnitt 2.4). Aus der Auswertung

$$\begin{aligned} D_{n,1}(w) &= D_{n,1}(g_{\varphi(1)})^{\psi(1)} \cdot D_{n,1}(g_{\varphi(2)})^{\psi(2)} \cdot \ldots \cdot D_{n,1}(g_{\varphi(L)})^{\psi(L)} \\ &= \prod_{\ell,\varphi(\ell)=n} D_{n,1}(g_n)^{\psi(\ell)} = \varrho^{\left(\sum_{\ell,\varphi(\ell)=n} \psi(\ell)\right) \bmod p} \end{aligned}$$

kann unmittelbar der Exponent

$$e_n = \left(\sum_{\ell,\varphi(\ell)=n} \psi(\ell)\right) \bmod p$$

abgelesen werden, wobei benutzt wurde, daß $D_{n,1}(g_k)$ für $1 \le k \le n-1$ trivial ist. Wir wissen nun, daß $w_{n-1} := g_n^{-e_n} w$ ein Element in G_{n-1} ist. Allerdings ist w_{n-1} im allgemeinen kein Wort in den Erzeugern $g_1, \ldots, g_{n-1}$, so daß nicht einfach auf dieselbe Weise induktiv vorgegangen werden kann.

Der Fall $i = n$ stellt den Anfang des iterativen WN-Algorithmus dar. Es seien bereits die Exponenten $e_n, \ldots, e_{i+1}$, $i \ge 1$ bestimmt, dann ist nun der Exponent e_i zu bestimmen. Sei $D_{i,1} \in \mathrm{Irr}(G_i, \mathcal{T}_i)$, analog zu $D_{n,1}$ wie oben im Fall $i = n$, die Erweiterung der trivialen Darstellung $D_{i-1,0} \in \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$ mit $D_{i,1}(g_i) = \varrho$, wobei $\varrho := \omega^{e/p_i}$ eine primitive p_i-te Einheitswurzel ist. Wir wollen nun $D_{i,1}$ auf w_i auswerten mit

$$w_i := g_{i+1}^{-e_{i+1}} \cdot \ldots \cdot g_n^{-e_n} w \in G_i. \tag{4.4}$$

Da w_i zwar in G_i liegt, aber im allgemeinen ein Wort in den Erzeugern $g_1, \ldots, g_n$ ist, kann $D_{i,1}(w_i)$ nicht direkt berechnet werden. Der Trick besteht nun darin, eine Darstellung $D \in \mathrm{Irr}(G_n, \mathcal{T}_n)$ zu verwenden, deren Einschränkung auf G_i die Darstellung $D_{i,1}$ enthält, also $\langle D, D_{i,1}\rangle \ge 1$ gilt. Ein solches D kann unmittelbar aus dem $\mathcal{T}$-Charaktergraphen abgelesen werden, der Teil der DFT-Datenstruktur ist. Aus Satz 2.1.2 (Clifford) folgt sofort, daß $D \downarrow G_i$ die direkte Summe eindimensionaler Darstellungen ist, wobei $D_{i,1}$ mindestens einmal unter den Summanden vorkommt. Wir numerieren die Summanden von 1 bis $\deg(D)$ und nehmen an, daß $D_{i,1}$ der m-te Summand von $D \downarrow G_i$ für ein $1 \le m \le \deg(D)$ ist. Dann erhält man den Exponenten e_i auf die folgende Weise.

- Berechne $D(w_i) = D(g_{i+1})^{-e_{i+1}} \cdot \ldots \cdot D(g_n)^{-e_n} \cdot D(g_{\varphi(1)})^{\psi(1)} \cdot \ldots \cdot D(g_{\varphi(L)})^{\psi(L)}$.
- Wegen $w_i \in G_i$ ist dies eine Diagonalmatrix, deren m-ter Eintrag $D_{i,1}(w_i)$ ist.
- Analog zum Fall $i = n$ ist $D_{i,1}(w_i)$ eine Potenz der p_i-ten Einheitswurzel ϱ, deren Exponent der gesuchte Wert e_i ist.

In Abbildung 4.1 sind am Beispiel der Gruppe G_{128} die Darstellungen $D_{i,1}$ durch einen Kreis und die entsprechenden Erweiterungen D auf Stufe $n = 7$ durch ein Quadrat gekennzeichnet. Zum Beispiel gilt für $D = D_{7,11}$, daß $D{\downarrow}G_2$ gleich der 4-fachen direkten Summe von $D_{2,1}$ ist. Damit ist jeder nicht-triviale Eintrag von $D(w_i)$ gleich $D_{2,1}(w_i)$.

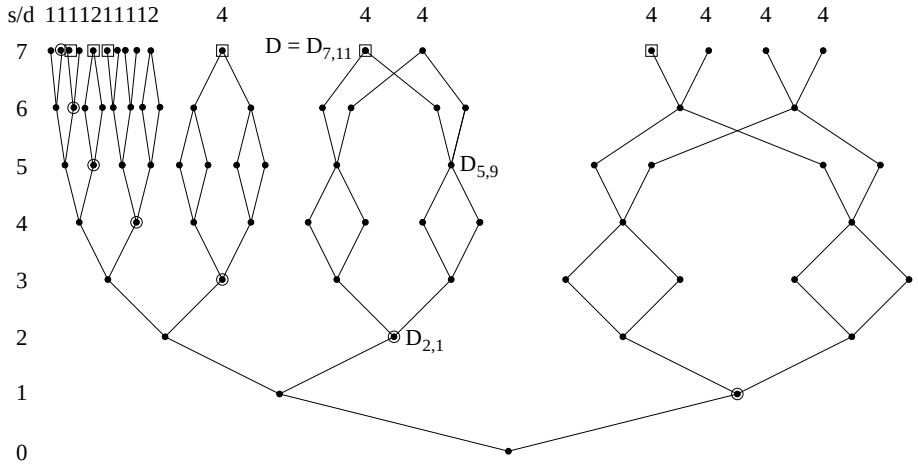


Abbildung 4.1: WN-Algorithmus am Beispiel von G_{128}.

Die Berechnung von $D_{i,1}(w_i)$ kann sehr effizient durchgeführt werden, da

- D eine e-monomiale Darstellung ist,
- nur der m-te Eintrag von $D(w_i)$ berechnet werden muß,
- $D \downarrow G_{i-1}$ trivial ist und
- anstelle $D(g_j)$, $i \leq j < n$, die Einschränkung $D{\downarrow}G_j(g_j)$ betrachtet werden kann (was auch konform zur DFT-Datenstruktur ist). Zum Beispiel kann die Berechnung des dritten Eintrages von $D_{7,11}(g_5)$ von Abbildung 4.1 auf die Berechnung des ersten Eintrages von $D_{5,9}(g_5)$ zurückgeführt werden.

Wir verschieben die Analyse und weitere Details des WN-Algorithmus auf den Abschnitt 4.2 und formulieren das Hauptergebnis.

Satz 4.1.2 *Sei G eine endliche pc-präsentierte überauflösbare Gruppe mit Kompositionsreihe $\mathcal{T}$ der Länge n, Exponent e und maximalem Primfaktor $p_{\max}$. Weiterhin sei die DFT-Datenstruktur von G und deren inversen Matrizen gegeben ($O(|G|)$ Speicherplatz). Sei $d_{\max} := \max_{D \in \mathrm{Irr}(G,\mathcal{T})}(\deg(D))$ und $A(w)$ die Anzahl der Additionen in $\mathbb{Z}_e$, die zur Berechnung der Normalform eines Wortes w der Komplexität $C(w)$ mit dem WN-Algorithmus benötigt werden. Dann gelten die folgenden oberen Schranken:*

$$A(w) \leq 2n \cdot \sum_{\ell=1}^{L} |\psi(\ell)| + n^2(p_{\max} - 1)$$

und die im allgemeinen bessere obere Schranke

$$A(w) \leq 2n \cdot C(w) + n^2 \cdot \log(p_{\max}) + n^2 \cdot \min(p_{\max}, d_{\max}) + 2n \cdot (\sum_{\ell=1}^{L} \min(|\psi(\ell)|, d_{\max})).$$

Für einen Beweis des Satzes und eine weitere Diskussion hinsichtlich Komplexitätsfragen verweisen wir auf den nächsten Abschnitt. Es sei bemerkt, daß für die meisten Gruppen diese oberen Schranken sehr grob sind. Für viele Gruppen sind z. B. der maximale Primfaktor $p_{\max}$ und der maximale Grad $d_{\max}$ vernachlässigbar klein im Verhältnis zu $|G|$ und $C(w)$. In diesem Fall ist die Laufzeit im wesentlichen linear in der Komplexität $C(w)$ und der Länge n der Kompositionsreihe $\mathcal{T}$. Als Spezialfall von Satz 4.1.2 ergibt sich das folgende Korollar.

Korollar 4.1.3 *Sei G eine pc-präsentierte p-Gruppe der Ordnung p^n mit den Voraussetzungen von Satz 4.1.2. Dann gilt:*

(1) *Die Normalform des Produkts von zwei Wörtern in Normalform von G kann mit $5 \cdot p \cdot n^2$ Additionen in $\mathbb{Z}_e$ berechnet werden.*

(2) *Die Normalform der Inversen eines in Normalform gegebenen Gruppenelements in G kann mit $3 \cdot p \cdot n^2$ Additionen in $\mathbb{Z}_e$ bestimmt werden.*

Beweis: Sind a, b Wörter in Normalform von G, so können alle Exponenten von $w = a \cdot b$ durch $(p-1)$ und L durch $2 \cdot n$ abgeschätzt werden. Damit folgt die Behauptung (1) unmittelbar aus der ersten Abschätzung von Satz 4.1.2. Ist $g = g_n^{e_n} \cdot \ldots \cdot g_1^{e_1}$ in Normalform gegeben, so gilt für die Inverse $g^{-1} = g_1^{-e_1} \cdot \ldots \cdot g_n^{-e_n}$. Normalisierung dieses Wortes kann nach der ersten Abschätzung von Satz 4.1.2 mit dem in (2) behaupteten Aufwand durchgeführt werden. □

Natürlich kann der WN-Algorithmus weiter verbessert und verallgemeinert werden. Zum einen werden nicht alle irreduziblen Darstellungen von $\mathrm{Irr}(G, \mathcal{T})$ verwendet, so daß man sich in der Vorverarbeitungsphase auf die Berechnung der Darstellungen, die für die Normalisierung notwendig sind, beschränken kann. Zum anderen können mit derselben Idee Wörter normalisiert werden, die in einer wesentlich komplexeren Form gegeben sind, z. B. Ausdrücke in den Erzeugern, die durch ein Straight-Line-Programm beschrieben werden können.

4.2 Analyse des WN-Algorithmus

In diesem Abschnitt analysieren wir den WN-Algorithmus und beweisen Satz 4.1.2. Zur Vorbereitung untersuchen wir, wie man schnell einzelne Einträge e-monomialer Matrizen berechnen kann. Bei unserer Analyse betrachten wir nur Additionen in $\mathbb{Z}_e$, da sich diese als die wesentlichen Operationen herausstellen (siehe auch Bemerkung 4.2.4).

Lemma 4.2.1 *Sei $A = \alpha\,\mathrm{diag}(\omega^{a_1}, \ldots, \omega^{a_N}) \in \mathrm{GL}(N, \mathbb{C})$ eine e-monomiale Matrix der Dimension $N \in \mathbb{N}$, wobei die Permutation $\alpha \in S_N$ als Permutationsmatrix interpretiert wird,*

die in der m-ten Spalte an Position $\alpha(m)$ eine Eins stehen hat. Darüber hinaus definieren wir $A_m := (\alpha(m), a_m)$. Sei $B = \beta\,\mathrm{diag}(\omega^{b_1}, \ldots, \omega^{b_N})$ analog definiert, dann gilt:

$$\begin{aligned}(A^{-1})_m &= (\alpha^{-1}(m),\ (-a_{(\alpha^{-1}(m)}) \bmod e),\\ (AB)_m &= (\alpha(\beta(m)),\ (a_{\beta(m)} + b_m) \bmod e),\\ (A^r)_m &= (\alpha^r(m),\ (a_{\alpha^{r-1}(m)} + \ldots + a_{\alpha(m)} + a_m) \bmod e).\end{aligned}$$

Ein Beweis dieser Gleichungen ergibt sich durch einfaches Nachrechnen. Die Formel für den m-ten Eintrag des Produkts zweier Matrizen läßt sich leicht auf L Matrizen für beliebiges $L \in \mathbb{N}$ verallgemeinern. Nach der Formel von Lemma 4.2.1 ist es in Hinblick auf die Permutationen effizienter, die Multiplikation von rechts nach links durchzuführen. Es gilt das folgende Lemma.

Lemma 4.2.2 *Seien $A_1, \ldots, A_L \in \mathrm{GL}(N, \mathbb{C})$ e-monomiale Matrizen und sei $m \in \{1, \ldots, N\}$. Dann läßt sich $(A_1 \cdot \ldots \cdot A_{L-1} \cdot A_L)_m$ mit $L - 1$ Additionen in $\mathbb{Z}_e$ bestimmen.*

Das nächste Lemma besagt, wie man für große $r \gg N$ mit Hilfe der binären Methode effizient den Eintrag $(A^r)_m$ berechnen kann.

Lemma 4.2.3 *Sei $A = \alpha\,\mathrm{diag}(\omega^{a_1}, \ldots, \omega^{a_N})$ wie in Lemma 4.2.1 und $r \in \mathbb{N}$. Es liege $m \in \{1, \ldots, N\}$ in einem Zykel der Länge M, $1 \le M \le N$, von α. Dann kann $(A^r)_m$ mit maximal $r - 1$ und für $r \ge M$ mit maximal $2\lfloor \log(r \operatorname{div} M) \rfloor + M$ Additionen in $\mathbb{Z}_e$ berechnet werden.*

Beweis: Offensichtlich kann die Summe $(a_{\alpha^{r-1}(m)} + \ldots + a_{\alpha(m)} + a_m) \bmod e$ mit $r - 1$ Additionen in $\mathbb{Z}_e$ berechnet werden, und die Schranke $r - 1$ zur Berechnung von $(A^r)_m$ ist trivial (siehe auch Lemma 4.2.2). Für $r \ge M$ läßt sich $(A^r)_m$ folgendermaßen berechnen: Es sei $s := r \operatorname{div} M$ und $t := r \bmod M$.

(i) Berechne $c_1 := ((a_{\alpha^{t-1}(m)} + \ldots + a_{\alpha(m)} + a_m) \bmod e)$ und $c_2 := ((a_{\alpha^{M-1}(m)} + \ldots + a_{\alpha(m)} + a_m) \bmod e)$ mit $M - 1$ Additionen in $\mathbb{Z}_e$.

(ii) Berechne $c_3 := ((c_2 + c_2 + \ldots + c_2) \bmod e)$ (insgesamt s Summanden) unter Verwendung der binären Methode mit maximal $2\lfloor \log s \rfloor$ Additionen in $\mathbb{Z}_e$.

(iii) Berechne $(A^r)_m = ((c_1 + c_3) \bmod e)$ mit einer weiteren Addition in $\mathbb{Z}_e$.

Daraus folgt unmittelbar die Behauptung. In (ii) könnte man natürlich $c_3 = ((s \cdot c_2) \bmod e)$ durch eine Multiplikation in $\mathbb{Z}_e$ berechnen. Diese wurde jedoch aus den in Bemerkung 4.2.4 angeführten Gründen auf Additionen in $\mathbb{Z}_e$ zurückgeführt. □

Wir kommen nun zur eigentlichen Analyse des WN-Algorithmus. Mit den Bezeichnungen aus Abschnitt 4.1 muß im i-ten Schritt, $1 \le i \le n$, des WN-Algorithmus der m-te Eintrag des Produktes

$$D(w_i) = D(g_{i+1})^{-e_{i+1}} \cdot \ldots \cdot D(g_n)^{-e_n} \cdot D(g_{\varphi(1)})^{\psi(1)} \cdot \ldots \cdot D(g_{\varphi(L)})^{\psi(L)} \tag{4.5}$$

berechnet werden, wobei nur die Daten der Vorverarbeitungsphase, also die DFT-Datenstruktur von G und deren inverse Matrizen, zur Verfügung stehen.

- Da $D \downarrow G_{i-1}$ trivial ist, verursachen Faktoren der Form $D(g_j)^r$, $r \in \mathbb{Z}$, $1 \leq j < i$, keine Kosten.

- Die Matrizen $D(g_j)$, $i \leq j < n$, sind in der DFT-Datenstruktur nicht unmittelbar gegeben. $D(g_j)$ ist eine Block-Diagonalmatrix mit Blöcken $F(g_j)$ für geeignete $F \in \mathrm{Irr}(G_j, \mathcal{T}_j)$, wobei die Matrizen $F(g_j)$ in der DFT-Datenstruktur bis auf einen konstanten Faktor $c \in \mathbb{C}$ (einer e-ten Einheitswurzel, siehe Abschnitt 3.2) gegeben sind. Die Berechnung eines nicht-trivialen Eintrags eines Faktors der Form $D(g_j)^r$, $r \in \mathbb{N}$, $i \leq j < n$, kann ohne Zusatzkosten auf die Berechnung eines nicht-trivialen Eintrages von $F(g_j)$ für ein geeignetes $F \in \mathrm{Irr}(G_j, \mathcal{T}_j)$ zurückgeführt werden. (Das geeignete F kann aus dem $\mathcal{T}$-Charaktergraphen abgelesen werde, welcher Bestandteil der DFT-Datenstruktur ist. Siehe auch Abbildung 4.1.) Aus Lemma 4.2.3 unter Berücksichtigung des zusätzlichen Faktors c folgt, daß ein nicht-trivialer Eintrag von $D(g_j)^r$ mit

$$\min\left(2r-1,\, 2\lfloor \log(r \text{ div } M)\rfloor + \min(2r-1, 2M-1)\right) \tag{4.6}$$

  Additionen in $\mathbb{Z}_e$ mit einem $M \leq \deg(F)$ berechnet werden kann.

- Für Faktoren der Form $D(g_j)^r$ mit negativem $r \in \mathbb{Z}_{<0}$, werden die inversen Matrizen, die in der Vorverarbeitungsphase berechnet wurden, verwendet. Dabei kann man wie im vorherigen Punkt verfahren.

Wir beweisen zunächst die erste Schranke für $A(w)$ von Satz 4.1.2. Bezeichne $A(i)$ die Anzahl der Additionen in $\mathbb{Z}_e$ zur Berechnung von $D(w_i)_m$ im i-ten Schritt. Das Produkt in (4.5) hat $n - i + L$ Faktoren. Aus Lemma 4.2.2 und der ersten Schranke in (4.6) ergibt sich damit die folgende Abschätzung:

$$A(i) \;\leq\; n - i + L - 1 + \sum_{\ell=i+1}^{n} (2e_\ell - 1) + \sum_{\ell=1}^{L} (2|\psi(\ell)| - 1).$$

Summation über alle Schritte $i = 1, \ldots, n$ ergibt dann eine obere Schranke für die Anzahl $A(w)$ der Additionen in $\mathbb{Z}_e$, die der WN-Algorithmus benötigt. Mit $p_{\max} := \max_{1 \leq i \leq n}(p_i)$ und $e_\ell \leq p_{\max} - 1$ folgt:

$$\begin{aligned} A(w) &= \sum_{i=1}^{n} A(i) \\ &\leq \binom{n}{2} + n \cdot L - n + \binom{n}{2}(2(p_{\max}-1)-1) + n \cdot \sum_{\ell=1}^{L}(2|\psi(\ell)|-1) \\ &\leq 2n \cdot \sum_{\ell=1}^{L} |\psi(\ell)| + n^2(p_{\max}-1). \end{aligned} \tag{4.7}$$

Wir kommen nun zum Beweis der zweiten Schranke für $A(w)$ von Satz 4.1.2. Hierfür verwenden wir Lemma 4.2.2 und die zweite Schranke in (4.6). Dort schätzen wir zusätzlich alle Zykellängen M durch $d_{\max} := \max_{D \in \mathrm{Irr}(G,\mathcal{T})}(\deg(D))$ und $\log(|r| \operatorname{div} M)$ durch $\log|r|$ ab:

$$\begin{aligned} A(i) &\leq n - i + L - 1 + \sum_{\ell=i+1}^{n} (2\lfloor \log e_\ell \rfloor + \min(2e_\ell - 1, 2d_{\max} - 1)) \\ &\quad + \sum_{\ell=1}^{L} (2\lfloor \log|\psi(\ell)| \rfloor + \min(2|\psi(\ell)| - 1, 2d_{\max} - 1)). \end{aligned}$$

Schätzen wir wieder $e_\ell \leq p_{\max} - 1$ ab und benutzen die Definition (4.2) für die Komplexität $C(w)$ des Wortes w, also $\sum_{\ell=1}^{L} \lfloor \log(|\psi(\ell)|) \rfloor = C(w) - L$, dann folgt

$$\begin{aligned} A(i) &\leq n - i + L - 1 + \sum_{\ell=i+1}^{n} (2\lfloor \log p_{\max} - 1 \rfloor + \min(2p_{\max} - 3, 2d_{\max} - 1)) \\ &\quad + 2 \cdot C(w) - 2 \cdot L + \sum_{\ell=1}^{L} (\min(2|\psi(\ell)| - 1, 2d_{\max} - 1)) \\ &\leq n - i + L - 1 + 2 \cdot (n - i)\,(\lfloor \log p_{\max} - 1 \rfloor + \min(p_{\max} - 1, d_{\max})) - (n - i) \\ &\quad + 2 \cdot C(w) - 2 \cdot L + 2 \cdot \sum_{\ell=1}^{L} (\min(|\psi(\ell)|, d_{\max})) \\ &\leq 2 \cdot (n - i)(\log(p_{\max} - 1) + \min(p_{\max} - 1, d_{\max})) \\ &\quad + 2 \cdot C(w) - L + 2 \cdot \sum_{\ell=1}^{L} \min(|\psi(\ell)|, d_{\max}). \end{aligned}$$

Damit ergibt sich für den Gesamtaufwand $A(w)$ folgende Abschätzung:

$$\begin{aligned} A(w) &= \sum_{i=1}^{n} A(i) \\ &\leq 2 \cdot \binom{n}{2} \cdot (\log(p_{\max} - 1) + \min(p_{\max} - 1, d_{\max})) \\ &\quad + 2n \cdot C(w) - n \cdot L + 2n \cdot \sum_{\ell=1}^{L} \min(|\psi(\ell)|, d_{\max}) \\ &\leq 2n \cdot C(w) + n^2 \cdot \log(p_{\max}) + n^2 \cdot \min(p_{\max}, d_{\max}) \\ &\quad + 2n \cdot (\sum_{\ell=1}^{L} \min(|\psi(\ell)|, d_{\max})). \end{aligned} \tag{4.8}$$

Mit (4.7) und (4.8) ist die Behauptung von Satz 4.1.2 vollständig bewiesen.

Bemerkung 4.2.4 Beim WN-Algorithmus treten neben den Additionen in $\mathbb{Z}_e$ weitere Operationen auf, die in unserer Komplexitätsanalyse nicht berücksichtigt wurden.

(1) Es müssen u. a. Werte von Produkten, Inversen und Potenzen von Permutationen der Form $\alpha(\beta(m))$, $\alpha^{-1}(m)$ oder $\alpha^r(m)$ bestimmt werden. Dies geschieht durch wiederholte „Table-Look-Ups“ (TLUs) in den durch die DFT-Datenstruktur gegebenen Daten. Man kann analog zur durchgeführten Analyse zeigen, daß die Gesamtzahl aller TLUs im WN-Algorithmus dieselbe Größenordnung haben wie die Anzahl der Additionen in $\mathbb{Z}_e$. Da jedoch TLUs wesentlich billiger sind als Additionen in $\mathbb{Z}_e$, wirken sich diese auf die Laufzeit des WN-Algorithmus praktisch nicht aus.

(2) Ähnliches kann über andere Hilfsoperationen gesagt werden, die im WN-Algorithmus auftreten. Z. B. müssen Divisionen mit Rest durchgeführt werden, um die Indizes m der Einträge von darstellenden Matrizen zu berechnen. Solche Operationen liegen in der Bitkomplexität bei einer Größenordnung wie bei der für die Anzahl der Additionen in $\mathbb{Z}_e$, und man kann sie daher ebenfalls vernachlässigen.

(3) Es soll betont werden, daß beim WN-Algorithmus nur Additionen aber keine Multiplikationen in $\mathbb{Z}_e$ verwendet werden (siehe z. B. Beweis von Lemma 4.2.3). Der Grund hierfür liegt darin, daß bei Multiplikationen modulo e die Zwischenergebisse nur mit der oberen Schranke e^2 abgeschätzt werden können. Beim Rechnen mit 32-Bit-Wörtern kann es schon für Gruppenordnungen der Größe $|G| > 2^{16}$ zu Überläufen kommen. Dies ist selbst für praktische Anwendungen nicht akzeptabel. Da wir nur Additionen modulo e verwenden, gelten im Hinblick auf die Implementierung analoge Schranken für die Gruppenordnungen wie in Abschnitt 3.4.

Es stellt also keine wesentliche Einschränkung dar, daß bei obiger Komplexitätsanalyse nur Additionen in $\mathbb{Z}_e$ betrachtet wurden. Dies bestätigen auch die Laufzeiten in konkreten Beispielen (siehe Abschnitt 4.3).

4.3 Implementierung und Laufzeiten

Unter denselben Bedingungen wie in Abschnitt 3.4.2 wurde der WN-Algorithmus implementiert und getestet. Wir fassen in diesem Abschnitt einige der durchgeführten Versuche zusammen und diskutieren die Laufzeiten. Um mögliche Schwankungen, die bei der Bestimmung der Laufzeiten auftreten, auszugleichen, wurden alle Berechnungen zehnmal durchgeführt und dann über die entsprechenden Laufzeiten gemittelt. In diesem Abschnitt sind alle Laufzeiten in Millisekunden angegeben.

Tabelle 4.1 illustriert das Laufzeitverhalten des WN-Algorithmus bei festen Wörtern in Abhängigkeit von der Gruppengröße $|G|$ und Länge n der Kompositionsreihe $\mathcal{T}$. Hierbei wurden als Gruppen m-fache direkte Produkte der symmetrischen Gruppen $(S_3)^m$ für wachsendes m herangezogen. In der vierten Spalte steht die Laufzeit der Vorverarbeitungsphase (DFT-Berechnung). In der fünften bis siebten Zeile sind die Laufzeiten $t(w_i)$ angeben, die zur Wortnormalisierung der Wörter w_i, $i = 1, 2, 3$, benötigt wurden. In den letzten beiden Zeilen sind jeweils die Länge $L(w_i)$ bzw. die Komplexität $C(w_i)$ angegeben (siehe (4.2)). Die

G	$\|G\|$	n	t(DFT)	$t(w_1)$	$t(w_2)$	$t(w_3)$
(S_3)	6	2	< 1	17	133	951
$(S_3)^2$	36	4	< 1	30	218	1808
$(S_3)^3$	216	6	< 1	55	301	2846
$(S_3)^4$	1296	8	1	56	436	4040
$(S_3)^5$	7776	10	17	92	558	5454
$(S_3)^6$	46656	12	82	103	692	6897
$(S_3)^7$	279936	14	281	128	856	8499
$(S_3)^8$	679616	16	1370	100	1035	10283
$(S_3)^9$	10077696	18	5124	120	1225	12183
$(S_3)^{10}$	60466176	20	22186	252	1464	14386
$L(w)$				10^4	10^5	10^6
$C(w)$				$2.2 \cdot 10^4$	$2.2 \cdot 10^5$	$2.2 \cdot 10^6$

Tabelle 4.1: Laufzeiten in Abhängigkeit von n.

Wörter w_i wurden dabei durch Zufallsfunktionen φ und ψ (siehe (4.1)) erzeugt, die jeweils gleichverteilt in $[1:n]$ bzw. $[1:6]$ (6 ist der Exponent für alle Gruppen $G = (S_3)^m$)) sind. Betrachtet man die Spalten der Tabelle, so erkennt man, daß sich der WN-Algorithmus in den angegebenen Beispielen im wesentlichen linear in n verhält. Aus den Zeilen der Tabelle ergibt sich, daß die Laufzeiten in etwa linear in $C(w)$ sind. Damit kann das Laufzeitverhalten der Implementierung des WN-Algorithmus in etwa durch $n \cdot C(w)$ beschrieben werden. Dies ist im Einklang mit der zweiten theoretischen Laufzeitabschätzung von Satz 4.1.2, wenn man bedenkt, daß die restlichen Summanden für kleine $p_{\max}$ und $d_{\max}$ vernachlässigbar und von $C(w)$ unabhängig sind.

Die nächste Tabelle 4.2 untermauert diese Aussagen. Die Laufzeiten des WN-Algorithmus sind hier bei festem $|G|$ und n im wesentlichen unabhängig von der restlichen Struktur der Gruppe, d. h. die Anzahl h der Konjugationsklassen oder die Grade der Darstellungen fallen kaum ins Gewicht. Bei den Gruppen handelt es sich um die in Abschnitt 3.4.1 beschriebenen, durch die pc-Präsentationen unserer Bibliothek erzeugten Gruppen. Analog zum vorherigen Versuch wurden dabei Worte w_i mit in geeigneten Bereichen gleichverteilten φ und ψ verwendet.

G	$\|G\|$	n	h	t(DFT)	$t(w_1)$	$t(w_2)$	$t(w_3)$
$\mathrm{Lib}_{44100}(6)$	44100	8	44100	240	125	891	8975
$\mathrm{Lib}_{44100}(1)$	44100	8	22050	441	124	895	8978
$\mathrm{Lib}_{44100}(21)$	44100	8	13230	436	123	893	8981
$\mathrm{Lib}_{44100}(19)$	44100	8	3675	49	122	901	9062
$L(w)$					10^4	10^5	10^6
$C(w)$					$12.4 \cdot 10^4$	$14.3 \cdot 10^5$	$14.5 \cdot 10^6$

Tabelle 4.2: Laufzeiten bei fester Gruppengröße.

Wie schon die Analyse in Abschnitt 4.2 des WN-Algorithmus zeigte, ist für die Laufzeit des WN-Algorithmus nicht die Gruppenordnung $|G|$, sondern nur die Länge n der Kompositionsreihe, die auch als Höhe des Charaktergraphen der Gruppe beschrieben werden kann, von Relevanz. Dieses Resultat wird auch durch die in Tabelle 4.3 angegebenen Laufzeiten untermauert. So sind die Laufzeiten für die Gruppe $G = \mathrm{Syl}_2(S_{16})$ $(n = 16)$ in etwa dreimal so

groß wie für die Gruppe $G = \mathrm{Lib}_{42875}(1)$ $(n = 6)$ oder doppelt so groß wie für $G = \mathrm{Lib}_{7560}(1)$ $(n = 8)$. In dieser Versuchsreihe wurden Wörter verwendet, bei denen alle Exponenten identisch eins sind, also $\psi(\ell) = 1$ für alle $1 \leq \ell \leq L(w)$ und damit $C(w) = L(w)$ gilt.

G	$\lvert G\rvert$	n	$t(\mathrm{DFT})$	$t(w_1)$	$t(w_2)$	$t(w_3)$	$t(w_4)$	$t(w_5)$
$\mathrm{Lib}_{1024}(1)$	1024	10	4	7	76	481	4670	46690
$\mathrm{Lib}_{2187}(1)$	2187	7	2	2	46	320	3015	29065
$\mathrm{Lib}_{7560}(1)$	7560	8	119	4	50	371	3810	36336
$\mathrm{Lib}_{7560}(54)$	7560	8	6	3	54	373	3439	34420
$\mathrm{Lib}_{7776}(1)$	7776	10	21	8	72	470	4609	45380
$\mathrm{Lib}_{15625}(35)$	15625	6	6	2	31	274	2470	23411
$\mathrm{Lib}_{16000}(1)$	16000	10	77	6	69	472	4521	45229
$\mathrm{Lib}_{22287}(1)$	22287	4	96	1	25	196	1695	15905
$\mathrm{Lib}_{23940}(1)$	23940	7	113	5	41	317	2975	28340
$\mathrm{Syl}_2(S_{16})$	32768	15	146	12	120	830	8299	82611
$\mathrm{Lib}_{42875}(1)$	42875	6	55	3	38	274	2570	24335
$\mathrm{Lib}_{179894}(3)$	179894	5	260	1	31	237	2185	20380
$L(w) = C(w)$				10^3	10^4	10^5	10^6	10^7

Tabelle 4.3: Laufzeiten für verschiedene Gruppen.

Die Ergebnisse der Implementierung des WN-Algorithmus wurden durch Vergleich mit den durch eine GAP-Routine [31] berechneten Normalformen verifiziert. Hierzu wurden unter Verwendung der GAP-Klasse `PcGroupFpGroup` die entsprechenden polyzyklischen Gruppen erzeugt. Die Normalformen der zu normalisierenden Wörter w_i, welche als Listen der Werte $\varphi(\ell)$ und $\psi(\ell)$ vorlagen, wurde dann mit der GAP-internen Routine, die auf den in [55] beschriebenen „Collection"-Verfahren beruht, berechnet. Für eine Diskussion dieser Verfahren verweisen wir auch auf [39] und [63]. In Tabelle 4.4 findet man einen Vergleich der Laufzeiten des WN-Algorithmus und der GAP-Rountine. In den durch mit WN bzw. mit GAP gekennzeichneten Zeilen findet man jeweils die Laufzeiten zur Wortnormalisierung bez. des WN-Algorithmus bzw. der GAP-Routine von Wörtern w_i verschiedener Komplexität. Uns geht es im folgenden weniger um die absoluten Laufzeiten, sondern um das relative Laufzeitverhalten. Wir fassen einige Beobachtungen zusammen:

- Die GAP-Routine hängt, ähnlich wie der WN-Algorithmus, in etwa linear von der Komplexität $C(w)$ ab.

- Im Unterschied zum WN-Algorithmus scheint die GAP-Routine weder wesentlich von der Gruppengröße noch von der Länge n der Hauptreihe abzuhängen. Dies wird durch Laufzeiten bezüglich der ersten drei Gruppen S_3, G_{128} und $\mathrm{Syl}_2(S_{16})$ illustriert.

- Entscheidend für die in GAP verwendeten „Collection"-Verfahren ist die Komplexität der die Gruppe definierenden pc-Präsentationen, also die Anzahl und Größe der nichttrivialen Koeffzienten $a_{i,\ell}$ und $b_{ij,\ell}$ in der pc-Präsentation (siehe (2.8)). Dies wird durch die Laufzeiten der Gruppen $\mathrm{Lib}_{44100}(19)$ und G_{44100} illustriert. Die pc-Präsentation der Gruppe G_{44100} ist trivial, d. h. alle Koeffizienten $a_{i,\ell}$ und $b_{ij,\ell}$ sind Null und

$$G_{44100} \simeq (C_2)^2 \times (C_3)^2 \times (C_5)^2 \times (C_7)^2.$$

Die Laufzeiten bei der GAP-Routine ist für diese Gruppe merklich schneller als für die Gruppe $\mathrm{Lib}_{44100}(19)$, bei der es einige nicht-triviale Relationen in der pc-Präsentation gibt. Im Unterschied hierzu ist jedoch das Laufzeitverhalten des WN-Algorithmus von diesen Größen unabhängig.

G	$\lvert G\rvert$	n		$t(w_1)$	$t(w_2)$	$t(w_3)$	$t(w_4)$	$t(w_5)$	$t(w_6)$
S_3	6	2	WN	< 1	10	114	2	35	304
			GAP	26	218	1755	54	382	4096
G_{128}	128	7	WN	3	51	310	7	106	839
			GAP	25	216	1704	59	402	4445
$\mathrm{Syl}_2(S_{16})$	32768	15	WN	11	121	830	22	191	1954
			GAP	35	226	1954	61	433	4820
$\mathrm{Lib}_{44100}(19)$	44100	8	WN	6	45	385	10	124	963
			GAP	30	241	2064	144	1349	17174
G_{44100}	44100	8	WN	3	49	398	8	124	953
			GAP	22	206	1660	45	411	4437
$L(w)$				10^3	10^4	10^5	10^3	10^4	10^5
$C(w)$				10^3	10^4	10^5	$9.0 \cdot 10^3$	$12.4 \cdot 10^4$	$15.7 \cdot 10^5$

Tabelle 4.4: Laufzeiten im Vergleich zu GAP.

Zusammenfassend kann man sagen, daß sich die beiden Verfahren zur Wortnormalisierung grundsätzlich in ihrer Strategie unterscheiden, was sich auch im völlig unterschiedlichen Laufzeitverhalten widerspiegelt. Während der WN-Algorithmus entscheidend von n abhängt, aber nicht wesentlich von der Beschaffenheit der pc-Präsentation, hat man bei der GAP-Routine eine umgekehrte Abhängigkeit. Es sei erinnert, daß der WN-Algorithmus für die DFT-Datenstruktur in der Gruppenordnung $|G|$ linear viel Speicherplatz benötigt, im Unterschied zu den in [55] beschriebenen „Collection"-Verfahren, die über den Speicherplatz für die pc-Präsentation im wesentlichen keinen weiteren Speicherplatz benötigen.

4.4 Multivariate Polynomdivision via Wortnormalisierung

Im Fall einer endlichen abelschen Gruppen G kann die Gruppenalgebra $\mathbb{C}G$ mit dem Quotienten eines Polynomrings modulo eines geeigneten Gitterideals I_G identifiziert werden. Wählt man eine geeignete monomiale Ordnung, so definiert die pc-Präsentation von G eine reduzierte Gröbnerbasis von I_G. Polynomdivision modulo I_G kann dann via Wortnormalisierung mit Hilfe des WN-Algorithmus durchgeführt werden (PD-Algorithmus). In Abschnitt 4.4.1 fassen wir die Grundlagen zu Gröbnerbasen in Polynomringen zusammen und stellen in Abschnitt 4.4.2 den Zusammenhang zu Gruppenalgebren abelscher Gruppen her. In Abschnitt 4.4.3 beschreiben und analysieren wir dann den PD-Algorithmus zur multivariaten Polynomdivision.

4.4.1 Grundlagen zu Gröbnerbasen

In diesem Abschnitt werden die benötigten Definitionen und Eigenschaften von Gröbnerbasen zusammengefaßt. Für eine Einführung in das Gebiet der algorithmischen algebraischen Geometrie verweisen wir auf [19].

Sei $\mathbb{K}$ ein beliebiger Körper und $\mathbb{K}[X] := \mathbb{K}[X_n, \ldots, X_1]$ der *Polynomring* über $\mathbb{K}$ in den Unbestimmten $X = (X_n, \ldots, X_1)$. Ein *Polynom* $f \in \mathbb{K}[X]$ ist dann in Multiindexschreibweise eine endliche Summe der Form $f = \sum_\alpha c_\alpha X^\alpha$ mit Koeffizienten $c_\alpha \in \mathbb{K}$. Der *totale Grad* von $f \neq 0$, im folgenden mit $\deg(f)$ bezeichnet, ist das Maximum $|\alpha| := \alpha_n + \ldots + \alpha_1$ über alle Koeffizienten $c_\alpha \neq 0$, $\alpha = (\alpha_n, \ldots, \alpha_1)$. Jedes *Monom* $X^\alpha = X_n^{\alpha_n} \cdot \ldots \cdot X_1^{\alpha_1} \in \mathbb{K}[X]$ kann mit dem n-Tupel der Exponenten $\alpha \in \mathbb{Z}_{\geq 0}^n$ identifiziert werden.

Definition 4.4.1 *Eine* monomiale Ordnung *auf* $\mathbb{K}[X]$ *ist eine totale Ordnung* $\succ$ *auf* $\mathbb{Z}_{\geq 0}^n$ *mit den folgenden Eigenschaften:*

(1) *Translationsinvarianz:* $\forall \alpha, \beta, \gamma \in \mathbb{Z}_{\geq 0}^n : \alpha \succ \beta \Rightarrow \alpha + \gamma \succ \beta + \gamma$.

(2) *Wohlordnung: Jede nicht-leere Teilmenge von* $\mathbb{Z}_{\geq 0}^n$ *hat ein kleinstes Element unter* $\succ$.

Zum Beispiel definiert die *lexikographische Ordnung*, die wir im folgenden auch mit $\succ_{\text{lex}}$ bezeichnen, eine monomiale Ordnung auf $\mathbb{Z}_{\geq 0}^n$. Dabei gilt $\alpha \succ_{\text{lex}} \beta$ genau dann, wenn der linkeste von Null verschiedene Eintrag der Differenz $\alpha - \beta \in \mathbb{Z}^n$ positiv ist. Es gilt also z. B. $(3, 2, 4) \succ_{\text{lex}} (3, 2, 1)$. Für die Monome schreiben wir ebenfalls $x^\alpha \succ_{\text{lex}} x^\beta$. In diesem Sinne gilt dann $x_n \succ_{\text{lex}} x_{n-1} \succ_{\text{lex}} \ldots \succ_{\text{lex}} x_1$.

Sei im folgenden $\succ$ eine beliebige, aber feste monomiale Ordnung. Der *Multigrad* von f ist definiert als $\text{multideg}(f) := \max(\alpha \in \mathbb{Z}_{\geq 0}^n : c_\alpha \neq 0)$, wobei das Maximum bez. $\succ$ zu nehmen ist. Hiermit definieren wir den *führenden Koeffizient* $\text{LC}(f) := c_{\text{multideg}(f)} \in \mathbb{K}$, das *führende Monom* $\text{LM}(f) := x^{\text{multideg}(f)}$ und den *führenden Term* $\text{LT}(f) := \text{LC}(f) \cdot \text{LM}(f)$.

Polynomdivision mit Rest kann im multivariaten Fall bezüglich einer monomialen Ordnung ähnlich durchgeführt werden wie im univariaten Fall. Wir fassen das Ergebnis im folgenden Satz zusammen.

Satz 4.4.2 (Divisionsalgorithmus in $\mathbb{K}[X_n, \ldots, X_1]$) *Sei* $\succ$ *eine monomiale Ordnung auf* $\mathbb{Z}_{\geq 0}^n$ *und* $F = (f_1, \ldots, f_s)$ *ein geordnetes* s*-Tupel von Polynomen in* $\mathbb{K}[X] = \mathbb{K}[X_n, \ldots, X_1]$. *Dann kann jedes* $f \in \mathbb{K}[X]$ *als*

$$f = c_1 f_1 + \ldots + c_s f_s + r$$

mit $c_i, r \in \mathbb{K}[X]$ *geschrieben werden, wobei* $r = 0$ *gilt oder* r *eine* $\mathbb{K}$*-Linearkombination von Monomen ist, welche nicht durch* $\text{LT}(f_1), \ldots, \text{LT}(f_s)$ *teilbar sind. Das Polyonom* r *heißt* Rest *von* f *bezüglich der Division durch* F. *Im Fall* $c_i f_i \neq 0$ *gilt darüber hinaus* $\text{multideg}(f) \succeq \text{multideg}(c_i f_i)$.

Im allgemeinen ist im multivariaten Fall der Rest r durch die Eigenschaft, daß kein Term durch $\text{LT}(f_1), \ldots, \text{LT}(f_s)$ teilbar sein soll, nicht eindeutig bestimmt. Gröbnerbasen sind nun gerade dadurch charakterisiert, daß der Divisionsalgorithmus bezüglich dieser Basen zu einem eindeutig bestimmten Rest führt. Wir präzisieren diese Aussage. Sei $I \subset \mathbb{K}[X_n, \ldots, X_1]$ ein beliebiges von Null verschiedenes Ideal, dann ist die Menge der führenden Terme von I durch $\text{LT}(I) := \{\text{LT}(f) : f \in I\}$ definiert. Das von $\text{LT}(I)$ erzeugte *Ideal der führenden Terme* $\langle \text{LT}(I) \rangle$ wird von Monomen erzeugt und ist damit ein sogenanntes *monomiales Ideal*. Das *Lemma von Dickson* besagt, daß jedes monomiale Ideal von einer endlichen Anzahl von Monomen erzeugt wird (siehe [19], S. 69).

Definition 4.4.3 *Fixiere eine monomiale Ordnung. Eine endliche Menge* $\mathcal{G} = \{g_1, \ldots, g_t\}$ *eines Ideals* $I \subset \mathbb{K}[X_n, \ldots, X_1]$, $I \neq \{0\}$, *heißt* Gröbnerbasis, *falls*

$$\langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle = \langle \mathrm{LT}(I) \rangle$$

gilt. Eine reduzierte Gröbnerbasis $\mathcal{G}$ *ist eine Gröbnerbasis mit den Zusatzeigenschaften, daß für alle* $p \in \mathcal{G}$ *erstens* $\mathrm{LC}(p) = 1$ *gilt und zweitens kein Monom von* p *in* $\langle \mathrm{LT}(\mathcal{G} - \{p\}) \rangle$ *liegt.*

Aus dem Hilbertschen Basissatz und dem Buchberger-Algorithmus folgt, daß jedes Ideal I eine eindeutig bestimmte reduzierte Gröbnerbasis hat (siehe [19], S. 90). Es soll betont werden, daß diese jedoch von der fest gewählten monomialen Ordnung abhängt.

Für eine Gröbnerbasis $\mathcal{G} = \{g_1, \ldots, g_t\}$ gibt es für jedes $f \in \mathbb{K}[X]$ eindeutig bestimmte Polynome $g, r \in \mathbb{K}[X]$ mit $f = g + r$, so daß $g \in I$ und kein Term von r durch $\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t)$ teilbar ist. Im folgenden wird der eindeutig bestimmte Rest r auch mit $\bar{f}^{\mathcal{G}}$ bezeichnet. Mit anderen Worten, jedes $f \in \mathbb{K}[X]$ ist kongruent modulo I zu dem eindeutig bestimmten Polynom $\bar{f}^{\mathcal{G}}$, welches eine $\mathbb{K}$-Linearkombination von Monomen im Komplement von $\langle \mathrm{LT}(I) \rangle$ ist. Man sieht leicht, daß die Menge $\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(I) \rangle\}$ dieser Monome, die wir im folgenden auch als die *Standardmonome* bezeichnen, linear unabhängig ist. Hieraus folgt leicht folgender für uns wichtige Satz:

Satz 4.4.4 *Sei* $I \subset \mathbb{K}[X_n, \ldots, X_1]$ *ein Ideal und* $\mathcal{G}$ *eine Gröbnerbasis bez. einer beliebigen monomialen Ordnung, dann induziert die Abbildung* $\mathbb{K}[X_n, \ldots, X_1] \ni f \mapsto \bar{f}^{\mathcal{G}}$ *einen* $\mathbb{K}$*-Vektorraumisomorphismus von* $\mathbb{K}[X_n, \ldots, X_1]/I$ *nach* $\mathrm{Span}(\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(I) \rangle\})$.

Korollar 4.4.5 *Sei* $I \subset \mathbb{K}[X_n, \ldots, X_1]$ *ein Ideal und* $\mathcal{G} = \{g_1, \ldots, g_t\} \subset I$ *eine Teilmenge mit der Eigenschaft* $\dim(\mathbb{K}[X]/I) = |\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle\}| < \infty$. *Dann ist* $\mathcal{G}$ *eine Gröbnerbasis von* I.

Beweis: Angenommen das Ideal $\langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle$ wäre echt in $\langle \mathrm{LT}(I) \rangle$ enthalten, dann wäre $|\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(I) \rangle\}| < |\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle\}|$. Hieraus würde aufgrund der Voraussetzung folgen, daß $\dim(\mathbb{K}[X]/I) > |\{X^\alpha : X^\alpha \notin \langle \mathrm{LT}(I) \rangle\}|$. Dies ist ein Widerspruch zu Satz 4.4.4. Also gilt $\langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_t) \rangle = \langle \mathrm{LT}(I) \rangle$ und damit die Behauptung. □

4.4.2 PC-Präsentationen abelscher Gruppen als Gröbnerbasen

In diesem Abschnitt stellen wir den Zusammenhang zwischen pc-Präsentationen abelscher Gruppen G und Gröbnerbasen bestimmter Ideale her. Sei G im folgenden durch die konsistente pc-Präsentation

$$G = \langle g_1, \ldots, g_n \mid g_i^{p_i} = u_i := g_{i-1}^{a_{i,i-1}} \cdot \ldots \cdot g_1^{a_{i,1}} \rangle \tag{4.9}$$

gegeben, wobei alle Kommutatorrelationen trivial sind. Dann kann ein beliebiges Wort w in den Erzeugern $g_1, \ldots, g_n$ durch Addition der Exponenten zu gleichen Erzeugern in der Form $w = g_n^{\alpha_n} \cdot \ldots g_1^{\alpha_1}$ mit $\alpha_i \in \mathbb{Z}$, $1 \leq i \leq n$, geschrieben werden. Im folgenden seien daher alle Wörter in dieser Form gegeben. Da G abelsch ist, definiert die auf einem Erzeugendensystem

definierte Abbildung $\varphi : \mathbb{C}[X_n, \ldots, X_1] \to \mathbb{C}G,\ X_i \mapsto g_i,\ i = 1, \ldots, n$, einen surjektiven Algebrenhomomorphismus. Sei $I_G := \ker(\varphi)$. Dann induziert φ den Algebrenisomorphismus

$$\bar{\varphi} : \mathbb{C}[X_n, \ldots, X_1]/I_G \xrightarrow{\simeq} \mathbb{C}G. \tag{4.10}$$

Dies impliziert umittelbar das folgende Korollar.

Korollar 4.4.6 *Aus der Äquivalenz zweier Wörter $w_1 = g_n^{\alpha_n} \cdot \ldots g_1^{\alpha_1} \sim w_2 = g_n^{\beta_n} \cdot \ldots g_1^{\beta_1}$ bez. der pc-Relationen folgt die Gleichheit der entsprechenden Monome modulo des Gitterideals I_G, d. h. $X_n^{\alpha_n} \cdot \ldots X_1^{\alpha_1} - X_n^{\beta_n} \cdot \ldots X_1^{\beta_1} \in I_G$.*

Im folgenden sei die monomiale Ordnung $\succ$ auf $\mathbb{Z}_{\geq 0}^n$ die lexikographische monomiale Ordnung $\succ_{\text{lex}}$ mit $X_n \succ_{\text{lex}} \ldots \succ_{\text{lex}} X_1$. Die in (4.9) angegebenen Gruppenrelationen definieren Polynome

$$f_i := X_i^{p_i} - X_{i-1}^{a_{i,i-1}} \cdot \ldots \cdot X_1^{a_{i,1}} \in \mathbb{C}[X] \tag{4.11}$$

für $1 \leq i \leq n$ mit $\text{LT}(f_i) = X_i^{p_i}$ bez. $\succ_{\text{lex}}$.

Satz 4.4.7 *Die Menge $\mathcal{G} := \{f_1, \ldots, f_n\}$ ist die reduzierte Gröbnerbasis des Ideals $I_G \subset \mathbb{C}[X_n, \ldots, X_1]$ bezüglich der lexikographischen Ordnung $\succ_{\text{lex}}$.*

Beweis: Offensichtlich gilt $\mathcal{G} = \{f_1, \ldots, f_n\} \subset I_G$ und

$$\{X^\alpha : X^\alpha \notin \langle \text{LT}(f_1), \ldots, \text{LT}(f_n) \rangle\} = \{X_\alpha : 0 \leq \alpha_i < p_i,\ 1 \leq i \leq n\}.$$

Wegen $\dim(\mathbb{C}[X]/I_G) \overset{(4.10)}{=} \dim(\mathbb{C}G) = |G| = p_1 \cdot \ldots \cdot p_n$ sind damit die Voraussetzungen von Korollar 4.4.5 erfüllt, aus welchem die Behauptung folgt. □

Damit entsprechen den konsistenten pc-Präsentationen endlicher abelscher Gruppen G die reduzierten Gröbnerbasen der Ideale I_G bezüglich $\succ_{\text{lex}}$.

Bemerkung 4.4.8 In additiver Schreibweise entspricht der endlichen Gruppe G dem Quotienten $\mathbb{Z}^n/\mathcal{L}$, wobei $\mathcal{L} \subset \mathbb{Z}^n$ das durch die Zeilenvektoren der Matrix

$$H := \begin{pmatrix} p_n & -a_{n,n-1} & -a_{n,n-2} & \ldots & -a_{n,2} & -a_{n,1} \\ & p_{n-1} & -a_{n-1,n-2} & \ldots & -a_{n-1,2} & -a_{n-1,1} \\ & & \ddots & & \vdots & \\ & & & & p_2 & -a_{2,1} \\ & & & & & p_1 \end{pmatrix}. \tag{4.12}$$

erzeugte Gitter vom endlichen Index $[\mathbb{Z}^n : \mathcal{L}]$ bezeichne. Das Ideal I_G wird daher auch als *Gitterideal* bezeichnet. Die obere Dreiecksmatrix H über $\mathbb{Z}$ ist dabei, bis auf eine andere Normierung der Elemente über der Hauptdiagonalen, die *Hermite-Normalform* eines beliebigen Erzeugendensystems des Gitters $\mathcal{L}$, deren Elemente als Zeilenvektoren einer Matrix interpretiert werden. Für weitere Einzelheiten bez. Gitter und einer gittertheoretischen Interpretation der Gröbnerbasen verweisen wir auf [55] bzw. [61].

4.4.3 PD-Algorithmus

In diesem Abschnitt wird der PD-Algorithmus zur multivariaten Polynomdivision bez. der im letzten Abschnitt beschriebenen Gitterideale vorgestellt. Dabei wird im PD-Algorithmus eine stark vereinfachte Version des WN-Algorithmus für den Fall abelscher Gruppen verwendet.

Mit der Notation von Abschnitt 4.4.1 bezeichne $f = \sum_\alpha c_\alpha X^\alpha \in \mathbb{C}[X]$ ein Polynom, $\mathcal{G} \subset \mathbb{C}[X]$ eine endliche Teilmenge und I das durch $\mathcal{G}$ erzeugte Ideal. Beim Standardalgorithmus zur multivariaten Polynomdivision bezüglich $\mathcal{G}$ bzw. I wird f durch sukzessive Subtraktion geeigneter Vielfacher von Elementen von $\mathcal{G}$ reduziert, bis das Ergebnis nicht weiter reduzierbar ist. Im Fall einer Gröbnerbasis $\mathcal{G}$ ist dann das Ergebnis die eindeutig bestimmte Normalform $\bar{f}^{\mathcal{G}}$. Sei z. B. $f = X^\alpha$ ein Monom und $g = X^\beta - X^\gamma \in \mathcal{G}$ ein Binom mit $X^\beta \succ X^\gamma$ und $X^\beta | X^\alpha$, so läßt sich f durch g reduzieren, und man erhält das Polynom $f - X^{\alpha-\beta} g = X^{\alpha-(\beta-\gamma)}$, welches dann weiter zu reduzieren ist. Ist also $|\alpha|$ sehr groß im Verhältnis zu $|\beta|$ und $|\gamma|$, so kann man eine Laufzeit des Standardalgorithmus erwarten, die linear in $|\alpha|$ ist. Ist mit anderen Worten der Multigrad des zu reduzierenden Polynoms f groß im Verhältnis zu den Multigraden der Polynome $g \in \mathcal{G}$, so kann es zu einer Laufzeit des Standardalgorithmus kommen, der linear vom Multigrad des zu reduzierenden Polynoms abhängt.

Für den Spezialfall, daß es sich beim Ideal I um ein Gitterideal I_G einer endlichen abelschen Gruppe G wie in Abschnitt 4.4.2 handelt, kann die Polynomdivision so durchgeführt werden, daß die Laufzeit nur logarithmisch vom Multigrad von f abhängt. Die Monome in $\mathbb{C}[X]$ können als Wörter in den Erzeugern der Gruppe interpretiert werden. Sei nun $f = \sum_\alpha c_\alpha X^\alpha \in \mathbb{C}[X]$ das zu reduzierende Polynom. Werden die Monome von f durch die den Normalformen in G entsprechende Monome ersetzt, dann folgt aus Korollar 4.4.6, daß das resultierende Polynom modulo I_G mit f übereinstimmt. Dies ist die Grundidee des PD-Algorithmus, der statt der „horizontalen“ Vorgehensweise des Standardalgorithmus eine „vertikale“ Vorgehensweise hat und damit auch parallelisiert werden kann. Wir präzisieren den PD-Algorithmus zur multivariaten Polynomdivision. Sei zunächst als monomiale Ordnung die lexikographische Ordnung $\succ_{\text{lex}}$ gewählt und bezeichne $\mathcal{G}_{\text{lex}}$ die zugehörige reduzierte Gröbnerbasis.

PD-Algorithmus zur multivariaten Polynomdivision bez. $\mathcal{G}_{\text{lex}}$:

(1) **Eingabe:** Polynom $f = \sum_\alpha c_\alpha X^\alpha \in \mathbb{C}[X]$.

(2) **Normalisierungsschritt:** Berechne die Normalformen aller Monome X^α mit $c_\alpha \neq 0$ z. B. mit Hilfe des WN-Algorithmus. Diese Berechnungen können parallel durchgeführt werden.

(3) **Summationsschritt:** Summiere die Koeffizienten $c_\alpha \neq 0$, deren Monome X^α in Schritt (2) zu denselben Normalformen führen.

(4) **Ausgabe:** Definiere das Polynom $\bar{f}$, dessen Terme die Monome aus Schritt (2) mit den entsprechenden Summen aus (iii) als Koeffizienten sind.

Offensichtlich ist $\bar{f}$ bezüglich $\mathcal{G}_{\text{lex}}$ nicht reduzierbar, und aus Korollar 4.4.6 folgt $\bar{f} = \bar{f}^{\mathcal{G}_{\text{lex}}}$, also das gewünschte Ergebnis. Im Fall einer beliebigen monomialen Ordnung $\succ$ und einer Gröbnerbasis $\mathcal{G}$ bez. $\succ$ kann folgendermaßen verfahren werden:

- Berechne zunächst mit Hilfe des PD-Algorithmus das Polynom $\bar{f} = \bar{f}^{\mathcal{G}_{\text{lex}}}$. Der totale Grad $|\beta|$ eines Monoms X^β dieses Polynoms ist durch $|\beta| = \beta_1 + \ldots + \beta_n < p_1 + \ldots + p_n$ beschränkt.

- Reduziere nun $\bar{f}^{\mathcal{G}_{\text{lex}}}$ bez. der Gröbnerbasis $\mathcal{G}$ mit dem Standardalgorithmus und erhalte das erwünschte Polynom $\bar{f} = \bar{f}^{\mathcal{G}}$. Dies geht schnell, da der Grad von $\bar{f}^{\mathcal{G}_{\text{lex}}}$ klein ist.

Im Fall einer endlichen abelschen Gruppe G vereinfacht sich der WN-Algorithmus wesentlich. Es bezeichne wie üblich e den Exponenten von G. Gegeben sei ein Wort w in den Erzeugern $g_1, \ldots, g_n$, das aufgrund der trivialen Kommutatorrelationen von G sofort in der Form $w := g_n^{\alpha_n} \cdot \ldots \cdot g_1^{\alpha_1}$ geschrieben werden kann, wobei für die Exponenten $\alpha_j < e$, $1 \leq j \leq n$, angenommen wird. Nun werden iterativ die Exponenten $e_n, \ldots, e_1$ der Normalform $g_n^{e_n} \cdot \ldots \cdot g_1^{e_1}$ von w berechnet. Der Koeffizient e_n ist durch $e_n := \alpha_n \bmod p_n$ gegeben. Seien

$$d := \alpha_n \text{ div } p_n \qquad \text{und} \qquad \alpha_j^{(n-1)} := \alpha_j + d \cdot a_{n,j} \bmod e \tag{4.13}$$

für $j = 2, \ldots, n$. Dann ist

$$w_{n-1} := g_{n-1}^{\alpha_{n-1}^{(n-1)}} \cdot \ldots \cdot g_1^{\alpha_1^{(n-1)}} \sim g_n^{-e_n} \cdot w \tag{4.14}$$

ein Element in G_{n-1} und ein Wort in den Erzeugern $g_{n-1}, \ldots, g_1$. Induktiv kann man nun mit dem Wort w_{n-1} fortfahren und auf diese Weise die Exponenten $e_{n-1}, \ldots, e_1$ sukzessiv bestimmen. Die Vereinfachung des WN-Algorithmus im abelschen Fall besteht also darin, daß aufgrund der trivialen Kommutatorrelationen im i-ten Schritt das Wort w_i sofort als Wort in den Erzeugern $g_i, \ldots, g_1$ geschrieben werden kann (vergleiche (4.14) mit (4.4)).

Die Multiplikationen modulo e in (4.13) werden aus denselben Gründen wie in (3) von Bemerkung 4.2.4 durch Additionen modulo e via binärer Methode ersetzt. Daraus folgt, daß die Anzahl der Additionen in $\mathbb{Z}_e$ des WN-Algorithmus im abelschen Fall logarithmisch von den Exponenten α_j abhängt und grob durch die Größenordnung

$$O(n^2 \cdot \log(e))$$

abgeschätzt werden kann. Es wird dabei keine Vorverarbeitungsphase oder zusätzlicher Speicherplatzaufwand für eine DFT benötigt. Die Division mit Rest zur Berechnung von d und e_i im i-ten Schritt spielen für die Gesamtkomplexität des Algorithmus keine Rolle.

Abschließend fassen wir das Ergebnis noch einmal zusammen. Beim PD-Algorithmus hängt die Gesamtkomplexität linear von der Anzahl der Terme des zu reduzierenden Polynoms f ab, aber nur logarithmisch im Multigrad der Terme. Allerdings können die Terme unabhängig voneinander und damit parallel verarbeitet werden. Damit eignet sich der PD-Algorithmus insbesondere für Polynome hohen Grades mit wenig Termen. Der Standardalgorithmus hängt dagegen im allgemeinen linear von den Multigraden der Terme ab und eignet sich für Polynome niedrigen Grades mit vielen Termen. Damit ist auch die Vorgehensweise bei beliebigen monomialen Ordnungen, erst mit dem PD-Algorithmus zu reduzieren und dann mit dem Standardalgorithmus fortzufahren, sinnvoll.

4.5 Nicht-kommutative Polynomringe

Abschließend diskutieren wir, wie sich die Überlegungen des vorherigen Abschnitts auf den Fall nicht-kommutativer Polynomringe übertragen lassen.

Wir führen als Vorbereitung einige Notationen ein und verweisen für Einzelheiten auf [55]. Sei X eine Menge, das sogenannte *Alphabet*. Die Menge aller Wörter über X, bezeichnet mit X^*, ist das von X erzeugte *freie Monoid* (freie Halbgruppe mit 1). Sei $X := \{X_n, \ldots, X_1\}$ und $\mathbb{K}$ ein Körper, dann wird die von X erzeugte Monoidalgebra $\mathbb{K}\langle X\rangle = \mathbb{K}\langle X_n, \ldots, X_1\rangle$ über $\mathbb{K}$ auch als *nicht-kommutativer Polynomring* über $\mathbb{K}$ in den unabhängigen nicht-kommutativen *Variablen* $X_n, \ldots, X_1$ bezeichnet. Als $\mathbb{K}$-Vektorraum wird dieser von *Monomen* der Form $X_{i_1} \cdots X_{i_k}$ erzeugt, die gerade den Elementen aus X^* entsprechen. Der nicht-kommutative Polynomring in n Variablen kann auch als Tensoralgebra eines n-dimensionalen freien $\mathbb{K}$-Moduls charakterisiert werden, wobei die Variablen den Elementen einer Basis des $\mathbb{K}$-Moduls entsprechen (siehe [38], S. 633).

Eine *monomiale Ordnung* auf $\mathbb{K}\langle X\rangle$ oder X^* ist, analog zur Definition 4.4.1, eine totale Ordnung $\succ$ der Monome von X^* mit den folgenden Eigenschaften:

(1) Translationsinvarianz: $\forall m_1, m_2, r, l \in X^* : m_1 \succ m_2 \Rightarrow lm_1r \succ lm_2r$.

(2) Wohlordnung: Jede nicht-leere Teilmenge von X^* hat ein kleinstes Element unter $\succ$.

(3) $m \succ 1$ für alle $m \in X^*$.

Man kann zeigen, daß im kommutativen Fall aus dem Lemma von Dickson folgt, daß für jede totale, translationsinvariante Ordnung die Voraussetzung der Wohlordnung äquivalent ist zu $\alpha \succ 0$ für alle $\alpha \in \mathbb{Z}_{\geq 0}$ (siehe [19], S. 70). Im nicht-kommutativen Fall ist dies nicht richtig. Ebenso ist das Lemma von Dickson nicht mehr gültig, wie das folgende Beispiel aus [44] zeigt: Sei $m_i := X_3^i X_2^i X_1 \in \mathbb{K}\langle X_3, X_2, X_1\rangle$, $i \in \mathbb{N}$, dann ist das von den m_i erzeugte Ideal nicht endlich erzeugt, da in der Menge $\{m_i | i \in \mathbb{N}\}$ kein Element ein Vielfaches eines anderen ist.

Für unsere weiteren Betrachtungen benötigen wir eine spezielle monomiale Ordnung auf X^*, für deren Konstruktion wir ein wenig ausholen müssen. Seien zunächst X und Y disjunkte Mengen und $\succ_X$ und $\succ_Y$ monomiale Ordnungen auf X^* bzw. Y^*. Wir konstruieren nun aus diesen Ordnungen eine monomiale Ordnung auf $(X \cup Y)^*$. Sei U ein Wort in $(X \cup Y)^*$. Dann kann U eindeutig in der Form

$$A_0 b_1 A_1 b_2 \ldots A_{r-1} b_r A_r$$

geschrieben werden, wobei die b_i in Y und die A_i in X^* sind. Sei V ein weiteres Wort in $(X \cup Y)^*$ mit der entsprechenden Zerlegung

$$C_0 d_1 C_1 d_2 \ldots C_{s-1} d_s C_s.$$

Definiere $V \succ U$, falls eine der beiden folgenden Bedingungen gilt:

(i) $d_1 \ldots d_s \succ_Y b_1 \ldots b_r$.

(ii) $b_1 \ldots b_r = d_1 \ldots d_s$ und $(A_0, \ldots, A_r)$ kommt vor $(C_0, \ldots, C_r)$ in der lexikographischen Ordnung von $(X^*)^{r+1}$, definiert durch $\succ_X$.

Man kann zeigen, daß $\succ$ eine monomiale Ordnung auf $(X \cup Y)^*$ definiert, die auch *Kranzprodukt* (wreath product) der Ordnungen $\succ_X$ und $\succ_Y$ genannt und mit $\succ_X \wr \succ_Y$ bezeichnet wird (siehe 2.1, [55]).

Sei nun wie oben $X := \{X_n, \ldots, X_1\}$. Auf $\{X_i\}^*$ gibt es eine eindeutig bestimmte monomiale Ordnung $\succ_i$, die durch die Länge des Wortes definiert ist. Dann ist

$$\succ_{\mathrm{w}} := \succ_1 \wr \succ_2 \wr \ldots \wr \succ_n \tag{4.15}$$

wegen der Assoziativität des Kranzproduktes wohldefiniert und bestimmt eine monomiale Ordnung auf X^*, die als die *grundlegende Kranzprodukt-Ordnung* (basic wreath-product ordering) bezeichnet wird. Es gilt z. B.

$$X_n \succ_{\mathrm{w}} X_{n-1} \succ_{\mathrm{w}} \ldots \succ_{\mathrm{w}} X_1 \qquad \text{und} \qquad X_i X_j \succ_{\mathrm{w}} X_j X_i,$$

für $1 \leq i < j \leq n$.

Die Grundlagen zur Theorie der nicht-kommutativen Gröbnerbasen wurden bereits von Bergman in [5] gelegt. Für eine Darstellung dieser Theorie verweisen wir auf [44]. Im folgenden sei ein Ideal $I \subset \mathbb{K}\langle X\rangle = \mathbb{K}\langle X_n, \ldots, X_1\rangle$ immer als zweiseitiges Ideal vorausgesetzt. Ist $\succ$ eine monomiale Ordnung auf $\mathbb{K}\langle X\rangle$, dann kann man völlig analog zum kommutativen Fall den führenden Term $\mathrm{LT}(f)$ für $f \in \mathbb{K}\langle X\rangle$ und das von $\mathrm{LT}(I)$ erzeugte Ideal der führenden Terme $\langle \mathrm{LT}(I)\rangle$ definieren. Eine Teilmenge $\mathcal{G} \subset I$ heißt *nicht-kommutative Gröbnerbasis*, falls $\langle \mathrm{LT}(I)\rangle = \langle \mathrm{LT}(g) | g \in \mathcal{G}\rangle$. Der Reduktionsprozeß (nicht-kommutative multivariate Polynomdivision) ist analog zum kommutativen Fall. Sei z. B. $f \in \mathbb{K}\langle X\rangle$ und o. B. d. A. $\mathrm{LC}(f) = 1$, also $f = \mathrm{LT}(f) + f'$ mit geeignetem $f' \in \mathbb{K}\langle X\rangle$. Ein Monom m hat $\mathrm{LT}(f)$ als Teilwort, falls $m = m_1 \mathrm{LT}(f) m_2$ für Monome m_1, m_2 gilt. Die *Reduktion* von m bez. f ist dann $m_1(-f')m_2$. Auf diese Weise kann rekursiv die Reduktion eines Elements aus $\mathbb{K}\langle X\rangle$ bez. einer beliebigen Folge von Elementen definiert werden. Wie im kommutativen Fall führt der Reduktionsprozeß bez. einer Gröbnerbasis $\mathcal{G}$ nach endlich vielen Schritten auf einen eindeutig bestimmten Rest. Allerdings treten bei der Erweiterung der Theorie der Gröbnerbasen auf den nichtkommutativen Fall große Schwierigkeiten auf (siehe [30]): Im allgemeinen ist das „Wortproblem“ für Ideale in $\mathbb{K}\langle X\rangle$ nicht entscheidbar, d. h. es kann keinen allgemeinen Algorithmus zur Konstruktion von Gröbnerbasen geben. Darüber hinaus sind Gröbnerbasen im nichtkommutativen Fall im allgemeinen nicht mehr endlich. In [44] findet man einen Algorithmus, der ausgehend von einer endlichen Basis eines Ideals $I \subset \mathbb{K}\langle X\rangle$ in endlich vielen Schritten eine endliche Gröbnerbasis von I berechnet, falls eine solche Basis existiert. Der Algorithmus stoppt nicht, wenn I keine endliche Gröbnerbasis hat, d. h. $\langle \mathrm{LT}(I)\rangle$ nicht endlich erzeugt ist.

Wir interessieren uns im folgenden für einen einfachen Spezialfall. Sei G wie in (2.7) eine konsistent pc-präsentierte endliche auflösbare Gruppe. Analog zum kommutativen Fall definiert $X_i \mapsto g_i$, $i = 1, \ldots, n$, einen surjektiven Homomorphismus $\varphi : \mathbb{C}\langle X_n, \ldots, X_1\rangle \to \mathbb{C}G$, dessen Kern $I_G := \ker(\varphi)$ ein zweiseitiges Ideal ist. Die pc-Relationen von G definieren analog zu (4.11) eine Menge $\mathcal{G} \subset I_G$ durch:

$$\begin{aligned} f_i &:= X_i^{p_i} - X_{i-1}^{a_{i,i-1}} \cdot \ldots \cdot X_1^{a_{i,1}}, && 1 \leq i \leq n, \\ f_{ij} &:= X_i X_j - X_j X_i X_{j-1}^{b_{ij,j-1}} \cdot \ldots \cdot X_1^{b_{ij,1}}, && 1 \leq i < j \leq n. \end{aligned}$$

Wir fixieren im folgenden die durch (4.15) definierte monomiale Ordnung $\succ_{\mathrm{w}}$ auf $\mathbb{C}\langle X\rangle$, dann gilt $\mathrm{LT}(f_i) = X_i^{p_i}$ und $\mathrm{LT}(f_{ij}) = X_i X_j$. In Verallgemeinerung des Satzes 4.4.7 gilt:

Satz 4.5.1 *Die Menge* $\mathcal{G} := \{f_i, 1 \leq i \leq n, f_{ij}, 1 \leq i < j \leq n\}$ *ist die reduzierte Gröbnerbasis von* $I_G \subset \mathbb{C}\langle X_n, \ldots, X_1\rangle$ *bezüglich der Kranzprodukt-Ordnung* $\succ_{\mathrm{w}}$.

Beweis: Dies zeigt man völlig analog wie im kommutativen Fall (siehe Beweis zu Satz 4.4.7). Aus Dimensionsgründen folgert man, daß I_G mit dem von $\mathcal{G}$ erzeugten Ideal I übereinstimmt. Die Menge $\{m \in X^* : m \notin \langle \mathrm{LT}(I)\rangle\}$ der Standardmonome besteht wiederum aus den $p_1 \cdot \ldots \cdot p_n$ Normalformen, woraus man wie im kommutativen Fall schließen kann, daß $\langle \mathrm{LT}(f_i), 1 \leq i \leq n, \mathrm{LT}(f_{ij}), 1 \leq i < j \leq n\rangle = \langle \mathrm{LT}(I_G)\rangle$ gelten muß. □

Die Reduktion nicht-kommutativer Polynome in $\mathbb{K}\langle X\rangle$ modulo Idealen der Form I_G kann damit analog zu Abschnitt 4.4.3 via Wortnormalisierung durchgeführt werden. Im Fall von überauflösbaren Gruppen kann dabei der in Abschnitt 4.1 beschriebene schnelle WN-Algorithmus als Unterroutine zur Wortnormalisierung verwendet werden.

Kapitel 5

FFTs überauflösbarer Gruppen

In Kapitel 3 wurde die *Generierung* einer DFT $D = \oplus_{k=1}^{h} D_k$ einer überauflösbaren Gruppe G behandelt. In diesem Kapitel wollen wir uns der *Auswertung* $\mathbb{C}G \ni a \mapsto D(a)$ des durch $D : \mathbb{C}G \longrightarrow \oplus_{k=1}^{h} \mathbb{C}^{d_k \times d_k}$ definierten Algebrenisomorphismus widmen. Aus algebraischer Sicht ist dies die simultane Auswertung einer Transversalen $D_1, \ldots, D_h$ von irreduziblen Darstellungen von G. In Matrixterminologie entspricht dies der Multiplikation der entsprechenden DFT-Matrix $\mathbf{D}$ mit einem Eingabevektor $\mathbf{a} := (a_g)_{g \in G}$.

Ein geeignetes Komplexitätsmaß für das DFT-Auswertungsproblem ist die von Baum und Clausen eingeführte lineare Komplexität von Matrizen und Gruppen (siehe [15], Kapitel 3, und Abschnitt 5.1 für eine Zusammenfassung). Bezüglich dieses Komplexitätsmaßes haben dieselben Autoren für allgemeine endliche Gruppen die untere Schranke $\frac{1}{4}|G| \log |G|$ für die DFT-Auswertung bewiesen.

Wie schon in der Einleitung erwähnt, wurde von Beth für endliche auflösbare Gruppen ein Algorithmus zur Auswertung verallgemeinerter DFTs angegeben, der dies mit $O(|G|^{3/2})$ Operationen bewerkstelligt [8]. Clausen hat in [14] für den Fall der symmetrischen Gruppen S_n einen Algorithmus beschrieben, der für die DFT-Auswertung $O(n^3 \cdot n!)$ Operationen benötigt, also asymptotisch fast linear in der Gruppenordnung $n!$ ist. Für den Fall endlicher überauflösbarer Gruppen hat Baum in [2] einen schnellen $O(|G| \log |G|)$-Algorithmus angegeben, der also im Hinblick auf die untere Schranke bis auf einen konstanten Faktor (der darüber hinaus klein ist und durch die Konstante 8.5 abgeschätzt werden kann) asymptotisch optimal ist. Weitere Resultate und Hinweise auf die Literatur findet man in [42].

In Abschnitt 5.2 fassen wir den in [2] beschriebenen rekursiven Baum-Algorithmus zusammen. Um auch während der Laufzeit des Algorithmus eine obere Schranke für den Speicherplatzbedarf zu garantieren, der linear in der Gruppenordnung $|G|$ ist, wurde der Baum-Algorithmus in einer iterativen Variante implementiert, die wir in Abschnitt 5.3 beschreiben. Man erkennt dann sofort, daß der Baum-Algorithmus einer Faktorisierung der DFT-Matrix in dünn besetzte Matrizen entspricht. In den beiden folgenden Abschnitten geben wir Beispiele und Details zur Implementierung und diskutieren das Laufzeitverhalten des Algorithmus.

5.1 Lineare Komplexität von Matrizen und Gruppen

Die *lineare Komplexität* $L(\mathbf{D})$ der DFT-Matrix $\mathbf{D}$ ist die minimal benötigte Anzahl von Additionen, Subtraktionen und Skalarmultiplikationen, um das Matrix-Vektor-Produkt $\mathbf{D} \cdot \mathbf{a}$ für beliebiges $\mathbf{a} \in \mathbb{C}^{|G|}$ zu berechnen. Falls die Programmkonstanten dem Betrage nach durch den Wert 2 beschränkt sind, wird die entsprechende minimale Anzahl $L_2(\mathbf{D})$ auch als 2-lineare Komplexität von $\mathbf{D}$ bezeichnet. Die lineare Komplexität $L(G)$ einer endlichen Gruppe G ist definiert durch

$$L(G) := \min\{L(\mathbf{D}) \mid \mathbf{D} \in \mathrm{DFT}(G)\}. \tag{5.1}$$

Analog definiert man $L_2(G)$. Es gelten trivialerweise die Abschätzungen

$$|G| - 1 \leq L(G) \leq 2|G| \cdot (|G| - 1) \quad \text{und} \quad L(G) \leq L_2(G). \tag{5.2}$$

Ein Satz von Morgenstern [43] zusammen mit den klassischen Schurrelationen führen zu der Abschätzung (siehe [3])

$$L_2(G) > \frac{1}{4}|G| \log |G|. \tag{5.3}$$

Kann also eine DFT mit nur $O(|G| \log |G|)$ Additionen, Subtraktionen und Skalarmultiplikationen mit kleinen Programmkonstanten ≤ 2 ausgeführt werden, so ist dies im L_2-Modell im wesentlichen optimal. Damit verdient also jeder Algorithmus, der eine DFT mit $O(|G| \log |G|)$ Operationen auswerten kann und dabei nur Programmkonstanten vom Betrag ≤ 2 benutzt, mit Recht den Namen schnelle Fouriertransformation bzw. FFT. Für weitere Details bezüglich unterer Komplexitätsschranken sei auf Kapitel 13 von [12] verwiesen.

Zum Beispiel treten bei der Cooley-Tukey FFT von $\mathbb{C}G$, G zyklische 2-Gruppe, nur Einheitswurzeln als Programmkonstanten auf. Die Anzahl der arithmetischen Operationen kann durch die obere Schranke $\frac{3}{2}|G| \log |G|$ abgeschätzt werden, und damit ist der Algorithmus im wesentlichen optimal im L_2-Modell.

5.2 Baum-Algorithmus

Aus Abschnitt 3.1 wissen wir bereits, daß jede überauflösbare Gruppe G mit Hauptreihe $\mathcal{T} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_1 \rhd G_0 = \{1\})$ eine e-monomiale $\mathcal{T}$-angepaßte DFT $D = \oplus_{k=1}^{h} D_k$ besitzt. Für alle $1 \leq i \leq n$ ist dann $D{\downarrow}G_i$ eine $\mathcal{T}_i$-angepaßte Darstellung, wobei $\mathcal{T}_i$ die Kette $(G_i \rhd \ldots \rhd G_0)$ bezeichne. Seien $F_1, \ldots, F_r$ die unterschiedlichen irreduziblen Konstituenten von $D{\downarrow}G_{n-1}$. Dann ist $D^{n-1} := \oplus_{\ell=1}^{r} F_\ell$ eine monomiale $\mathcal{T}_{n-1}$-angepaßte DFT von $\mathbb{C}G_{n-1}$. In unserem Komplexitätsmodell von Abschnitt 5.1 verursacht Kopieren keine Kosten, und es gilt daher $L(D^n{\downarrow}G_{n-1}) = L(D^{n-1})$. Diese Eigenschaft von $\mathcal{T}$-angepaßten DFTs macht sich nun der Baum-Algorithmus zunutze.

Statt $D = \oplus_{k=1}^{h} D_k$ an einem $a \in \mathbb{C}G_n$ direkt auszuwerten, wird a entsprechend der Nebenklassenzerlegung $G = \bigsqcup_{j=0}^{p-1} g^j G_{n-1}$, $p := [G : G_{n-1}]$, umgeschrieben. Mit geeigneten $a_j \in \mathbb{C}G_{n-1}$ gilt dann $a = \sum_{j=0}^{p-1} g^j a_j$ und damit

$$D(a) \;=\; \sum_{j=0}^{p-1} D(g^j)(D{\downarrow}G_{n-1})(a_j) \;=\; \sum_{j=0}^{p-1} \bigoplus_{k=1}^{h} D_k(g^j)(D_k{\downarrow}G_{n-1})(a_j). \tag{5.4}$$

Die Berechnung von $D(\downarrow G_{n-1})(a_j)$ entspricht bis auf zusätzliche (kostenlose) Kopieroperationen der Berechnung von $D^{n-1}(a_j)$. Damit kann also die Auswertung $D(a)$ der DFT D von G_n zurückgeführt werden auf

(i) p Auswertungen $D^{n-1}(a_0), \ldots, D^{n-1}(a_{p-1})$ der DFT D^{n-1} von G_{n-1},

(ii) p Multiplikationen mit den entsprechenden „Twiddle"-Faktoren $D(g^j)$ und

(iii) anschließender Addition der entprechenden p Summanden.

Diese Vorgehensweise führt zu einem rekursiven „Divide & Conquer"-Algorithmus. Im „Divide"-Schritt wird (i) und im „Conquer"-Schritt werden (ii) und (iii) berechnet. Dabei kann der „Conquer"-Schritt sehr effizient durchgeführt werden. Hierfür betrachten wir eine Darstellung D_k von $\mathbb{C}G$. Nach Abschnitt 2.4 gibt es für die Einschränkung $F := D_k{\downarrow}G_{n-1}$ genau zwei Fälle:

Fall 1. F ist irreduzibel, und es gibt genau p inäquivalente irreduzible Erweiterungen $D_k = D_{k_0}, \ldots, D_{k_{p-1}} \in \mathrm{Irr}(G, \mathcal{T})$ von F. Wegen der $\mathcal{T}$-Angepaßtheit sind die D_{k_l} Tensorprodukte $D_{k_l} = \chi^l \otimes D_k$ von D_k mit den linearen Charakteren $1 = \chi^0, \ldots, \chi^{p-1}$ der zyklischen Gruppe G/G_{n-1}. Es gilt also für $0 \le l \le p-1$

$$D_{k_l}(a) = \sum_{j=0}^{p-1} \chi^l(g^j G_{n-1}) D_k(g^j) F(a_j). \tag{5.5}$$

Damit können die Einträge der darstellenden Matrizen $D_{k_0}(a), \ldots, D_{k_{p-1}}(a)$ an der Stelle a simultan mit Hilfe von „lokalen" FFTs der zyklischen Gruppe der Ordnung p berechnet werden.

Fall 2. F ist nicht irreduzibel und läßt sich als direkte Summe von genau p inäquivalenten irreduziblen Darstellungen $F_{k_0}, \ldots, F_{k_{p-1}} \in \mathrm{Irr}(G_{n-1}, \mathcal{T}_{n-1})$ schreiben. In diesem Fall gilt also

$$D_k(a) = \sum_{j=0}^{p-1} D_k(g^j) \cdot (F_{k_0}(a_j) \oplus \ldots \oplus F_{k_{p-1}}(a_j)). \tag{5.6}$$

Die „Twiddle"-Faktoren $D_k(g^j)$ sind monomial. Darüber hinaus sind, wie unmittelbar aus der Definition der Induktion (siehe Abschnitt 2.1) ersichtlich ist, die p Summanden $\deg(F_{k_0})$-blockmonomiale Matrizen mit paarweise disjunktem Träger. Sind also die darstellenden Matrizen $F_{k_l}(a_j)$, $0 \le l, j < p$, gegeben, so ist zur Berechnung eines Eintrages von $D_k(a)$ nur eine Multiplikation nötig.

Eine Analyse des Baum-Algorithmus findet man in [2] bzw. [15]. Wir fassen das Ergebnis im folgenden Satz zusammen:

Satz 5.2.1 (Baum) *Ist G eine endliche überauflösbare Gruppe mit Hauptreihe $\mathcal{T}$, dann ist die $\mathcal{T}$-angepaßte DFT von G monomial und kann mit maximal $\gamma \cdot |G| \cdot \log|G|$ Operationen ausgewertet werden, wobei $1.5 \le \gamma \le 8.5$ von den Primteilern von $|G|$ abhängt.*

Der Baum-Algorithmus ist also, bis auf eine Konstante, optimal in dem in Abschnitt 5.1 definierten Komplexitätsmaß und verdient damit den Namen „Fast Fourier Transform" (FFT).

5.3 Baum-Algorithmus als Matrixfaktorisierung

Wie schon in der Einleitung erwähnt, entspricht die DFT-Auswertung in Matrixterminologie der Multiplikation der entsprechenden $|G| \times |G|$-DFT-Matrix $\mathbf{D}$ mit einem Eingabevektor $\mathbf{a} := (a_g)_{g \in G} \in \mathbb{C}^{|G|}$. In diesem Abschnitt untersuchen wir, wie der Baum-Algorithmus sich matrixtheoretisch deuten läßt.

Hierfür benötigen wir die folgende Notation. Sei G eine überauflösbare Gruppe der Ordnung $N := |G|$. Es sei $\mathcal{T}$ eine Hauptreihe von G mit Primfaktoren $p_i := [G_i : G_{i-1}]$, $i = 1, \ldots, n$. Für $1 \leq i < j \leq n$ definieren wir

$$I_{[j:i]} := [0 : p_j - 1] \times \ldots \times [0 : p_i - 1].$$

Mit $I := I_{[n:1]}$ gilt dann offensichtlich $I = I_{[n:i+1]} \times I_{[i:1]}$. Wir ordnen die Gruppenelemente $g \in G$ lexikographisch in den Exponenten ihrer Normalform (siehe (2.6)) an. Damit erhalten wir eine Numerierung der Gruppenelemente, die als Standardbasis der Gruppenalgebra $\mathbb{C}G$ dienen. In dieser Basis entspricht jedem Element $a \in \mathbb{C}G$ ein Vektor $\mathbf{a} \in \mathbb{C}^N$ der Form

$$\mathbf{a} = [a_{(0,\ldots 0,0)}, a_{(0,\ldots 0,1)}, \ldots, a_{(0,\ldots 0,p_1-1)}, a_{(0,\ldots 1,0))}, \ldots, a_{(\alpha_n,\ldots,\alpha_2,\alpha_1)}, \ldots, a_{(p_n-1,\ldots p_2-1,p_1-1)}]^T,$$

so daß

$$a = \sum_{(\alpha_n,\ldots,\alpha_1) \in I} a_{(\alpha_n,\ldots,\alpha_1)} \cdot g_n^{\alpha_n} \cdot \ldots \cdot g_1^{\alpha_1}$$

gilt. Definiert man für $(\alpha_n, \ldots, \alpha_{i+1}) \in I_{[n:i+1]}$

$$a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})} := \sum_{(\gamma_i,\ldots,\gamma_1) \in I_{[i:1]}} a_{(\alpha_n,\ldots,\alpha_{i+1},\gamma_i,\ldots,\gamma_1)} \cdot g_i^{\gamma_i} \cdot \ldots \cdot g_1^{\gamma_1} \in \mathbb{C}G_i,$$

so hat a bezüglich der Nebenklassenzerlegung von G_i in G die Zerlegung

$$a = \sum_{(\alpha_n,\ldots,\alpha_{i+1}) \in I_{[n:i+1]}} g_n^{\alpha_n} \cdot \ldots \cdot g_{i+1}^{\alpha_{i+1}} \cdot a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})},$$

und es gilt die Rekursionsformel

$$a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})} = \sum_{j=0}^{p_i-1} g_i^j \cdot a^{(i-1)}_{(\alpha_n,\ldots,\alpha_{i+1},j)}.$$

Wie in Abschnitt 3.3 bezeichne $D_{i,k}$, $0 \leq i \leq n$, $0 \leq k < h_i$, die k-te Darstellung von $\mathrm{Irr}(G_i, \mathcal{T}_i)$, wobei h_i die Anzahl der Konjugationsklassen von G_i ist. (Die Numerierung der D_k, die im folgenden keine wesentliche Rolle spielt, kann beliebig gewählt werden und muß dann fixiert werden. Wir wählen die durch die Implementierung des BC-Algorithmus vorgegebene Konstruktionsreihenfolge als Numerierung mit den in Abschnitt 3.3 beschriebenen Eigenschaften.) Im folgenden werden die Matrizen $D_{i,k}(b) \in \mathbb{C}^{d \times d}$, $d = \deg(D_{i,k}), b \in \mathbb{C}G_i$, durch Aneinanderhängen ihrer Zeilenvektoren als Zeilenvektoren der Länge d^2 interpretiert. Mit dieser Konvention entspricht die Auswertung der DFT in $a \in \mathbb{C}G$ der Berechnung des Matrix-Vektor-Produkts

$$\mathbf{D} \cdot \mathbf{a} = [D_{n,0}(a), D_{n,1}(a), \ldots, D_{n,h_n-1}(a)]^T.$$

In der iterativen, matrixtheoretischen Form hat der Baum-Algorithmus im i-ten Schritt, $1 \leq i < n$, den Vektor

$$\mathbf{a}^{(i-1)} := \left[D_{i-1,0}(a^{(i-1)}_{(0,\ldots,0)}), \ldots, D_{i-1,h_{i-1}-1}(a^{(i-1)}_{(0,\ldots,0)}), \ldots\ldots, D_{i-1,h_{i-1}-1}(a^{(i-1)}_{(p_n-1,\ldots,p_i-1)}) \right]^T$$

als Eingabe und den Vektor

$$\mathbf{a}^{(i)} := \left[D_{i,0}(a^{(i)}_{(0,\ldots,0)}), \ldots, D_{i,h_i-1}(a^{(i)}_{(0,\ldots,0)}), \ldots\ldots, D_{i,h_i-1}(a^{(i)}_{(p_n-1,\ldots,p_{i+1}-1)}) \right]^T$$

als Ausgabe. In dieser Notation gilt also $\mathbf{a}^{(0)} = \mathbf{a}$ und $\mathbf{a}^{(n)} = \mathbf{D} \cdot \mathbf{a}$. Die Vektoren $\mathbf{a}^{(i)}$, $0 \leq i \leq n$, haben alle die Länge

$$|I_{[n:i+1]}| \cdot \sum_{k=0}^{h_i-1} \deg(D_{i,k})^2 = [G : G_i] \cdot |G_i| = |G| = N,$$

und der Übergang von $\mathbf{a}^{(i-1)}$ zu $\mathbf{a}^{(i)}$ wird durch eine Matrix $\mathbf{M}^{(i)} \in \mathbb{C}^{N \times N}$ beschrieben. Im i-ten Schritt müssen nach Abschnitt 5.2 bei der Berechnung der Einträge von

$$D_{i,k}(a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})}) = \sum_{j=0}^{p_i-1} D_{i,k}(g_i)^j \cdot D_{i,k}(a^{(i-1)}_{(\alpha_n,\ldots,\alpha_{i+1},j)}), \tag{5.7}$$

die ja gerade für $k = 0, \ldots, h_i - 1$ und $(\alpha_n, \ldots, \alpha_{i+1}) \in I_{[n:i+1]}$ die Einträge von $\mathbf{a}^{(i)}$ ausmachen, zwei Fälle unterschieden werden. Im ersten Fall berechnen sich die Einträge von (5.7) nach der Formel (5.5). Mit anderen Worten ist in diesem Fall ein jeder solcher Eintrag von $\mathbf{a}^{(i)}$ eine komplexe Linearkombination von genau p_i Einträgen von $\mathbf{a}^{(i-1)}$ und die Matrix $\mathbf{M}^{(i)}$ hat in den entsprechenden Zeilen genau p_i von Null verschiedene Einträge. Im zweiten Fall berechnen sich die Einträge von (5.7) nach der Formel (5.6). Damit ist ein jeder solcher Eintrag von $\mathbf{a}^{(i)}$ ein Vielfaches eines Eintrages von $\mathbf{a}^{(i-1)}$ mit einem von Null verschiedenen komplexen Faktor, und die Matrix $\mathbf{M}^{(i)}$ hat in den entsprechenden Zeilen genau einen von Null verschiedenen Eintrag. Aus den Formeln (5.5) und (5.6) folgt weiterhin, daß für die Spalten der Matrix $\mathbf{M}^{(i)}$ „simultan“ analoge Aussagen gelten. Da die von Null verschiedenen Einträge der e-monomialen „Twiddle“-Faktoren e-te Einheitswurzeln sind, gilt dasselbe für die Matrizen $\mathbf{M}^{(i)}$. Wir fassen das Ergebnis in dem folgenden Korollar zusammen.

Korollar 5.3.1 *Aus matrixtheoretischer Sicht entspricht dem Baum-Algorithmus eine Faktorisierung der DFT-Matrix* $\mathbf{D}$ *in ein Produkt*

$$\mathbf{D} = \mathbf{M}^{(n)} \cdot \ldots \cdot \mathbf{M}^{(2)} \cdot \mathbf{M}^{(1)} \tag{5.8}$$

dünn besetzter Matrizen. Dabei hat die Matrix $\mathbf{M}^{(i)}$ *„simultan“ in jeder Zeile und Spalte entweder genau einen oder genau* p_i *von Null verschiedene Einträge. (D. h. ist* (u, v) *die Position eines von Null verschiedenen Eintrags von* $\mathbf{M}^{(i)}$*, dann haben die* u*-te Zeile und* v*-te Spalte von* $\mathbf{M}^{(i)}$ *entweder genau einen oder genau* p_i *von Null verschiedene Einträge.) Dabei sind alle von Null verschiedenen Einträge* e*-te Einheitswurzeln.*

Darüber hinaus folgt aus Abschnitt 5.2, daß jeweils p_i Zeilen mit p_i von Null verschiedenen Einträgen der Matrix $\mathbf{M}^{(i)}$ simultan durch eine „lokale“ zyklische FFT der Länge p_i ausgewertet werden können. Da eine zyklische FFT der Länge p_i mit $O(p_i \log(p_i))$ Operationen durchgeführt werden kann (siehe z. B. Kapitel 4 von [15]), ergibt sich bei der Matrix-Vektor-Multiplikation mit $\mathbf{M}^{(i)}$ im i-ten Schritt ein Aufwand der Größenordnung $O(\log(p_i))$ je Vektoreintrag (im zweiten Fall sogar nur $O(1)$). Damit folgt aus der Matrixfaktorisierung (5.8) sofort die folgende Größenordnung des Baum-Algorithmus (vergleiche mit Satz 5.2.1):

$$\sum_{i=1}^{n} O\left(|G| \cdot \log(p_i)\right) = O\left(|G| \log \prod_{i=1}^{n} p_i\right) = O\left(|G| \log |G|\right).$$

Da alle Programmkonstanten (Matrixeinträge) e-te Einheitswurzeln sind, gilt diese Abschätzung sogar im L_2-Modell von Abschnitt 5.1.

5.4 Beispiele

In diesem Abschnitt setzen wir die Beispiele von Abschnitt 3.3 fort und verwenden die dort eingeführte Notation.

5.4.1 Symmetrische Gruppe S_3

Mit unseren Konventionen zur Numerierung der Gruppenelemente und der Matrixkoeffizienten der darstellenden Matrizen hat die S_3 die folgende DFT-Matrix (ω primitive 6-te Einheitswurzel):

$$\mathbf{D} = \begin{array}{c} \begin{array}{cccccc} 1 & g_1 & g_1^2 & g_2 & g_2 g_1 & g_2 g_1^2 \end{array} \\ \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & \omega^2 & \omega^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \omega^4 & \omega^2 \\ 0 & 0 & 0 & 1 & \omega^2 & \omega^4 \\ 1 & \omega^4 & \omega^2 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{c} D_{2,0} \\ D_{2,1} \\ (D_{2,2})_{(0,0)} \\ (D_{2,2})_{(0,1)} \\ (D_{2,2})_{(1,0)} \\ (D_{2,2})_{(1,1)} \end{array}$$

Der Baum-Algorithmus impliziert eine Matrixzerlegung $\mathbf{D} = \mathbf{M}^{(2)} \cdot \mathbf{M}^{(1)}$, die wir im folgenden explizit herleiten. Mit der Notation aus Abschnitt 5.3 gilt

$$a = a_{(0)}^{(1)} + g_2 \cdot a_{(1)}^{(1)} = (a_{(0,0)} + a_{(0,1)} \cdot g_1 + a_{(0,2)} \cdot g_1^2) + g_2 \cdot (a_{(1,0)} + a_{(1,1)} \cdot g_1 + a_{(1,2)} \cdot g_1^2).$$

Die Vektoren $\mathbf{a}^{(i)}$, $0 \leq i \leq 3$, sind durch

$$\begin{aligned}
\mathbf{a}^{(0)} &= \left[a_{(0,0)}, a_{(0,1)}, a_{(0,2)}, a_{(1,0)}, a_{(1,1)}, a_{(1,2)}\right]^T \\
\mathbf{a}^{(1)} &= \left[D_{1,0}(a^{(1)}_{(0)}), D_{1,1}(a^{(1)}_{(0)}), D_{1,2}(a^{(1)}_{(0)}), D_{1,0}(a^{(1)}_{(1)}), D_{1,1}(a^{(1)}_{(1)}), D_{1,2}(a^{(1)}_{(1)})\right]^T \\
&= \left[a_{(0,0)}+a_{(0,1)}+a_{(0,2)}, a_{(0,0)}+\omega^2 a_{(0,1)}+\omega^4 a_{(0,2)}, a_{(0,0)}+\omega^4 a_{(0,1)}+\omega^2 a_{(0,2)}, a_{(1,0)}+a_{(1,1)}+a_{(1,2)}, \ldots\right]^T \\
\mathbf{a}^{(2)} &= \left[D_{2,0}(a), D_{2,1}(a), D_{2,2}(a)\right]^T \\
&= \left[D_{1,0}(a^{(1)}_{(0)})+D_{1,0}(a^{(1)}_{(1)}), D_{1,0}(a^{(1)}_{(0)})-D_{1,0}(a^{(1)}_{(1)}), D_{1,1}(a^{(1)}_{(0)}), D_{1,2}(a^{(1)}_{(1)}), D_{1,1}(a^{(1)}_{(1)}), D_{1,2}(a^{(1)}_{(0)})\right]^T
\end{aligned}$$

gegeben. Daraus ergeben sich unmittelbar die Matrizen

$$\mathbf{M}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & \omega^2 & \omega^4 & 0 & 0 & 0 \\ 1 & \omega^4 & \omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & \omega^2 & \omega^4 \\ 0 & 0 & 0 & 1 & \omega^4 & \omega^2 \end{bmatrix} \quad \text{und} \quad \mathbf{M}^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Bei $\mathbf{a}^{(1)} = \mathbf{M}^{(1)} \cdot \mathbf{a}^{(0)}$ können jeweils die ersten drei und die letzten drei Einträge simultan durch „lokale“ zyklische FFTs der Länge 3 berechnet werden. Bei $\mathbf{a}^{(2)} = \mathbf{M}^{(2)} \cdot \mathbf{a}^{(1)}$ können die ersten beiden Einträge simultan durch eine „lokale“ zyklische FFT der Länge 2 berechnet werden.

5.4.2 Zyklische Gruppe C_{2^n}

Der Baum-Algorithmus ist im Fall der zyklischen Gruppe C_{2^n} nichts anderes als der Cooley-Tukey-Algorithmus. Es sei im folgenden $N := 2^n$ und ω eine primitive N-te Einheitswurzel. Für $G = C_N$ besteht $\mathrm{Irr}(G)$ nur aus 1-dimensionalen Darstellungen und wird zusammen mit dem Tensorprodukt als Multiplikation ebenfalls zu einer zyklischen Gruppe der Ordnung N. Bei der klassischen DFT-Matrix $(\omega^{jk})_{0 \leq j,k < N}$ sind die Elemente der Standardbasen in $\mathbb{C}G$ bzw. $\mathrm{Irr}(G)$, also die Gruppenelemente bzw. die 1-dimensionalen Darstellungen, entsprechend dem Exponenten ihres jeweiligen Erzeugers angeordnet. Diese Numerierung stimmt nicht mit unseren Konventionen zur Numerierung überein. Sei $k \in [0 : 2^n - 1]$ und $(\alpha_1 \alpha_2 \ldots \alpha_n)$ die Binärdarstellung von k, dann folgt aus der pc-Darstellung (3.11)

$$g_n^k = g_n^{\alpha_n} \cdot \ldots \cdot g_2^{\alpha_2} \cdot g_1^{\alpha_1}. \tag{5.9}$$

Mit anderen Worten, die beiden Numerierungen der Gruppenelemente sind durch die Permutation β_n aus (3.12) ineinander überführbar. Sei $\mathbf{B}^{(n)}$ die Permutationsmatrix zu β_n, dann ist $\mathbf{B}^{(n)}$ offensichtlich zu sich selbst invers. Aus (3.13) und (5.9) folgt dann für die DFT-Matrix

$$\mathbf{D} = \mathbf{B}^{(n)} \cdot (\omega^{jk}) \cdot \mathbf{B}^{(n)}$$

Der Baum-Algorithmus liefert eine Matrixzerlegung $\mathbf{D} = \mathbf{M}^{(n)} \cdot \ldots \cdot \mathbf{M}^{(2)} \cdot \mathbf{M}^{(1)}$. Mit der Notation aus Abschnitt 5.3 gilt für ein Element $a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})} \in \mathbb{C}G_i$

$$D_{i,k}(a^{(i)}_{(\alpha_n,\ldots,\alpha_{i+1})}) = D_{i-1,\lfloor k/2 \rfloor}(a^{(i-1)}_{(\alpha_n,\ldots,\alpha_{i+1},0)}) \; + \; D_{i,k}(g_i) \cdot D_{i-1,\lfloor k/2 \rfloor}(a^{(i-1)}_{(\alpha_n,\ldots,\alpha_{i+1},1)}).$$

Hieraus folgt mit $b(i,k) := D_{i,k}(g_i) = \omega^{2^{n-i}\beta_i(k)}$ (unter Benutzung von (3.13)), daß $\mathbf{M}^{(i)}$ eine Blockdiagonalmatrix der Form $\mathrm{Id}_{[G:G_i]} \otimes \mathbf{N}^{(i)}$ ist mit einer $|G_i| \times |G_i|$-Matrix

$$\mathbf{N}^{(i)} := \begin{bmatrix} 1 & 0 & \dots & 0 & b(i,0) & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & b(i,1) & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & b(i,2) & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & b(i,3) & \dots & 0 \\ & & \ddots & & & & \ddots & \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & b(i,2^i-2) \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & b(i,2^i-1) \end{bmatrix}.$$

Es folgt leicht $b(i,2k+1) = -b(i,2k)$, $0 \le k < 2^{i-1}$, so daß bei $\mathbf{a}^{(i)} = \mathbf{M}^{(i)} \cdot \mathbf{a}^{(i-1)}$ jeweils zwei Einträge mittels einer Multiplikation und einer „lokalen“ zyklischen FFT der Länge 2 berechnet werden können.

Abschließend sei angemerkt, daß die abelschen Gruppen, wie schon im Fall der DFT-Datenstruktur und des BC-Algorithmus, den „worst case“ hinsichtlich der Laufzeit des Baum-Algorithmus darstellen. Dies geht auch aus den Tabellen des Abschnitts 5.5 hervor.

5.5 Implementierung

Der Baum-Algorithmus wurde in seiner iterativen, matrixtheoretischen Version unter denselben Bedingungen wie in Abschnitt 3.4 implementiert und getestet. Die Eingabe des Baum-Algorithmus besteht aus der DFT-Datenstruktur, die durch den BC-Algorithmus berechnet wurde, und einem Vektor (oder Signal) $\mathbf{a}^{(0)} = \mathbf{a} \in \mathbb{C}^{|G|}$. Es werden dann iterativ die Vektoren $\mathbf{a}^{(i)}$, $i = 1, \dots, n$, bestimmt, wobei zur Berechnung von $\mathbf{a}^{(i)}$ nur die Daten von $\mathbf{a}^{(i-1)}$ benötigt werden. Daher kommt die Implementierung des Baum-Algorithmus mit einem Speicherplatz aus, der linear in der Gruppenordnung $|G|$ ist.

Beim Baum-Algorithmus werden die Darstellungen $D_{i,k}$, wie aus der Formel (5.4) hervorgeht, nur auf dem Erzeuger g_i ausgewertet, nicht aber auf den Erzeugern $g_1, \dots, g_{i-1}$. Dies sind genau die darstellenden Matrizen, die von der DFT-Datenstruktur zur Verfügung gestellt werden, so daß sich die DFT-Datenstruktur optimal zur Verwendung im Baum-Algorithmus eignet.

Zur Durchführung der „lokalen“ zyklischen FFTs der Länge p wurde der Bluestein-Algorithmus (siehe z. B. [9] bzw. [15]) implementiert, der eine Laufzeit der Größenordnung $O(|p| \log |p|)$ garantiert. Falls $|G|$ einen großen Primteiler hat, sind gerade diese „lokalen“ FFTs entscheidend für die schnelle Laufzeit des Baum-Algorithmus.

Die Inverse der DFT-Matrix $\mathbf{D}$ wird durch die inversen Matrizen $(\mathbf{M}^{(i)})^{-1}$ faktorisiert, die ebenfalls dünn besetzt sind und dieselbe Struktur wie die $\mathbf{M}^{(i)}$ haben. Die „lokalen“ zyklischen FFTs müssen nun durch die inversen „lokalen“ FFTs ersetzt werden. Hieraus ergibt sich unmittelbar, daß sich die Inversion der DFT durch eine leicht modifizierte Variante des Baum-Algorithmus realisieren läßt und eine Laufzeit in derselben Größenordnung hat. Die

Implementierung des Baum-Algorithmus und der inversen Variante werden im folgenden mit FFT bzw. IFFT bezeichnet.

5.5.1 Laufzeiten

In den folgenden Tabellen sind für verschiedene endliche überauflösbare Gruppen (siehe Abschnitt 3.4.1) die Laufzeiten der FFT und IFFT angegeben. Experimentell hat sich gezeigt, daß bei den Implementierungen die Laufzeiten völlig unabhängig von dem zu transformierenden Eingabevektor $\mathbf{a} \in \mathbb{C}^{|G|}$ sind. Daher wurde in den folgenden Versuchen bei allen Transformationen derselbe Eingabevektor verwendet. In den ersten vier Spalten der Tabellen findet man den Gruppennamen, die Gruppengröße $|G|$, die Länge n der Hauptreihe und die Anzahl h der Konjugationsklassen. In den folgenden Spalten sind die Laufzeiten t(DFT) für die DFT-Berechnung, t(FFT) für die FFT-Berechnung und t(IFFT) für die IFFT-Berechnung jeweils in Millisekunden angegeben. Wie zuvor wurden dabei alle Berechnungen mehrfach (meist zehnmal) durchgeführt und die entsprechenden Laufzeiten gemittelt. Um einen besseren Vergleich der Laufzeiten für unterschiedliche Gruppen zu gewährleisten, sind auch die Quotienten $t(\text{FFT})/|G|$ und $t(\text{IFFT})/|G|$ in den Tabellen aufgeführt. Es sei daran erinnert, daß die DFT-Berechnung rein symbolisch in der additiven Gruppe $\mathbb{Z}_e$ erfolgt, während es sich bei der FFT und IFFT um Transformationen eines komplexen Signals handelt, also die Gleitpunktarithmetik verwendet wird (siehe Abschnitt 5.5.2). Gleitkommaoperationen sind natürlich wesentlich komplexer als Additionen in $\mathbb{Z}_e$, was sich in den unterschiedlichen absoluten Laufzeiten für die DFT und die FFT niederschlägt.

Wir starten unsere Diskussion wie in den vorherigen Abschnitten mit direkten Produkten $G = (S_3)^m$ der symmetrischen Gruppe für wachsendes $m = 3, 4, \ldots, 8$ (siehe Tabelle 5.1). Die Laufzeiten steigen nur wenig mehr als linear in der Gruppenordnung. Dieses Laufzeitverhalten spiegelt sehr gut die in Satz 5.2.1 angegebene theoretische obere Schranke $O(|G| \log |G|)$ wider. Aufgrund der begrenzten Speicherplatzkapazität des Hauptspeichers konnten nur Versuche bis $m = 8$ durchgeführt werden (siehe Abschnitt 5.5.2).

G	$\lvert G\rvert$	n	h	t (DFT)	t (FFT)	$t/\lvert G\rvert$	t (IFFT)	$t/\lvert G\rvert$
$(S_3)^3$	216	6	27	< 1	< 1		< 1	
$(S_3)^4$	1296	8	81	2	9	0.0070	14	0.0108
$(S_3)^5$	7776	10	243	16	97	0.0125	72	0.0093
$(S_3)^6$	46656	12	729	92	401	0.0086	522	0.0112
$(S_3)^7$	279936	14	2187	264	2874	0.0103	3710	0.0133
$(S_3)^8$	1679616	16	6561	1187	19586	0.0117	23925	0.0142

Tabelle 5.1: Laufzeiten in Millisekunden der FFT und IFFT für $(S_3)^m$.

Wie zuvor erwähnt, spielen die durch den Bluestein-Algorithmus realisierten „lokalen“ zyklischen FFTs eine entscheidende Rolle für die Effizienz der FFT. Bei den in Tabelle 5.2 verwendeten zyklischen Gruppen $G = C_p$ von Primzahlordnung p besteht die FFT bez. G aus einer „lokalen“ zyklischen FFT dieser Länge. Die angegebenen Laufzeiten spiegeln damit das Laufzeitverhalten des implementierten Bluestein-Algorithmus wider, die auch in diesem Fall gut mit der theoretischen Größenordnung $O(|G| \log |G|)$ korrespondieren.

G	$\|G\|$	n	h	t (DFT)	t (FFT)	$t/\|G\|$	t (IFFT)	$t/\|G\|$
C_{97}	97	1	97	< 1	< 1		< 1	
C_{997}	997	1	997	< 1	18	0.0181	22	0.0221
C_{9973}	9973	1	9973	9	336	0.0337	303	0.0304
C_{99991}	99991	1	99991	105	3507	0.0351	3530	0.0353

Tabelle 5.2: Laufzeiten des Bluestein-Algorithmus.

Bei unterschiedlichen Gruppen derselben Gruppenordnung gibt es nur kleine Unterschiede im Laufzeitverhalten der FFT und IFFT (siehe Tabelle 5.3 für Beispiele). Während bei der DFT insbesondere die Zahlen $d^1(G)$ und κ (Anzahl der Knoten im Charaktergraph) und weniger die Gruppenordnung $|G|$ entscheidend für die Laufzeiten sind, gibt es bei der FFT und IFFT nur eine leichte Tendenz in Richtung sinkender Laufzeiten bei fallenden h und $d^1(G)$. Dies ist eigentlich überraschend. Aus Korollar 5.8 und der anschließenden Diskussion folgt, daß Induktionen auf Stufe i der DFT-Generierung zu besonders dünn besetzten Matrizen $\mathbf{M}^{(i)}$ (mit nur einem von Null verschiedenen Eintrag in den entsprechenden Zeilen) führen. Da viele Induktionen zu einem zur Gruppenordnung kleinen $d^1(G)$ korrespondieren, sollte man daher auch bei der FFT und IFFT eine stärkere Abhängigkeit der Laufzeiten von $d^1(G)$ erwarten. Eine mögliche Erklärung ist, daß bei den Implementierungen zur Verwaltung der Eingabevektoren $\mathbf{a} \in \mathbb{C}^{|G|}$ zusätzliche Operationen benötigt werden (z. B. Kopieroperationen, Speicherplatzallokationen), die linear in der Gruppenordnung $|G|$, aber unabhängig von der jeweiligen Struktur der DFT sind. Diese DFT-unabhängigen, zusätzlichen Operationen machen die Abhängigkeit der FFT-Laufzeiten von der DFT-Struktur weniger signifikant.

G	$\|G\|$	n	h	t (DFT)	t (FFT)	$t/\|G\|$	t (IFFT)	$t/\|G\|$
$\mathrm{Lib}_{44100}(6)$	44100	8	44100	239	992	0.0225	1122	0.0254
$\mathrm{Lib}_{44100}(1)$	44100	8	22050	444	1032	0.0234	1068	0.0242
$\mathrm{Lib}_{44100}(13)$	44100	8	13230	75	915	0.0207	1033	0.0234
$\mathrm{Lib}_{44100}(3)$	44100	8	3750	140	857	0.0194	920	0.0209
$\mathrm{Lib}_{44100}(19)$	44100	8	3675	41	870	0.0197	910	0.0206

Tabelle 5.3: Laufzeiten für verschiedene Gruppen derselben Größe.

G	$\|G\|$	n	h	t (DFT)	t (FFT)	$t/\|G\|$	t (IFFT)	$t/\|G\|$
$\mathrm{Lib}_{1024}(1)$	1024	10	175	7	10	0.0098	12	0.0117
$\mathrm{Lib}_{2187}(1)$	2187	7	155	4	17	0.0078	22	0.0101
$\mathrm{Lib}_{7560}(1)$	7560	8	2295	114	115	0.0152	129	0.0171
$\mathrm{Lib}_{7560}(54)$	7560	8	945	4	169	0.0224	145	0.0192
$\mathrm{Lib}_{7776}(1)$	7776	10	333	20	92	0.0118	71	0.0091
$\mathrm{Lib}_{15625}(35)$	15625	6	265	3	349	0.0223	324	0.0207
$\mathrm{Lib}_{16000}(1)$	16000	10	760	37	319	0.0199	308	0.0193
$\mathrm{Lib}_{22287}(1)$	22287	4	22287	90	447	0.0201	473	0.0212
$\mathrm{Syl}_2(S_{16})$	32768	15	230	145	369	0.0113	396	0.0121
$\mathrm{Lib}_{23940}(1)$	23940	7	3990	115	403	0.0168	430	0.0180
$\mathrm{Lib}_{179894}(3)$	179894	5	179894	255	4375	0.0243	4333	0.0241

Tabelle 5.4: Laufzeiten für verschiedene Gruppen der Bibliothek.

In Tabelle 5.4 sind die FFT- und IFFT-Laufzeiten für die Gruppen aus Tabelle 3.4 angegeben. Die in den vorherigen Versuchen getroffenen Aussagen zu den Laufzeiten spiegeln sich auch in diesen Beipielen wider. Auffallend sind die vergleichsweise geringen Laufzeiten der 2-Gruppe $\mathrm{Syl}_2(S_{16})$ der Ordnung $32768 = 2^{15}$ oder der Gruppe $\mathrm{Lib}_{7776}(1)$ der Ordnung $7776 = 2^5 \cdot 3^5$. Diese sind darin begründet, daß die bei der Berechnung der „lokalen“ FFTs benötigten zyklischen FFTs für die Längen $p = 2$ und $p = 3$ optimiert wurden und für die restlichen Primzahlfaktoren der allgemeine Bluestein-Algorithmus verwendet wird.

Die FFTs bez. elementar-abelscher Gruppen sind gerade die sogenannten *Walsh-Hadamard-Transformationen*. Um einen Vergleich unserer FFT-Implementierung für allgemeine überauflösbare Gruppen mit gängigen Implementierungen für die Walsh-Hadamard-Transformation zu erlauben, sind in der folgenden Tabelle 5.5 die FFT- und IFFT-Laufzeiten bez. einiger elementar-abelscher Gruppen angegeben.

G	$\lvert G\rvert$	n	h	t (DFT)	t (FFT)	$t/\lvert G\rvert$	t (IFFT)	$t/\lvert G\rvert$
$(C_2)^{17}$	131072	17	131072	5035	2431	0.0185	2643	0.0202
$(C_3)^{11}$	177147	11	177147	1627	2421	0.0137	3141	0.0177
$(C_7)^6$	117649	6	117649	359	2981	0.0253	3005	0.0255
$(C_{11})^5$	161051	5	161051	353	4153	0.0258	4195	0.0260
$(C_{19})^4$	130321	4	130321	250	2999	0.0230	3043	0.0234
$(C_{53})^3$	148877	3	148877	231	2651	0.0178	2756	0.0185
$(C_{383})^2$	146689	2	146689	205	2768	0.0189	2571	0.0175
C_{149993}	149993	1	149993	199	7196	0.048	7193	0.0480

Tabelle 5.5: Laufzeiten für elementar-abelsche Gruppen.

5.5.2 Gleitkommaarithmetik

In diesem Abschnitt wollen wir kurz auf die Probleme eingehen, die bei Benutzung von Gleitkommaarithmetik entstehen. Die Auswertung der DFT einer endlichen Gruppe G ist eine invertierbare Transformation $\mathbb{C}^{|G|} \to \mathbb{C}^{|G|}$ komplexer Vektoren (Signale). Eine komplexe Zahl kann in einer Maschine, in der die Menge der darstellbaren Zahlen endlich ist, nur approximativ dargestellt werden. Für reelle Zahlen wird meist die sogenannte *Gleitkommadarstellung* verwendet, in der eine Maschinenzahl durch Exponent und Mantisse mit einer jeweils festen Anzahl an Stellen dargestellt ist (siehe [58]). In unserer Implementierung wird eine reelle Zahl durch den Datentyp `double` dargestellt. Ein `double` besteht aus 8 Bytes und deckt einen Zahlenbereich von $+/-$ 1.7E308 mit einer Genauigkeit von mindestens 15 Stellen ab. Ein Vektor $\mathbf{a} \in \mathbb{C}^{|G|}$ ist also in Form von $2|G|$ Zahlen vom Typ `double` gegeben. Ist z. B. $G = S_3^8$, dann benötigt ein komplexer Eingabevektor $\mathbf{a}$ schon $16 \cdot |G| \approx 27$ MB, so daß man für noch größere Gruppen schnell an die Grenzen der Hauptspeicherkapazität stößt (bei unserem Rechner 128 MB). Dies ist auch der Grund, warum unsere in Tabelle 5.1 dargestellte Versuchsreihe mit der Gruppe $G = S_3^8$ endet.

Die Einträge einer durch den BC-Algorithmus berechneten DFT-Matrix (gegeben in Form der DFT-Datenstruktur) sind e-te Einheitswurzeln und liegen symbolisch in Form der Exponenten einer primitiven e-ten Einheitswurzel ω vor. Zur Auswertung der DFT-Matrix mittels der FFT auf einem Eingabevektor $\mathbf{a}$, der in Gleitkommadarstellung gegeben ist, müssen daher

die Einheitswurzeln in Gleitkommazahlen umgewandelt werden, wobei aus signaltheoretischen Gründen die Einheitswurzel $\omega = \exp(-\frac{2\pi i}{e})$ gewählt wurde. Hierbei kommt es im allgemeinen zu Rundungsfehlern, die sich bei Anwendung der FFT auf das zu transformierende Signal $\mathbf{a}$ fortpflanzen und durch die Gleitkommaarithmetik verstärken. Nachfolgende Anwendung der IFFT führt also nicht auf das ursprüngliche Signal $\mathbf{a}$ zurück, sondern auf ein leicht gestörtes Signal. Man spricht hierbei auch von *algorithmischem Rauschen*. Durch kaskadierte Anwendung der FFT und IFFT (und eventueller Zwischenverarbeitung) können sich die Rundungsfehler aufschaukeln, so daß es in konkreten Anwendungen, insbesondere bei hochauflösenden Signalanalysen, wo lange FFTs benötigt werden, zu nicht mehr akzeptablen Verzerrungen des Signals kommen kann. Für die zyklische FFT (Cooley-Tukey) gibt es theoretische obere Schranken der Rundungsfehler bei Verwendung der Gleitkommaarithmetik (siehe z. B. [50]). Wir wollen mit den folgenden Tabellen illustrieren, wie sich die Rundungsfehler bei verallgemeinerten FFTs verhalten. Dabei ist $\mathbf{a}$ ein komplexer Eingabevektor der Länge $|G|$ und $\|\mathbf{a}\|_2$ die euklidische Norm, d. h. die Energie von $\mathbf{a}$. Die Energie des Fehlersignals nach m-facher Kaskadierung, $m \in \mathbb{N}$, der FFT und IFFT ist mit $\mathrm{err}(m)$ bezeichnet.

Natürlich hängt bei Gleitkommarechnung der Rundungsfehler von der Energie des Eingabevektors ab, so daß nur ein Vergleich der normalisierten Rundungsfehler $\mathrm{err}(m)/\|\mathbf{a}\|_2$ sinnvoll ist. Allerdings sind die Rundungsfehler nicht nur von der Energie des Signals, sondern auch von den Mantissen der Signalwerte abhängig, wie die beiden Beispiele von Tabelle 5.6 illustrieren. Während es sich beim ersten Beispiel um ein Chirpsignal handelt, sind die Signalwerte des zweiten Beispiels ganze Zahlen (so daß in diesem Fall fast alle Stellen der Mantisse Null sind).

G	$\lvert G\rvert$	$\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(1)$	$\mathrm{err}(10)$	$\mathrm{err}(1)/\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(10)/\lVert\mathbf{a}\rVert_2$
$\mathrm{Lib}_{179894}(1)$	179894	$4.4 \cdot 10^{7}$	$3.3 \cdot 10^{-3}$	$3.3 \cdot 10^{-2}$	$7.5 \cdot 10^{-11}$	$7.5 \cdot 10^{-10}$
$\mathrm{Lib}_{179894}(1)$	179894	$4.2 \cdot 10^{2}$	$4.2 \cdot 10^{-6}$	$4.2 \cdot 10^{-5}$	$1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-7}$

Tabelle 5.6: Rundungsfehler in Abhängigkeit vom Eingabevektor.

Bei unseren Versuchen stellte sich heraus, daß die Rundungsfehler in etwa linear in der Länge m der Kaskade sind, wie auch die Beispiele in den Tabellen 5.6 und 5.7 zeigen.

G	$\lvert G\rvert$	$\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(1)/\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(10)/\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(10^2)/\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(10^3)/\lVert\mathbf{a}\rVert_2$	$\mathrm{err}(10^4)/\lVert\mathbf{a}\rVert_2$
G_{128}	128	$1.13 \cdot 10^{1}$	$7.37 \cdot 10^{-15}$	$7.27 \cdot 10^{-14}$	$7.25 \cdot 10^{-13}$	$7.23 \cdot 10^{-12}$	$7.23 \cdot 10^{-11}$
D_{97}	194	$1.39 \cdot 10^{1}$	$4.81 \cdot 10^{-13}$	$4.81 \cdot 10^{-12}$	$4.81 \cdot 10^{-11}$	$4.81 \cdot 10^{-10}$	$4.81 \cdot 10^{-9}$

Tabelle 5.7: Rundungsfehler in Abhängigkeit von der Anzahl der Kaskaden.

Interessanter ist die Beobachtung, daß die Rundungsfehler bei verallgemeinerten FFTs nicht von der Länge des Signals abhängen, sondern vom Exponenten $e(\mathcal{T})$ der in G gewählten Hauptreihe $\mathcal{T}$. (siehe Abschnitt 3.5.2). Dies wird durch die Beispiele in Tabelle 5.8 belegt. So sind z. B. die Rundungsfehler bei den zyklischen Gruppe C_{99991}, $C_{2^{16}}$ und C_{44100}, bei denen der Exponent $e(\mathcal{T})$ mit der jeweiligen Gruppenordnung übereinstimmt, um Ordnungen größer als bei den Gruppen mit kleinem Exponenten wie z. B. $(C_2)^{16}$ und $(S_3)^7$.

G	$\lvert G\rvert$	$e(\mathcal{T})$	$\lVert\mathbf{a}\rVert_2$	err(1)	err(1)/$\lVert\mathbf{a}\rVert_2$
C_{99991}	99991	99991	$1.8 \cdot 10^{7}$	$4.9 \cdot 10^{-1}$	$2.7 \cdot 10^{-8}$
$(S_3)^7$	279936	6	$5.3 \cdot 10^{2}$	$2.6 \cdot 10^{-10}$	$4.8 \cdot 10^{-13}$
$C_{2^{16}}$	65536	65536	$2.6 \cdot 10^{2}$	$6.3 \cdot 10^{-7}$	$2.4 \cdot 10^{-9}$
$(C_2)^{16}$	65536	2	$2.6 \cdot 10^{2}$	$1.5 \cdot 10^{-11}$	$6.0 \cdot 10^{-14}$
C_{44100}	44100	44100	$2.1 \cdot 10^{2}$	$3.8 \cdot 10^{-7}$	$1.8 \cdot 10^{-9}$
$\mathrm{Lib}_{44100}(19)$	44100	1050	$2.1 \cdot 10^{2}$	$6.8 \cdot 10^{-9}$	$3.2 \cdot 10^{-11}$
$(C_2)^2 \times (C_3)^2 \times (C_5)^2 \times (C_7)^2$	44100	210	$2.1 \cdot 10^{2}$	$1.1 \cdot 10^{-10}$	$5.3 \cdot 10^{-13}$

Tabelle 5.8: Rundungsfehler in Abhängigkeit vom Exponenten $e(\mathcal{T})$.

Die Gleitkommaarithmetik ist zwar recht genau, aber auch bei Verwendung von spezialisierten Prozessoren für die meisten Echtzeitanwendungen zu langsam und für Hardwareimplementierungen zu teuer. Wird aus diesen Gründen die günstigere Festkommaarithmetik verwendet, so hat man den Nachteil eines wesentlich verstärkten algorithmischen Rauschens. Man muß daher nach anderen Wegen zur schnellen Realisierung der komplexen Arithmetik suchen. Clausen stellt in [13], Abschnitt 11, ein Verfahren vor, das die Explosion von Rundungsfehlern beim approximativen Rechnen mit komplexen Zahlen durch das (parallele) Rechnen in mehreren (kleinen) endlichen Körpern verhindert. Grundlage hierfür ist der dort bewiesene Satz:

Satz 5.5.1 *Sei ω eine primitive n-te Einheitswurzel. $\mathbb{Z}[\omega]$ liegt genau dann dicht in $\mathbb{C}$, wenn $n \notin \{1, 2, 3, 4, 6\}$.*

Die komplexen Einträge des Eingabevektors werden für geeignetes ω durch ganze Zahlen $\mathbb{Z}[\omega]$ des Kreisteilungskörpers $\mathbb{Q}[\omega]$ approximiert. Nur in diesem Schritt treten Approximationsfehler auf, die aber durch die Güte der Approximation kontrolliert werden können. Anschließende Berechnungen der FFT und IFFT können dann rein symbolisch über $\mathbb{Z}[\omega]$ durchgeführt werden, so daß keine weiteren Rundungsfehler entstehen. Um eine Explosion der ganzen Zahlen zu verhindern, wird zusätzlich eine auf dem Chinesischen Restsatz basierende modulare Technik angewendet. Shokrollahi hat in [53], Kapitel 4, ein effizientes Verfahren zur schnellen Approximation komplexer Zahlen in $\mathbb{Z}[\exp(2\pi i/2^n)]$ für $n > 3$ beschrieben, so daß schnelle zyklische FFT-Berechnungen mit Festkommaprozessoren und hoher Präzision möglich werden.

Da die durch den BC-Algorithmus berechnete verallgemeinerte DFT überauflösbarer Gruppen rein symbolisch und damit exakt erfolgt, ist eine Verbindung zu den von Clausen und Shokrollahi entwickelten Verfahren zur symbolischen Auswertung der DFT naheliegend. Eine wichtige Größe wird hier der Exponent $e(\mathcal{T})$ der Hauptreihe spielen, so daß auch für diese Aufgabe die in Abschnitt 3.5.2 aufgeworfene Fragestellung zur geeigneten Wahl der Hauptreihe für eine feste Gruppe relevant ist.

Kapitel 6

DFT-basierte Zerlegung der Gruppenalgebra

Eine diskrete Fouriertransformation D einer endlichen Gruppe G ist nach Definition 2.2.2 ein Algebrenisomorphismus von der Gruppenalgebra $\mathbb{C}G$ in eine Algebra von Blockdiagonalmatrizen. Die Urbilder dieser Blöcke unter D liefern eine direkte Summenzerlegung der Gruppenalgebra $\mathbb{C}G$ in minimale zweiseitige Ideale. In diesem Kapitel geht es um direkte Summenzerlegungen von $\mathbb{C}G$, die diese „Wedderburn-Zerlegung“ verfeinern. Mit Hilfe von Idempotenten der Gruppenalgebra und Partitionen der 1 in $\mathbb{C}G$ lassen sich die Zerlegungen elegant formulieren. In Abschnitt 6.1 führen wir die benötigten Begriffe ganz allgemein für Ringe A mit 1 ein. Je mehr Struktur diese Ringe haben, desto stärkere Aussagen lassen sich bezüglich der durch die Idempotenten definierten Summenzerlegungen treffen. In den folgenden Abschnitten nehmen wir eine „Top-Down“-Sichtweise ein und spezialisieren sukzessive den allgemeinen Fall über Algebren, halbeinfache Algebren, Gruppenalgebren bis hin zu Gruppenalgebren endlicher überauflösbarer Gruppen. Im letzteren Fall erhält man schließlich eine unter der G-Linksoperation invariante direkte Summenzerlegung von $\mathbb{C}G$ in eindimensionale Unterräume. Berücksichtigt man schließlich noch die Hilbertraumstruktur von $\mathbb{C}G$ bezüglich des Standardskalarprodukts, so entspricht diese Zerlegung, wie wir in Abschnitt 6.3 zeigen werden, gerade der verallgemeinerten Spektraldarstellung einer Funktion $f \in \mathbb{C}G$, die sich mit einem schnellen FFT-Algorithmus berechnen läßt. Abschnitt 6.4 rundet mit einigen Beispielen zu den Koordinatenfunktionen verallgemeinerter Spektraltransformationen das Kapitel ab.

6.1 Idempotente und Partition der Eins

Das Konzept der idempotenten Elemente und der Partitionen der Eins, welches wir in Abschnitt 6.1.1 einführen, macht in beliebigen Ringen mit Einselement Sinn. In Abschnitt 6.1.2 behandeln wir dann den Spezialfall halbeinfacher Algebren. Für Einzelheiten verweisen wir auf [21, 22].

6.1.1 Beliebige Ringe A mit Einselement

Sei also A zunächst als ein beliebiger Ring mit Einselement 1 vorausgesetzt. Ein Element $\epsilon \in A$ heißt ein *Idempotent*, falls $\epsilon^2 = \epsilon \neq 0$ gilt. Zwei Idempotente $\epsilon_1, \epsilon_2 \in A$ heißen *orthogonal*, falls $\epsilon_1\epsilon_2 = \epsilon_2\epsilon_1 = 0$. Sei

$$A = L_1 \oplus \ldots \oplus L_n \tag{6.1}$$

eine direkte Zerlegung von A in Linksideale L_i, $1 \leq i \leq n$. Zerlege $1 \in A$ und erhalte dadurch

$$1 = \epsilon_1 + \ldots + \epsilon_n, \quad \epsilon_i \in L_i. \tag{6.2}$$

Dann folgt leicht

$$L_i = A\epsilon_i, \quad \epsilon_i^2 = \epsilon_i, \quad \epsilon_i\epsilon_j = 0 \quad \text{für } 1 \leq i \neq j \leq n. \tag{6.3}$$

Mit anderen Worten, $(\epsilon_1, \ldots, \epsilon_n)$ ist eine Folge paarweise orthogonaler Idempotente von A. Diese sind durch die Zerlegung (6.1) eindeutig bestimmt (im allgemeinen gibt es natürlich viele solcher Zerlegungen von A). Umgekehrt definiert jede Folge $\vec{\epsilon} := (\epsilon_1, \ldots, \epsilon_n)$ paarweise orthogonaler Idempotente mit $\sum_i \epsilon_i = 1$, die wir im folgenden auch als *Partition der Eins* bezeichnen, eine direkte Summenzerlegung $A = \bigoplus_i A\epsilon_i$ in Linksideale. Die Menge $\mathcal{P}(A)$ aller Partitionen der Eins von A entspricht also eineindeutig der Menge aller direkten Summenzerlegungen von A in Linksideale.

Ist $\epsilon_1 = \epsilon' + \epsilon''$ mit orthogonalen Idempotenten ϵ' und ϵ'', $L_1 = A\epsilon_1$, dann folgt $L_1 = L_1\epsilon' \oplus L_1\epsilon''$. Die Summe ist direkt, da ϵ' durch Rechtsmultiplikation auf $L_1\epsilon'$ identisch operiert und $L_1\epsilon''$ auf Null abbildet. Ist umgekehrt $L_1 = L' \oplus L''$ eine direkte Summe von Linksidealen von A, dann gilt $\epsilon_1 = \epsilon' + \epsilon''$ mit orthogonalen Idempotenten $\epsilon' \in L'$ und $\epsilon'' \in L''$. Ein Idempotent $\epsilon \in A$ heißt *primitiv*, falls sich ϵ nicht als Summe zweier orthogonaler Idempotente ausdrücken läßt. Wir haben gerade gezeigt, daß ϵ genau dann primitiv ist, wenn das Linksideal $A\epsilon$ unzerlegbar ist. Völlig analoge Aussagen gelten für Rechtsideale.

Wir führen auf $\mathcal{P}(A)$ eine Halbordnung $\prec$ ein. Seien $\vec{\epsilon} = (\epsilon_1, \ldots \epsilon_n), \vec{\delta} = (\delta_1, \ldots \delta_m) \in \mathcal{P}(A)$, dann definiere

$$\vec{\epsilon} \prec \vec{\delta} \quad :\Longleftrightarrow \quad \forall k \in [1:n]\ \exists K \subset [1:m]: \quad \epsilon_k = \sum_{j \in K} \delta_j. \tag{6.4}$$

Idealtheoretisch drückt diese Halbordnung aus, daß eine direkte Summenzerlegung von A in Linksideale kleiner ist als eine zweite, wenn die zweite Zerlegung die erste verfeinert.

Beispiel 6.1.1 Sei A ein beliebiger Ring mit 1 und $\epsilon \in A$, $\epsilon \neq 1$, ein Idempotent. Dann ist $1 - \epsilon$ ein zu ϵ orthogonales Idempotent und $(\epsilon, 1 - \epsilon)$ eine Partition der Eins.

Beispiel 6.1.2 Sei $A = \mathbb{C} \oplus \mathbb{C}^{2\times 2}$, aufgefaßt als Unterring des Rings der 3×3-Matrizen über $\mathbb{C}$. Definiere

$$\epsilon_1 := \begin{bmatrix} 0 & & \\ & 2 & -1 \\ & 2 & -1 \end{bmatrix}, \quad \epsilon_2 = 1 - \epsilon_1 = \begin{bmatrix} 1 & & \\ & -1 & 1 \\ & -2 & 2 \end{bmatrix}$$

und

$$\delta_1 = \epsilon_1, \quad \delta_2 := \begin{bmatrix} 0 & & \\ & -1 & 1 \\ & -2 & 2 \end{bmatrix}, \quad \delta_3 = \begin{bmatrix} 1 & & \\ & 0 & 0 \\ & 0 & 0 \end{bmatrix}.$$

Dann sind $\vec{\epsilon} = (\epsilon_1, \epsilon_2)$ und $\vec{\delta} = (\delta_1, \delta_2, \delta_3)$ Partitionen der Eins und $\vec{\epsilon} \prec \vec{\delta}$. Weiterhin sind alle Idempotente von $\vec{\delta}$ primitiv. Das Linksideal $A\epsilon_2$ läßt sich als direkte Summe in die minimalen Linksideale $A\delta_2$ und $A\delta_3$ zerlegen, wobei $A\delta_2$ als $\mathbb{C}$-Vektorraum die Dimension 2 und $A\delta_3$ die Dimension 1 hat.

Ein Idempotent ϵ heißt *zentral*, wenn es mit jedem Ringelement vertauschbar ist. Analog zu oben nennen wir ein Idempotent *zentral-primitiv*, wenn es sich nicht als Summe zweier orthogonaler zentraler Idempotente ausdrücken läßt. Man kann nun leicht sehen, daß die zentralen Idempotente gerade den zweiseitigen Idealen in A entsprechen. Wir fassen diese Aussagen im folgenden Satz zusammen.

Satz 6.1.3 *Direkte Zerlegungen von Ringen mit Eins in Linksideale entsprechen den Partitionen der Eins in orthogonale Idempotente. Sei $A = L_1 \oplus \ldots \oplus L_n$ eine Zerlegung von A in eine direkte Summe von Linksidealen und $1 = \epsilon_1 + \ldots + \epsilon_n$ mit $\epsilon_i \in L_i$. Dann gilt:*

(1) *Die ϵ_i sind orthogonale Idempotente und $L_i = A\epsilon_i$.*

(2) *Ist L_i unzerlegbar, dann ist ϵ_i primitiv.*

(3) *Sind alle Linksideale L_i zweiseitig, dann sind die Idempotente ϵ_i zentral.*

(4) *Sind alle L_i zweiseitig und ist L_j als Ideal unzerlegbar, dann ist ϵ_j zentral-primitiv.*

Geht man umgekehrt von einer Partition der Eins aus, dann gelten die umgekehrten Aussagen für die durch die Idempotente definierten Linksideale. Darüber hinaus sind Partitionen der Eins in zentral-primitive orthogonale Idempotente (bis auf die Reihenfolge) eindeutig bestimmt.

Ist $\epsilon \in A$ ein Idempotent, dann ist offensichtlich $\epsilon A \epsilon \subset A$ ein Unterring von A. Im folgenden leiten wir eine Charakterisierung dieses Ringes mit Hilfe eines geeigneten Endomorphismenrings her. Zunächst sei bemerkt, daß jedes $f \in \mathrm{Hom}_A(A\epsilon, A)$ identisch zu dem durch Rechtsmultiplikation mit dem Element $a := f(\epsilon) \in A$ definierten Homomorphismus ist. Es gilt nämlich für alle $x \in A\epsilon$:

$$f(x) = f(x\epsilon) = xf(\epsilon) = xa. \tag{6.5}$$

Sei $\mathrm{End}_A(A\epsilon) := \mathrm{Hom}_A(A\epsilon, A\epsilon)$ der Endomorphismenring von $A\epsilon$. Dann ist $f \in \mathrm{End}_A(A\epsilon)$ nach (6.5) durch Rechtsmultiplikation mit $a = f(\epsilon)$ gegeben, und damit folgt für alle $x \in A\epsilon$ wegen $f(x) \in A\epsilon$:

$$f(x) = f(x)\epsilon = xa\epsilon = x\epsilon a\epsilon.$$

f ist also auch durch Rechtsmultiplikation mit $\epsilon a \epsilon$ gegeben. Hieraus ergibt sich nun leicht das folgende Lemma:

Lemma 6.1.4 *Für ein Idempotent ϵ eines beliebigen Ringes A mit 1 definiert die Abbildung $\mathrm{End}_A(A\epsilon) \to \epsilon A \epsilon$, $f \mapsto \epsilon f(\epsilon)\epsilon$ einen Anti-Ringisomorphismus.*

6.1.2 Halbeinfache Algebren

Wir konzentrieren uns nun auf den Fall halbeinfacher Algebren, zu denen unter anderem die für uns relevanten Gruppenalgebren über $\mathbb{C}$ gehören. Für Einzelheiten und Beweise der folgenden Aussagen verweisen wir auf [36]. Es bezeichne $\mathbb{K}$ einen Körper, und alle $\mathbb{K}$-Vektorräume seien als endlich-dimensional vorausgesetzt. Sei A eine $\mathbb{K}$-Algebra, also ein $\mathbb{K}$-Vektorraum, welcher ein Ring mit 1 ist, so daß $\forall c \in \mathbb{K}, x, y \in A : (cx)y = c(xy) = x(cy)$ gilt. Eine Algebra A heißt *halbeinfach*, wenn jedes Linksideal $L \subset A$ ein komplementäres Linksideal L' besitzt mit $A = L \oplus L'$. Es bezeichne V einen A-Linksmodul, also einen $\mathbb{K}$-Vektorraum, auf dem A von links operiert. V heißt *einfach*, falls 0 und V die einzigen Untermoduln sind. Das Lemma von Schur formuliert sich modultheoretisch wie folgt:

Lemma 6.1.5 (Schur) *Seien V, W einfache A-Linksmoduln, dann gelten:*

(1) *Sei $\varphi \in \mathrm{Hom}_A(V, W)$, $\varphi \neq 0$, dann existiert eine Inverses zu φ in $\mathrm{Hom}_A(W, V)$.*

(2) *$\mathrm{End}_A(V)$ ist eine Divisionsalgebra, d. h. alle $\varphi \neq 0$ sind invertierbar.*

(3) *Ist $\mathbb{K}$ algebraisch abgeschlossen, dann folgt $\mathrm{End}_A(V) = \{c \cdot \mathrm{Id}_V, c \in \mathbb{K}\}$.*

Sei M ein einfacher A-Modul, dann ist die *M-isotypische Komponente* $M(V)$ eines A-Linksmoduls V definiert als Summe aller Untermoduln $W \subset V$ mit $W \simeq M$. Ist M' ein zu M isomorpher A-Modul, so gilt offensichtlich $M'(V) = M(V)$. Läßt man A durch Linksmultiplikation auf sich selbst operieren, so wird A zu einem A-Linksmodul, dem sogenannten *regulären Modul*. Wir fassen die wichtigsten Eigenschaften im folgenden Satz zusammen.

Satz 6.1.6 *Sei A eine halbeinfache $\mathbb{K}$-Algebra.*

(1) *Jeder einfache A-Modul ist isomorph zu einem Untermodul von A.*

(2) *Es gibt bis auf Isomorphie nur endlich viele einfache A-Linksmoduln.*

(3) *Sei $M_1, \ldots, M_h$ eine Transversale einfacher A-Linksmoduln, dann ist die M_k-isotypische Komponente $A_k := M_k(A)$, $1 \leq k \leq h$, ein minimales zweiseitiges Ideal in A. Weiterhin gilt $A_k A_\ell = 0$ für $k \neq \ell$.*

(4) *Es gilt $A \simeq A_1 \oplus \ldots \oplus A_h$ (Wedderburn-Zerlegung). Diese Zerlegung in minimale zweiseitige Ideale ist eindeutig bis auf die Reihenfolge der Summanden.*

(5) *Jedes minimale Linksideal L von A ist in genau einem der A_k enthalten.*

(6) *Sei $1 = \epsilon_1 + \ldots + \epsilon_h$ die zur Wedderburn-Zerlegung gehörige Partition der Eins. Dann sind die ϵ_k orthogonale zentral-primitive Idempotente. Weiterhin ist A_k eine $\mathbb{K}$-Algebra mit Einselement ϵ_k.*

(7) *Jedes Element $a \in A_k$ definiert mittels Linksmultiplikation ein Element in $\mathrm{End}_{\mathbb{K}}(M_k)$. Ist $\mathbb{K}$ algebraisch abgeschlossen, dann definiert diese Zuordnung einen Algebrenisomorphismus von A_k auf $\mathrm{End}_{\mathbb{K}}(M_k)$.*

Bevor wir uns den Gruppenalgebren zuwenden, beschreiben wir zum Schluß dieses Abschnitts, wie sich mit Hilfe der Idempotente eindimensionale $\mathbb{K}$-Unterräume von A konstruieren lassen.

Lemma 6.1.7 *Sei A eine halbeinfache Algebra über einem algebraisch abgeschlossenen Körper $\mathbb{K}$. Dann gilt:*

(1) *Sei $\epsilon \in A$ ein primitives Idempotent, dann ist $\epsilon A \epsilon$ eine eindimensionale $\mathbb{K}$-Algebra.*

(2) *Seien $\epsilon, \delta \in A$ Idempotente mit $A\epsilon \simeq A\delta$. Dann existiert eine Einheit $u \in A$ mit $\epsilon = u\delta u^{-1}$.*

(3) *Seien $\epsilon, \delta \in A$ primitive Idempotente. Falls $A\epsilon \simeq A\delta$, dann ist $\delta A \epsilon$ ein eindimensionaler $\mathbb{K}$-Vektorraum, ansonsten gilt $\delta A \epsilon = 0$.*

Beweis: (1) folgt sofort aus Lemma 6.1.4 und dem Lemma von Schur ((3) von Lemma 6.1.5). Wir zeigen nun (2). Es sei $A\epsilon \simeq A\delta$. Aus der Isomorphie $A\epsilon \oplus A(1-\epsilon) \simeq A\delta \oplus A(1-\delta)$ von A-Linksmoduln folgt, daß auch $A(1-\epsilon) \simeq A(1-\delta)$ gilt. Da jeder Isomorphismus von A aufgefaßt als A-Linksmodul durch Rechtsmultiplikation mit einer Einheit $u \in A$ gegeben ist, folgt

$$A\epsilon \cdot u = A\delta, \quad A(1-\epsilon)\cdot u = A(1-\delta), \quad A\delta \cdot u^{-1} = A\epsilon, \quad A(1-\delta)\cdot u^{-1} = A(1-\epsilon).$$

Hieraus folgt $u\delta \cdot u^{-1} \in A\epsilon$ und $u(1-\delta)\cdot u^{-1} \in A(1-\epsilon)$. Aus

$$1 = u \cdot 1 \cdot u^{-1} = u \cdot (\delta + (1-\delta)) \cdot u^{-1} = u\delta \cdot u^{-1} + u(1-\delta)\cdot u^{-1}$$

folgt wegen der Eindeutigkeit der Zerlegung der 1 in Elemente aus $A\epsilon$ und $A(1-\epsilon)$ die Behauptung $u\delta u^{-1} = \epsilon$ von (2). Falls $A\epsilon \simeq A\delta$, ergibt sich (3) sofort aus (1) und (2). Im Fall $A\epsilon \not\simeq A\delta$ folgt aus Satz 6.1.6, daß $A\epsilon$ und $A\delta$ in verschiedenen isotypischen Komponenten von A liegen und sich damit annullieren. Daher gilt $\delta A\epsilon = 0$ und (3) ist bewiesen. □

6.2 Geometrischer Zugang zur Darstellungstheorie

In diesem Abschnitt wird die Darstellungstheorie endlicher auflösbarer Gruppen unter Verwendung von Idempotenten aus einer geometrischen Richtung beleuchtet. Hierzu leiten wir in Abschnitt 6.2.1 explizite Formeln für die primitiven Idempotente her und geben eine Zerlegung der Gruppenalgebra in eine direkte Summe eindimensionaler Unterräume an. In Abschnitt 6.2.2 untersuchen wir, wie sich die Idempotente und entsprechenden Zerlegungen als Bilder unter einer geeigneten DFT-Abbildung im Fourierbereich darstellen. Im Fall überauflösbarer Gruppen ist die Zerlegung invariant unter Konjugation mit Gruppenelementen, was unmittelbar die Existenz monomialer Darstellungen beweist. Dies ist Inhalt von Abschnitt 6.2.3.

6.2.1 Zerlegung der Gruppenalgebra mittels Idempotenten

Sei G eine endliche Gruppe mit h Konjugationsklassen und $\mathrm{Irr}(G) = \{D_1, \ldots, D_h\}$ eine Transversale irreduzibler Darstellungen. Dann ist die Menge der zugehörigen irreduziblen Charaktere $\mathrm{irr}(G) := \{\chi_D := \mathrm{Spur}(D), D \in \mathrm{Irr}(G)\}$ eine Untermenge der Klassenfunktionen $\mathrm{CF}(G, \mathbb{C})$. Für einen Charakter χ über dem Körper $\mathbb{C}$ gilt weiterhin $\overline{\chi(g)} = \chi(g^{-1})$, so daß sich das Skalarprodukt zweier Charaktere χ und ψ als $\langle \chi | \psi \rangle = |G|^{-1} \sum_{g \in G} \chi(g)\psi(g^{-1})$ schreiben läßt. Ein klassisches Resultat besagt, daß die Charaktere von $\mathrm{irr}(G)$ eine Orthonormalbasis von $\mathrm{CF}(G, \mathbb{C})$ bilden (siehe [52]):

$$\langle \chi_{D_k} | \chi_{D_\ell} \rangle \;=\; \delta_{k,\ell}. \tag{6.6}$$

Sei M_k ein einfacher Modul mit Charakter χ_{D_k}, $1 \leq k \leq h$. Im Fall der Gruppenalgebra $\mathbb{C}G$ läßt sich die Projektionsabbildung auf die M_k-isotypische Komponente explizit angeben. Sie ist durch Linksmultiplikation mit

$$e_k := e_{\chi_{D_k}} := \frac{\chi_{D_k}(1)}{|G|} \sum_{g \in G} \chi_{D_k}(g^{-1})g \;\in \mathbb{C}G \tag{6.7}$$

gegeben (für einen Beweis siehe Kapitel 2 von [52]). Nach dem Satz von Maschke ist $\mathbb{C}G$ eine halbeinfache Algebra (dies gilt allgemein für Körper $\mathbb{K}$ mit $\mathrm{char}(\mathbb{K}) \nmid |G|$), und damit ergibt sich aus Satz 6.1.6 das folgende Korollar.

Korollar 6.2.1 *Für die Gruppenalgebra $\mathbb{C}G$ einer endlichen Gruppe G mit h Konjugationsklassen gilt:*

(1) *Die in* (6.7) *definierten $e_1, \ldots, e_h$ sind gerade die zentral-primitiven Idempotente von $\mathbb{C}G$. Sie bilden als solche eine Basis des Zentrums von $\mathbb{C}G$.*

(2) *$1 = e_1 + \ldots + e_h$ ist eine Partition der Eins in orthogonale Idempotente.*

(3) *Die $\mathbb{C}Ge_k$, $1 \leq k \leq h$, sind die minimalen beidseitigen Ideale und $\mathbb{C}G = \oplus_{k=1}^{h} \mathbb{C}Ge_k$ (Wedderburn-Zerlegung).*

(4) *Jeder einfache Untermodul mit Charakter χ_{D_k} ist in $\mathbb{C}Ge_k$ enthalten.*

Im folgenden wollen wir für die auflösbaren Gruppen die isotypischen Komponenten weiter in minimale Linksideale aufspalten und die entsprechenden primitiven Idempotente angeben. Bezeichne also von nun an G eine endliche auflösbare Gruppe mit Kompositionsreihe $\mathcal{T} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_0 = \{1\})$. Wir fixieren einen irreduziblen Charakter $\chi \in \mathrm{irr}(G)$. Die Menge

$$\mathrm{W}(\chi) := \{(\chi_0, \ldots, \chi_n = \chi) \mid \chi_i \in \mathrm{Irr}(G_i),\ \langle \chi_i | \chi_{i+1} \rangle > 0\}$$

entspricht allen Wegen von der Wurzel zum Knoten χ des Charaktergraphen von G. Da der Charaktergraph von G multiplizitätenfrei ist (Satz 2.1.2 [Clifford]), folgt aus der Voraussetzung $\langle \chi_i | \chi_{i+1} \rangle > 0$ sofort $\langle \chi_i | \chi_{i+1} \rangle = 1$. Für jede Folge $w = (\chi_0, \ldots, \chi_n = \chi) \in \mathrm{W}(\chi)$ definieren wir das Element

$$e(w) := e_{\chi_0} \cdot \ldots \cdot e_{\chi_n} \in \mathbb{C}G,$$

welches aufgrund der Voraussetzung $\langle \chi_i | \chi_{i+1} \rangle = 1$ von Null verschieden ist. Wir fassen die Eigenschaften der $e(w)$ im folgenden Satz zusammen, dessen Beweis man in [16] findet.

Satz 6.2.2 *Sei G eine endliche auflösbare Gruppe. Dann gilt mit den vorherigen Bezeichnungen:*

(1) *Für $w \in W(\chi)$ ist $e(w)$ ein primitives Idempotent in $\mathbb{C}G$ und daher $\mathbb{C}Ge(w)$ ein einfacher $\mathbb{C}G$-Linksmodul mit Charakter χ. Die Dimension von $\mathbb{C}Ge(w)$ ist $\chi(1) = |W(\chi)|$.*

(2) *Es gilt $e_\chi = \sum_{w \in \mathrm{W}(\chi)} e(w)$.*

(3) *Die primitiven Idempotente $e(w)$ induzieren eine direkte Summenzerlegung von $\mathbb{C}G$ in minimale Linksideale $\mathbb{C}Ge(w)$:*

$$\mathbb{C}G = \bigoplus_{\chi \in \mathrm{irr}(G)} \quad \bigoplus_{w \in \mathrm{W}(\chi)} \mathbb{C}Ge(w).$$

(4) *Die Gruppenalgebra zerlegt sich in eindimensionale Unterräume der Form $e(v)\mathbb{C}Ge(w)$:*

$$\mathbb{C}G = \bigoplus_{\chi \in \mathrm{irr}(G)} \quad \bigoplus_{(v,w) \in \mathrm{W}(\chi)^2} e(v)\mathbb{C}Ge(w).$$

Die Eindimensionalität der Unterräume $e(v)\mathbb{C}Ge(w)$ ergibt sich aufgrund der Primitivität der Idempotente $e(v)$ und $e(w)$ ganz allgemein aus Lemma 6.1.7. Ein alternatives, darstellungstheoretisches Argument geht wie folgt: Sei $v = (\psi_0, \ldots, \psi_n = \chi) \in \mathrm{W}(\chi)$, dann ergeben die Faktoren von $e(v)$ eine isotypische Filtration von $\mathbb{C}Ge(w)$ der Form

$$e(v)\mathbb{C}Ge(w) = e_{\psi_0} \cdot (e_{\psi_1} \cdot (\ldots \cdot (e_{\psi_{n-1}} \mathbb{C}Ge(w)) \ldots))$$

und $e(v)\mathbb{C}Ge(w)$ ist wegen $\langle \psi_i | \psi_{i+1} \rangle = 1$ ein einfacher $\mathbb{C}G_0$-Modul und daher eindimensional.

6.2.2 Zerlegung der Gruppenalgebra im Fourierbereich

Seien die Notation wie im vorherigen Abschnitt und G als auflösbar vorausgesetzt. Im folgenden wollen wir untersuchen, wie die im vorherigen Abschnitt definierten Idempotente und die induzierten Zerlegungen von $\mathbb{C}G$ als Bilder unter der $\mathcal{T}$-adaptierten diskreten Fouriertransformation $D = \oplus_{k=1}^h D_k : \mathbb{C}G \xrightarrow{\simeq} \oplus_{k=1}^h \mathbb{C}^{d_k \times d_k}$, $D_k \in \mathrm{Irr}(G, \mathcal{T})$, im Fourierbereich der Blockdiagonalmatrizen aussehen. Hierfür benötigen wir das folgende Lemma, das sich unmittelbar aus dem Lemma von Schur (siehe Abschnitt 2.1) ergibt und in Abschnitt 2.5 von [52] bewiesen ist.

Lemma 6.2.3 *Sei $\phi \in \mathrm{CF}(G)$ eine Klassenfunktion und D eine irreduzible Darstellung von G vom Grad d und Charakter χ. Dann ist die durch $D_\phi := \sum_{g \in G} \phi(g) D(g)$ definierte lineare Abbildung eine* Homothetie, *d. h. ein skalares Vielfaches der Identität der Form $\lambda \cdot \mathrm{Id}_d$, $\lambda \in \mathbb{C}$. Der Skalar ist durch $\lambda = \frac{1}{d} \sum_{g \in G} \phi(g)\chi(g)$ gegeben.*

Sei $D_\ell \in \mathrm{Irr}(G,\mathcal{T})$ mit $d := \deg(D_\ell)$ und e_k wie in (6.7) definiert, dann folgt aus Lemma 6.2.3 sofort

$$D_\ell(e_k) = \frac{\chi_{D_k}(1)}{|G|} \sum_{g\in G} \chi_{D_k}(g^{-1}) D_\ell(g) = \lambda \cdot \mathrm{Id}_d\,, \quad \lambda = \frac{1}{|G|} \sum_{g\in G} \chi_{D_k}(g^{-1})\chi_{D_\ell}(g) = \langle \chi_{D_\ell} | \chi_{D_k} \rangle.$$

Aus den Orthogonalitätsrelationen (6.6) ergibt sich dann

$$D_\ell(e_k) = \delta_{k,\ell} \cdot \mathrm{Id}_d. \tag{6.8}$$

Sei $E_k \in \oplus_{k=1}^h \mathbb{C}^{d_k \times d_k}$ die Matrix, dessen k-ter Block die Identität ist und deren andere Blöcke alle identisch Null sind. Das Bild von e_k unter der DFT $D = \oplus_{k=1}^h D_k$ ist also $D(e_k) = E_k$. Dies folgt natürlich auch sofort aus der Tatsache, daß die E_k offensichtlich die zentral-primitiven Idempotente von $\oplus_{k=1}^h \mathbb{C}^{d_k \times d_k}$ sind und die DFT D als Algebrenisomorphismus zentral-primitive Idempotente auf ebensolche abbildet.

Sei $D_k \in \mathrm{Irr}(G,\mathcal{T})$ und $w = \{(\chi_0, \ldots, \chi_n = \chi_{D_k}) \in \mathrm{W}(\chi_{D_k})$. Aus der $\mathcal{T}$-Adaptiertheit von D_k folgt für $0 \le i \le n$, daß $D_k{\downarrow}G_i = \oplus_{\ell=1}^q F_\ell$, $q \in \mathbb{N}$, für geeignete $F_\ell \in \mathrm{Irr}(G_i, \mathcal{T}_i)$, wobei alle Darstellungen F_ℓ denselben Grad haben (siehe Satz 2.1.2 [Clifford]). Damit ergibt sich aus (6.8), daß $D_k(e_{\chi_i}) = \oplus F_\ell(e_{\chi_i})$ eine Blockdiagonalmatrix ist, wobei genau diejenigen Blöcke $F_\ell(e_{\chi_i})$ die Identität sind, für die $\chi_{F_\ell} = \chi_i$ gilt und alle anderen Blöcke identisch Null sind. Aus der Voraussetzung $\langle \chi_i | \chi_{i+1} \rangle = 1$ folgt hieraus durch Induktion, daß $D_k(e(w)) = D_k(e_{\chi_0}) \cdot \ldots \cdot D_k(e_{\chi_n})$ eine Diagonalmatrix mit genau einem von Null verschiedenen Eintrag 1 ist, dessen Position eindeutig durch w bestimmt ist. Allgemeiner können die Positionen der Einträge der darstellenden Matrizen von D_k durch Paare $(v,w) \in \mathrm{W}(\chi_{D_k})^2$ parametrisiert werden (es gilt $\deg(D_k) = |W(\chi)|$ nach (1) von Satz 6.2.2). Sei $E_{(k,v,w)} \in \oplus_{k=1}^h \mathbb{C}^{d_k \times d_k}$ die Matrix mit genau einem von Null verschiedenen Eintrag 1 im k-ten Block an der Position (v,w). Dann folgt für $(v,w) \in \mathrm{W}(\chi_{D_k})^2$:

$$D(e(v)\mathbb{C}Ge(w)) = E_{(k,v,v)} \left(\bigoplus_{k=1}^h \mathbb{C}^{d_k \times d_k} \right) E_{(k,w,w)} = \mathbb{C} \cdot E_{(k,v,w)}.$$

Wir fassen das Ergebnis in dem folgenden Satz zusammen.

Satz 6.2.4 *Sei G eine endliche auflösbare Gruppe und $D = \oplus_{k=1}^h D_k$ eine $\mathcal{T}$-adaptierte DFT. Mit den vorherigen Bezeichnungen gilt:*

(1) $D(e_k) = E_k$.

(2) $D(e(w)) = E_{(k,w,w)}$ *für* $w \in \mathrm{W}(\chi_{D_k})$.

(3) $D(e(v)\mathbb{C}Ge(w)) = \mathbb{C} \cdot E_{(k,v,w)}$ *für* $(v,w) \in \mathrm{W}(\chi_{D_k})^2$.

6.2.3 Invarianz der Zerlegung bei überauflösbaren Gruppen

In diesem Abschnitt wird gezeigt, daß im überauflösbaren Fall die in (4) von Satz 6.2.2 angegebene Zerlegung von $\mathbb{C}G$ in eindimensionale Unterräume invariant unter Linksmultiplikation von G ist. Dabei folgen wir den Ausführungen in [16], wo man auch weitere Einzelheiten findet.

Sei also G als endliche überauflösbare Gruppe vorausgesetzt. Wegen der Normalität von G_i in G für $0 \leq i \leq n$ operiert G via Konjugation auf $\mathrm{Irr}(G_i, \mathcal{T}_i)$ und $\mathrm{irr}(G_i, \mathcal{T}_i)$. Für $\chi \in \mathrm{irr}(G, \mathcal{T})$ erhält man daher via $g(\chi_0, \ldots, \chi_n) := (g\chi_0, \ldots, g\chi_n)$ eine G-Operation auf $W(\chi)$. Aus der im Abschnitt 2.4 angegebenen Version des Satzes von Clifford folgt sofort, daß G transitiv auf den irreduziblen Konstituenten von $\chi{\downarrow}G_{n-1}$ operiert. Da G_{n-1} trivial auf $\mathrm{Irr}(G_{n-1}, \mathcal{T}_{n-1})$ operiert, folgt aus einer Induktion nach n, daß G transitiv auf $W(\chi)$ operiert.

Sei $g \in G$, dann gilt $ge_{\chi_i}g^{-1} = e_{g\chi_i}$ für alle $\chi_i \in \mathrm{Irr}(G_i)$ und damit auch $ge(v)g^{-1} = e(gv)$ für alle $v \in W(\chi)$. Für alle $(v, w) \in W(\chi)^2$ folgt

$$ge(v)\mathbb{C}Ge(w) = e(gv)g\mathbb{C}Ge(w) = e(gv)\mathbb{C}Ge(w).$$

Mit anderen Worten, G operiert transitiv auf der Menge der Geraden $\{e(v)M | v \in W(\chi)\}$, wobei $M := \mathbb{C}Ge(w)$ für beliebiges, aber festes $w \in W(\chi)$ definiert ist. Wählt man eine beliebige Basis von M aus diesem Geradenbüschel, dann ist die zugehörige Matrixdarstellung monomial. Dies beweist die wohlbekannte Tatsache, daß überauflösbare Gruppen M-Gruppen sind.

Sei $U \leq G$ der Stabilisator von $w \in W(\chi)$ und L eine Transversale der Linksnebenklassen von U in G. Da $0 \neq ge(w) = e(gw)ge(w) \in e(gw)M$ für $g \in G$ gilt und G transitiv auf $W(\chi)$ operiert, ist die Menge $\{ge(w) | g \in L\}$ eine $\mathbb{C}$-Basis von M. Sei D_M die zugehörige monomiale Matrixdarstellung. Wir zeigen, daß D_M sogar e-monomial ist. Da U die Gerade $e(w)M = \mathbb{C}e(w)$ stabilisiert, existiert ein linearer Charakter λ von U, so daß $ue(w) = \lambda(u)e(w)$ für alle $u \in U$ gilt. Der lineare Charakter λ ist als eindimensionale Darstellung offensichtlich e-monomial. Sei $e_\lambda = |U|^{-1} \sum_{u \in U} \lambda(u^{-1})u \in \mathbb{C}U$ das zentral-primitive Idempotent zu λ. Dann gilt

$$e_\lambda e(w) = |U|^{-1} \sum_{u \in U} \lambda(u^{-1})ue(w) = |U|^{-1} \sum_{u \in U} \lambda(u^{-1})\lambda(u)e(w) = e(w).$$

Analog zeigt man $e(w) = e(w)e_\lambda$. Es folgt $\mathbb{C}Ge(w) \leq \mathbb{C}Ge_\lambda$ und daher $e_\lambda = ae(w)$ für ein geeignetes $a \in \mathbb{C}G$, also $e(w) = e_\lambda e(w) = ae(w)e(w) = ae(w) = e_\lambda$. Damit haben wir $D_M = \lambda{\uparrow}_L G$ gezeigt, woraus die e-Monomialität von D_M aus der von λ folgt.

6.3 Gruppenalgebra als Hilbertraum

In den bisherigen Ausführungen dieses Kapitels haben wir die Hilbertraumstruktur der Gruppenalgebra $\mathbb{C}G \simeq \mathbb{C}^{|G|}$, die vermöge des Standardskalarproduktes induziert wird, nicht berücksichtigt. In diesem Abschnitt untersuchen wir die Idempotente und die hergeleiteten Zerlegungen in Hinblick auf das Standardskalarprodukt. Wie aus Satz 6.2.4 ersichtlich ist, spielt dabei die inverse diskrete Fouriertransformation eine entscheidende Rolle. In Abschnitt 6.3.1

leiten wir für eine beliebige DFT $D = \oplus_{k=1}^{h} D_k$ eine Formel für die Inverse $\mathbf{D}^{-1}$ her. Handelt es sich bei den D_k um unitäre Darstellungen, so wird die DFT-Abbildung D bis auf geeignete Normierungen zu einer orthogonalen Transformation $\mathbb{C}G \to \oplus_{k=1}^{h} \mathbb{C}^{d_k \times d_k}$ bezüglich der Standardskalarprodukte, und die Idempotente $e(w)$ lassen sich unmittelbar aus der DFT-Matrix ablesen (siehe Abschnitt 6.3.2). Abschließend kommen wir in Abschnitt 6.3.3 auf die verallgemeinerten Spektraltransformationen zu sprechen, die uns dann auch als Sprungbrett zur Signalverarbeitung dienen.

Im folgenden verwenden wir das in der Signalverarbeitung übliche Skalarprodukt

$$(f \mid f') := \sum_{g \in G} f(g)\overline{f'(g)}, \quad f, f' \in \mathbb{C}G, \tag{6.9}$$

auf $\mathbb{C}G \simeq \mathbb{C}^{|G|}$, das sich von dem in (2.1) definierten Skalarprodukt nur um den Vorfaktor $|G|^{-1}$ unterscheidet.

6.3.1 Inverse der DFT-Matrix

Sei $\mathbf{D}$ eine DFT-Matrix einer endlichen Gruppe G. Die Inverse von $\mathbf{D}$ ist eng verwandt mit der transponierten Matrix $\mathbf{D}^T$. Dies folgt unmittelbar aus den *Schur-Relationen*, die im folgenden Satz formuliert sind (für einen Beweis siehe Abschnitt 5.2 von [15]).

Satz 6.3.1 (Schur-Relationen) *Seien $D_1, \ldots, D_h$ eine Transversale von irreduziblen Darstellungen von G vom Grad $d_1, \ldots, d_h$. Dann gelten für alle $1 \le k, k' \le h$ und $1 \le \mu, \nu \le d_k$ und $1 \le \mu', \nu' \le d_{k'}$ die folgenden Relationen:*

$$\sum_{g \in G} D_k(g)_{\mu\nu} \cdot D_{k'}(g^{-1})_{\mu'\nu'} = \delta_{kk'}\delta_{\mu\nu'}\delta_{\nu\mu'}\frac{|G|}{d_k}.$$

Wie in Abschnitt 2.2 beschrieben, werden die Spalten von $\mathbf{D}$ von den Elementen $g \in G$ und die Zeilen durch $\bigcup_{1<k<h}\{(k, \mu, \nu) | 1 \le \mu, \nu \le d_k\}$ parametrisiert, wobei (k, μ, ν) die Position (μ, ν) in D_k beschreibt. Die Zeile (k, μ, ν) ist durch die Koordinatenfunktionen

$$D_{k\mu\nu} \in \mathbb{C}G, \quad D_{k\mu\nu}(g) := D_k(g)_{\mu\nu} \tag{6.10}$$

beschrieben. Sei $N := |G|$. Wir definieren Permutationsmatrizen $P, Q \in \mathbb{C}^{N \times N}$, die bezüglich der Standardbasen der Inversion $(G \ni g \mapsto g^{-1})$ bzw. der Abbildung $(k, \mu, \nu) \mapsto (k, \nu, \mu)$ entsprechen. Offensichtlich gilt $P = P^T = P^{-1}$ bzw. $Q = Q^T = Q^{-1}$, und aus den Schur-Relationen folgt

$$\mathbf{D} \cdot P \cdot \mathbf{D}^T \cdot Q = \bigoplus_{k=1}^{h} \frac{|G|}{d_k} \cdot \mathrm{Id}_{d_k^2}. \tag{6.11}$$

Damit ergibt sich folgende Formel für die Inverse der DFT-Matrix:

$$\mathbf{D}^{-1} = P \cdot \mathbf{D}^T \cdot Q \cdot \bigoplus_{k=1}^{h} \frac{d_k}{|G|} \cdot \mathrm{Id}_{d_k^2}. \tag{6.12}$$

6.3.2 Unitäre Darstellungen

Eine Matrixdarstellung D von G heißt *unitär*, falls alle $D(g)$, $g \in G$, unitäre Matrizen sind. Man kann zeigen, daß jede Darstellung zu einer unitären Darstellung äquivalent ist. Wichtig ist für uns der folgende Fall:

Lemma 6.3.2 *Jede e-monomiale Matrix ist unitär. Insbesondere ist jede e-monomiale Darstellung unitär.*

Beweis: Sei ω eine primitive e-te Einheitswurzel und $A = \alpha \cdot \mathrm{diag}(\omega^{a_1}, \ldots, \omega^{a_N}) \in \mathrm{GL}(N, \mathbb{C})$ eine e-monomiale Matrix beschrieben wie in Lemma 4.2.1. Dann folgt

$$\overline{A}^T = \mathrm{diag}(\omega^{-a_1}, \ldots, \omega^{-a_N}) \cdot \alpha^{-1} = A^{-1}.$$

□

Sei im folgenden $D = \oplus_{k=1}^{h} D_k$ eine *unitäre DFT*, d. h. alle Darstellungen D_k seien unitär. Aus den Schur-Relationen folgt dann die Orthogonalität der in (6.10) definierten Koordinatenfunktionen:

$$\begin{aligned}
(D_{k\mu\nu} \mid D_{k'\mu'\nu'}) &= \sum_{g\in G} D_k(g)_{\mu\nu} \overline{D_{k'}(g)_{\mu'\nu'}} \\
&= \sum_{g\in G} D_k(g)_{\mu\nu} \overline{D_{k'}(g)^T_{\nu'\mu'}} \\
&= \sum_{g\in G} D_k(g)_{\mu\nu} D_{k'}(g^{-1})_{\nu'\mu'} \\
&= \delta_{kk'}\delta_{\mu\mu'}\delta_{\nu\nu'} \frac{|G|}{d_k}. \qquad (6.13)
\end{aligned}$$

Aus (6.11) folgt dann sofort

$$\overline{\mathbf{D}}^T = P\mathbf{D}^T Q \quad \text{und} \quad \mathbf{D}^{-1} = \overline{\mathbf{D}}^T \cdot \oplus_{k=1}^{h} \frac{d_k}{|G|} \cdot \mathrm{Id}_{d_k^2}. \qquad (6.14)$$

Sei $W := \oplus_k \sqrt{d_k/|G|} \cdot \mathrm{Id}_{d_k^2}$, dann ist die Matrix $W\mathbf{D}$ unitär:

$$W\mathbf{D} \cdot \overline{W\mathbf{D}}^T = W \cdot \mathbf{D} \cdot \overline{\mathbf{D}}^T \cdot W = W \cdot \sum_k \frac{|G|}{d_k} \mathrm{Id}_{d_k^2} \cdot W = \mathrm{Id}_N. \qquad (6.15)$$

Wir fassen im folgenden Satz zusammen, welche Folgerungen dies für die Idempotente und die Zerlegungen der Gruppenalgebra hat.

Satz 6.3.3 *Sei $D = \oplus_{k=1}^{h} D_k$ eine DFT einer endlichen auflösbaren Gruppe G mit unitären Darstellungen D_k und DFT-Matrix $\mathbf{D}$. Dann gilt:*

(1) *Die zentral-primitiven Idempotente e_k sind paarweise orthogonal bez. $(\cdot \mid \cdot)$. Die Wedderburn-Zerlegung $\mathbb{C}G = \oplus_{k=1}^{h} \mathbb{C}Ge_k$ ist eine orthogonale Summe von Unterräumen.*

(2) *Die Idempotente $e(w)$ sind paarweise orthogonal bez. $(\cdot \mid \cdot)$ und definieren eine orthogonale Summenzerlegung $\mathbb{C}G = \bigoplus_{\chi \in \mathrm{irr}(G)} \bigoplus_{w \in \mathrm{W}(\chi)} \mathbb{C}Ge(w)$.*

(3) *Der durch (k, v, w) bestimmte Spaltenvektor von $\mathbf{D}^{-1}$ aufgefaßt als Koeffizientenvektor eines Elements in $\mathbb{C}G$ bez. der Standardbasis erzeugt den eindimensionalen Unterraum $e(v)\mathbb{C}Ge(w)$, und die Zerlegung $\mathbb{C}G = \bigoplus_{\chi \in \mathrm{irr}(G)} \bigoplus_{(v,w) \in \mathrm{W}(\chi)^2} e(v)\mathbb{C}Ge(w)$ ist orthogonal.*

Insbesondere gelten diese Aussagen für jede $\mathcal{T}$-adaptierte DFT einer überauflösbaren Gruppe.

Beweis: Wegen Satz 6.2.2 sind nur noch die Orthogonalitätsaussagen zu beweisen. Es wurde in (6.15) gezeigt, daß $W\mathbf{D}$ und damit auch $(W\mathbf{D})^{-1}$ unitär ist. Da W eine Diagonalmatrix ist, folgt daraus die Orthogonalität der Spalten von $\mathbf{D}^{-1} = (W\mathbf{D})^{-1} \cdot W$ bez. $(\cdot \mid \cdot)$. Aus Satz 6.2.4 folgt, daß der Koeffizientenvektor von $e(w)$ bez. der Standardbasis in $\mathbb{C}G$ mit dem durch (k, w, w) bestimmten Spaltenvektor von $\mathbf{D}^{-1}$ übereinstimmt und der durch (k, v, w) bestimmte Spaltenvektor von $\mathbf{D}^{-1}$ aufgefaßt als Koeffizientenvektor eines Elements in $\mathbb{C}G$ bez. der Standardbasis den eindimensionalen Unterraum $e(v)\mathbb{C}Ge(w)$ erzeugt. Hieraus folgt sofort die Orthogonalität der $e(w)$ und die der eindimensionalen Unterräume $e(v)\mathbb{C}Ge(w)$. Da die Summenzerlegungen von $\mathbb{C}G$ in (2) und (3) Vergröberungen der Zerlegung von (4) sind und $e_k \in \mathbb{C}Ge_k$ gilt, folgen die Aussagen (1) bis (3). Ist G eine überauflösbare Gruppe, so ist jede $\mathcal{T}$-adaptierte Darstellung e-monomial und damit aufgrund von Lemma 6.3.2 unitär. □

6.3.3 Spektraldarstellung

Wir verwenden im folgenden die Notation der vorherigen Abschnitte. Dabei sei $D = \oplus_{k=1}^{h} D_k$ die DFT einer auflösbaren Gruppe G mit unitären Darstellungen D_k. Wie in (6.13) sind die normalisierten Koordinatenfunktionen

$$\Delta_{k\mu\nu} = \sqrt{d_k/|G|} \cdot D_{k\mu\nu}, \quad 1 \le k \le h,\ 1 \le \mu, \nu \le d_k, \tag{6.16}$$

eine weitere Orthonormalbasis von $\mathbb{C}G$. Die Darstellung einer Funktion $f \in \mathbb{C}G$ bezüglich dieser Orthonormalbasis

$$f = \sum_{k,\mu,\nu} (f \mid \Delta_{k\mu\nu}) \Delta_{k\mu\nu} \tag{6.17}$$

nennt man auch *Fourierdarstellung* von f bez. D. Die *Fourierkoeffizienten* $\hat{f}_{k\mu\nu} := (f \mid \Delta_{k\mu\nu})$ genügen

$$\hat{f}_{k\mu\nu} = \sqrt{\frac{d_k}{|G|}} \cdot \sum_{g \in G} f(g) \overline{D_k(g)_{\mu\nu}} = \sqrt{\frac{d_k}{|G|}} \cdot \overline{D_k(\overline{f})_{\mu\nu}} \tag{6.18}$$

und können daher über eine Auswertung der DFT D auf $\overline{f}$ und anschließender Konjugation und Skalierung mit $\sqrt{d_k/|G|}$ berechnet werden. Ist D eine $\mathcal{T}$-adaptierte DFT einer überauflösbaren Gruppe G, so können mit Hilfe des FFT-Algorithmus alle Fourierkoeffizienten mit $O(|G| \cdot \log |G|)$ Operationen bestimmt werden (siehe Satz 5.2.1).

6.4 Beispiele

In diesem Abschnitt geben wir einige Beispiele zu den verallgemeinerten Spektraltransformationen, welche u. a. die klassische Fouriertransformation (Abschnitt 6.4.2) sowie die Walsh-Hadamard-Transformation - oder auch Haar-Wavelet-Paket-Zerlegung (Abschnitt 6.4.3) - als einfache Spezialfälle umfassen. In den folgenden Abbildungen sind die Realteile und Imaginärteile der in (6.16) definierten Koordinatenfunktion $\Delta_{k\mu\nu} : G \to \mathbb{C}$ für einige Gruppen G dargestellt. Dabei wird die „Zeitachse“ durch die Gruppenelemente $g \in G$ parametrisiert, die lexikographisch in den Exponenten ihrer Normalform angeordnet sind (siehe Abschnitt 5.3)). Die Koordinatenfunktionen wiederum sind lexikographisch bezüglich der Tripel (k, μ, ν) angeordnet (siehe (2.4)). Zum Zwecke eines besseren optischen Eindrucks sind die Werte der jeweiligen Koordinatenfunktionen durch einen Polygonzug verbunden, so daß der Eindruck kontinuierlicher Funktionen entsteht. (Eigentlich handelt es sich bei den Graphen der diskreten Koordinatenfunktionen um diskrete Punktmengen.)

6.4.1 Symmetrische Gruppe S_3

Wir setzen unser Beispiel aus Abschnitt 5.4.1 fort. Die Inverse der dort angegebenen DFT-Matrix $\mathbf{D}$ der S_3 läßt sich mit Hilfe von (6.14) sofort angeben:

$$\mathbf{D}^{-1} = \overline{\mathbf{D}}^T \cdot \frac{1}{6} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} = \frac{1}{6} \cdot \begin{bmatrix} 1 & 1 & 2 & 0 & 0 & 2 \\ 1 & 1 & 2\omega^4 & 0 & 0 & 2\omega^2 \\ 1 & 1 & 2\omega^2 & 0 & 0 & 2\omega^4 \\ 1 & -1 & 0 & 2 & 2 & 0 \\ 1 & -1 & 0 & 2\omega^2 & 2\omega^4 & 0 \\ 1 & -1 & 0 & 2\omega^4 & 2\omega^2 & 0 \end{bmatrix}.$$

Hieraus können sofort die zentral-primitiven Idempotente e_1 aus der ersten, e_2 aus der zweiten und e_3 aus der Summe der dritten und sechsten Spalte abgelesen werden:

$$\begin{aligned} e_1 &= \tfrac{1}{6}(1 + g_1 + g_1^2 + g_2 + g_2 g_1 + g_2 g_1^2), \\ e_2 &= \tfrac{1}{6}(1 + g_1 + g_1^2 - g_2 - g_2 g_1 - g_2 g_1^2), \\ e_3 &= \tfrac{1}{6}(2 + 2\omega^4 g_1 + 2\omega^2 g_1^2 + 2 + 2\omega^2 g_1 + 2\omega^4 g_1^2) \;=\; \tfrac{1}{3}(2 - g_1 - g_1^2). \end{aligned}$$

Das Idempotent e_3 läßt sich weiter in orthogonale primitive Idempotente $e_3 = e' + e''$ zerlegen (e' und e'' sind natürlich nicht mehr zentral-primitiv), wobei

$$\begin{aligned} e' &= \tfrac{1}{3}(1 + \omega^4 g_1 + \omega^2 g_1^2), \\ e'' &= \tfrac{1}{3}(1 + \omega^2 g_1 + \omega^4 g_1^2). \end{aligned}$$

In Abbildung 6.1 sind schließlich die Graphen der Real- und Imaginärteile der Koordinatenfunktionen $\Delta_{k\mu\nu}$ dargestellt. Hierbei bezeichnet D_1 die triviale Darstellung, D_2 die durch das Signum definierte Darstellung und D_3 die in (3.10) definierte zweidimensionale Darstellung. Die Koordinatenfunktionen $\Delta_{1,1,1}$ bzw. $\Delta_{2,1,1}$ korrespondieren mit den Darstellungen D_1 bzw. D_2. Die vier Koordinatenfunktionen $\Delta_{3,1,1}, \Delta_{3,1,2}, \Delta_{3,2,1}$ und $\Delta_{3,2,2}$ entsprechen den vier Einträgen der darstellenden Matrizen von D_3. Die Gruppenelemente sind entsprechend ihrer lexikographischen Ordnung $1, g_1, g_1^2, g_2, g_2g_1, g_2g_1^2$ durchnumeriert. Die Numerierung der Darstellungen (lexikographisch in den Tripeln (k, μ, ν)) steht an der rechten Seite jeder Zeile. Analog sind auch die Abbildungen der folgenden Beispiele zu verstehen.

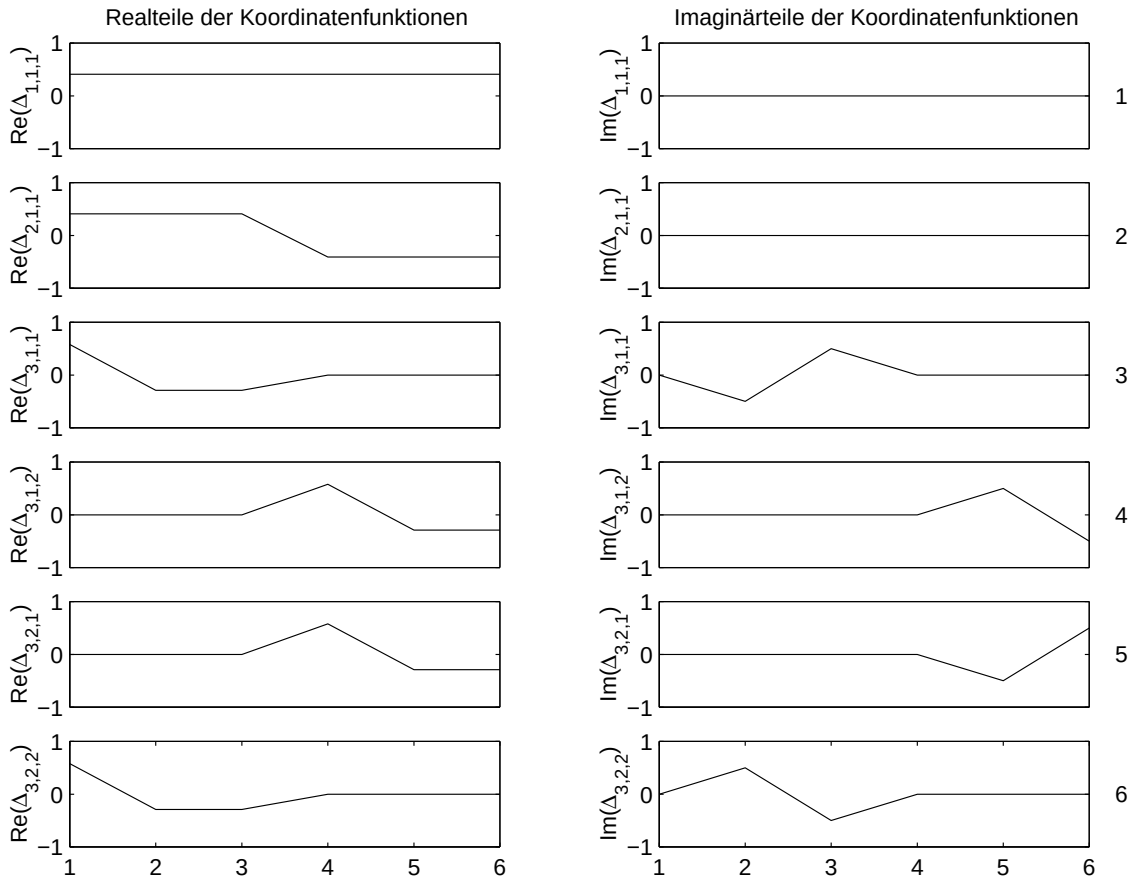


Abbildung 6.1: Koordinatenfunktionen $\Delta_{k\mu\nu}$ der symmetrischen Gruppe S_3

6.4.2 Klassische Spektralzerlegung: Zyklische Gruppe C_{37}

Im Fall der zyklischen Gruppe G der Ordnung N erhält man (bis auf eine eventuelle Umnumerierung der Gruppenelemente und der Darstellungen, vgl. Abschnitt 5.4.2) die klassische Spektralzerlegung. Bei den k-ten Koordinatenfunktionen $\Delta_{k,1,1}$, $k = 1, \ldots, N$, handelt es sich um die an den Stellen $t = 0, \ldots, N-1$ abgetasteten Exponentialfunktionen

$$t \mapsto \exp\left(t \cdot \frac{(k-1) \cdot 2\pi i}{N}\right).$$

In Abbildung 6.2 ist der Fall der zyklischen Gruppe C_{37} dargestellt. Hier hat die Hauptreihe die Länge $n = 1$ und die lexikographische Ordnung stimmt mit der durch die Größe des Exponenten des Erzeugers g_1 definierten Ordnung überein. Für hohe Frequenzen (ab der Nyquist-Frequenz $N/2$) erkennt man die Aliasingeffekte, d. h. große positive Frequenzen $N-k$ sind nicht von kleinen negativen Frequenzen $-k$ zu unterscheiden (siehe auch [65]).

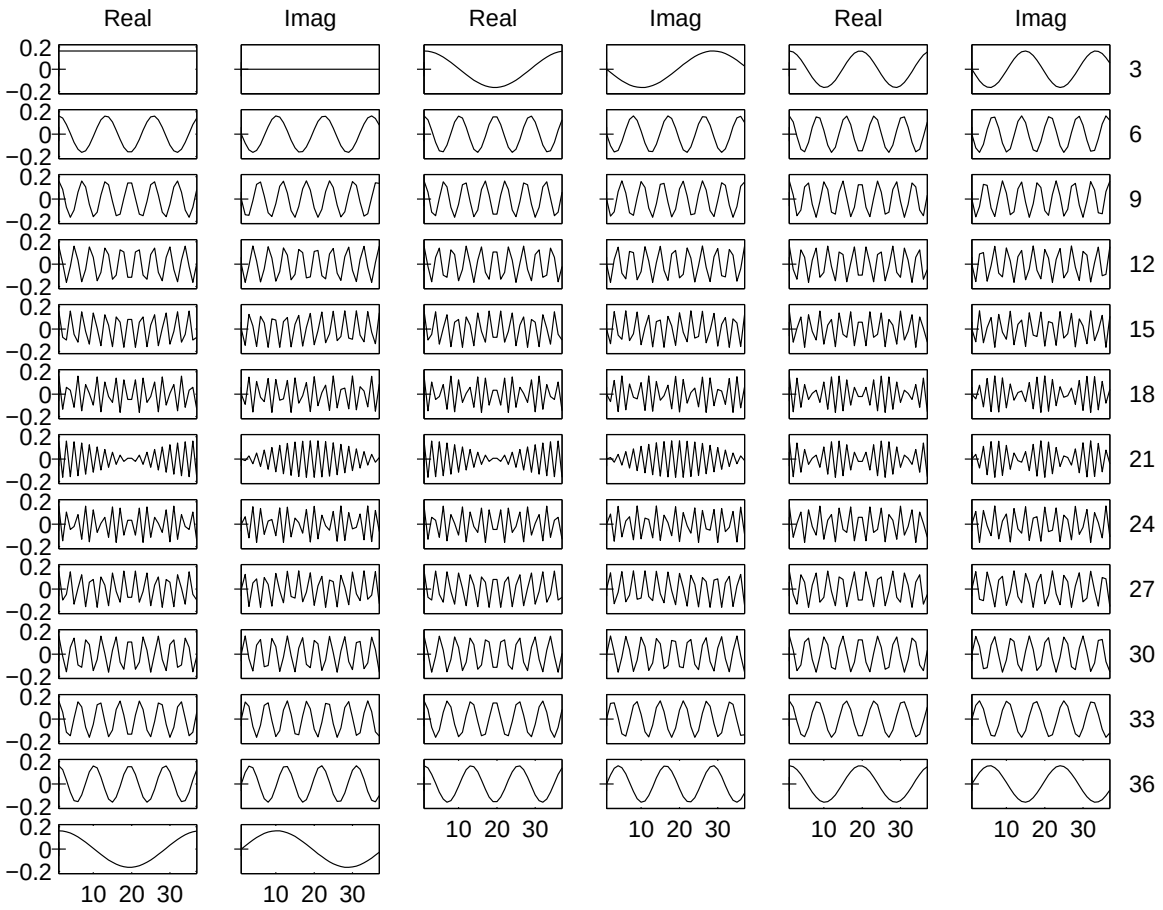


Abbildung 6.2: Koordinatenfunktionen $\Delta_{k\mu\nu}$ der zyklischen Gruppe C_{37}

6.4.3 Walsh-Hadamard-Transformation: Elementar-abelsche Gruppe $(C_2)^5$

Im Fall der elementar-abelschen Gruppen $G = (C_2)^n$, $n \in \mathbb{N}$, wird die diskrete Fouriertransformation auch *Walsh-Hadamard-Transformation* und die zugehörige Spektralzerlegung *Haar-Wavelet-Paket-Zerlegung* genannt. In Abbildung 6.3 sieht man die Haar-Wavelet-Pakete der Stufe $n = 5$, die den eindimensionalen Darstellungen, also den Charakteren der Gruppe $(C_2)^5$ entsprechen, welche rein reell sind. Für eine signaltheoretische Frequenz-Interpretation der entsprechenden Spektralkoeffizienten müssen die Koordinatenfunktionen umsortiert werden (nach der sogenannten Paley-Ordnung). Für weitere Einzelheiten verweisen wir auf [65].

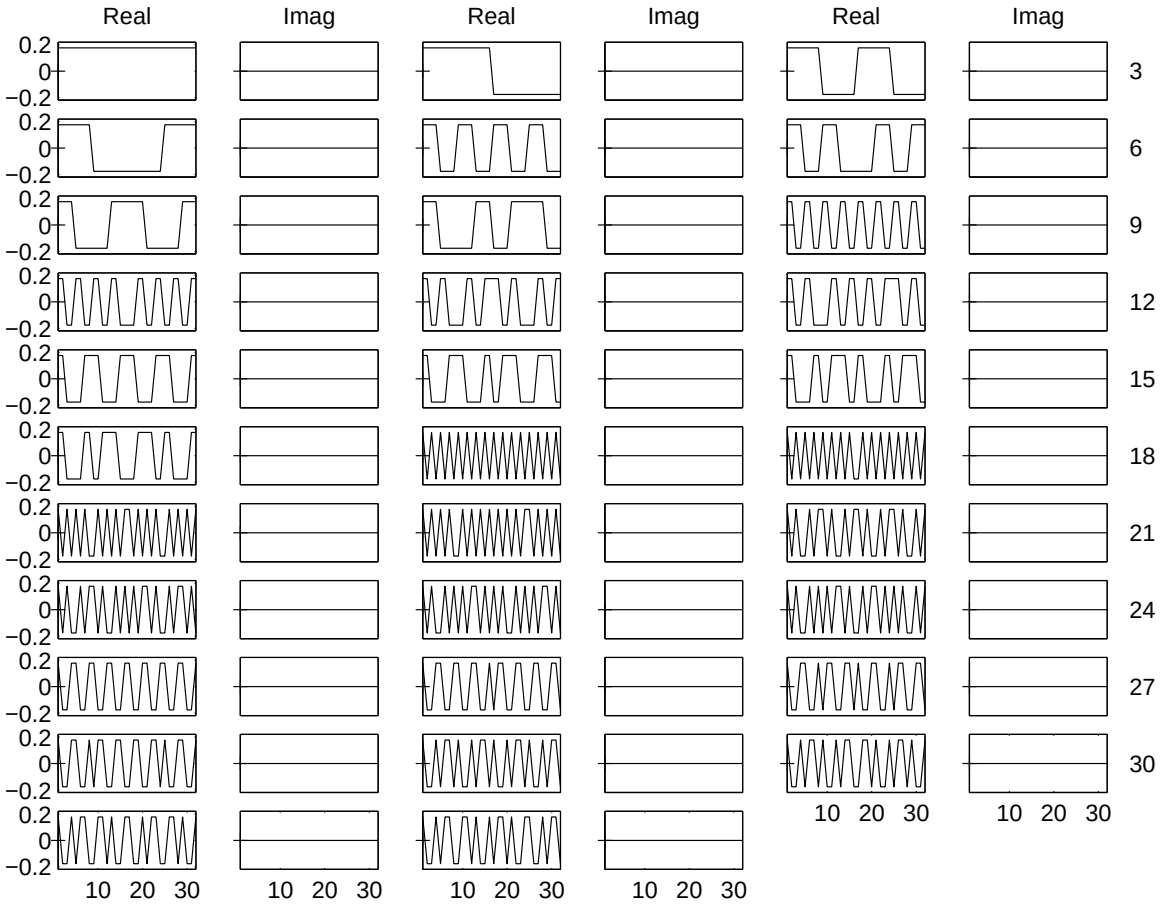


Abbildung 6.3: Koordinatenfunktionen der Haar-Wavelet-Paket-Zerlegung der elementar-abelschen Gruppe $(C_2)^5$.

6.4.4 Diedergruppe D_{19}

Es sei p eine Primzahl und D_p die Diedergruppe der Ordnung $N = 2 \cdot p$. Eine pc-Präsentation ist durch

$$D_p \simeq \langle g_1, g_2 \,|\, g_1^p = 1, g_2^2 = 1, [g_1, g_2] = g_1^{p-2} \rangle, \tag{6.19}$$

gegeben. Sei $D = \oplus_{k=1}^{h} D_k$ die durch den BC-Algorithmus konstruierte e-monomiale DFT. D_p hat genau zwei eindimensionale irreduzible Darstellungen, die mit D_1 (triviale Darstellung) und D_2 bezeichnet seien. Alle anderen irreduziblen Darstellungen D_k, $k = 3, \ldots, h = \frac{p-1}{2}$, sind zweidimensional. Da es sich bei diesen Darstellungen um monomiale Darstellungen handelt, haben die darstellenden Matrizen eine der folgenden Formen:

$$D_k(g_1^\ell) = \begin{bmatrix} * & 0 \\ 0 & * \end{bmatrix} \qquad \text{oder} \qquad D_k(g_2 g_1^\ell) = \begin{bmatrix} 0 & * \\ * & 0 \end{bmatrix} \tag{6.20}$$

für $\ell = 0, \ldots p-1$, $k \geq 3$. Die entsprechenden Koordinatenfunktionen verschwinden also entweder auf g_1^ℓ oder auf $g_2 g_1^\ell$ für $\ell = 0, \ldots, p-1$. Diese Trägereigenschaft der Koordinatenfunktionen wird auch durch Abbildung 6.4 am Beispiel der D_{19} illustriert. Wir werden noch einmal in den Beispielen von Abschnitt 7.3 hierauf zurückkommen.

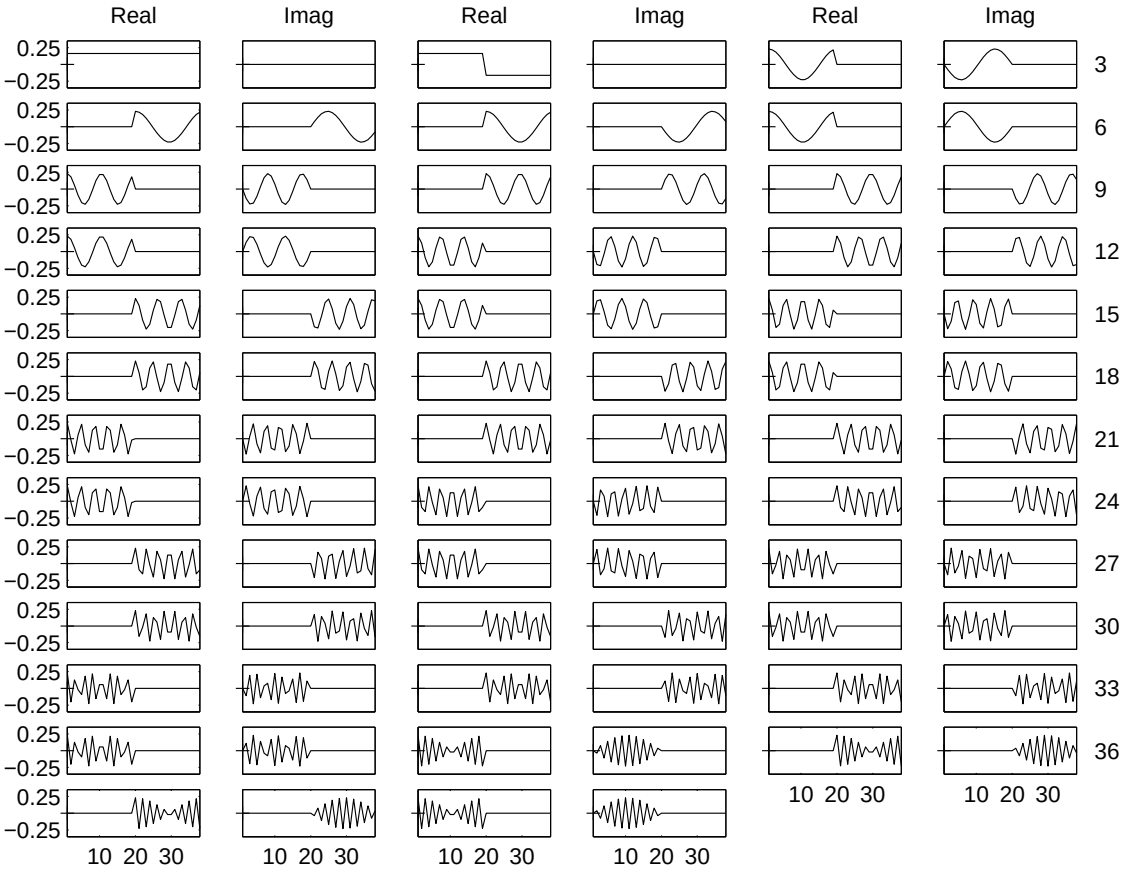


Abbildung 6.4: Koordinatenfunktionen $\Delta_{k\mu\nu}$ der Diedergruppe D_{19}.

6.4.5 2-Gruppe G_{128}

Zum Abschluß dieses Kapitels seien an dieser Stelle noch die 128 Koordinatenfunktionen zu den durch den BC-Algorithmus konstruierten Darstellungen der in Abschnitt 2.3 definierten Gruppe G_{128} abgebildet. Wir kommen im nächsten Kapitel noch einmal auf dieses Beispiel zurück.

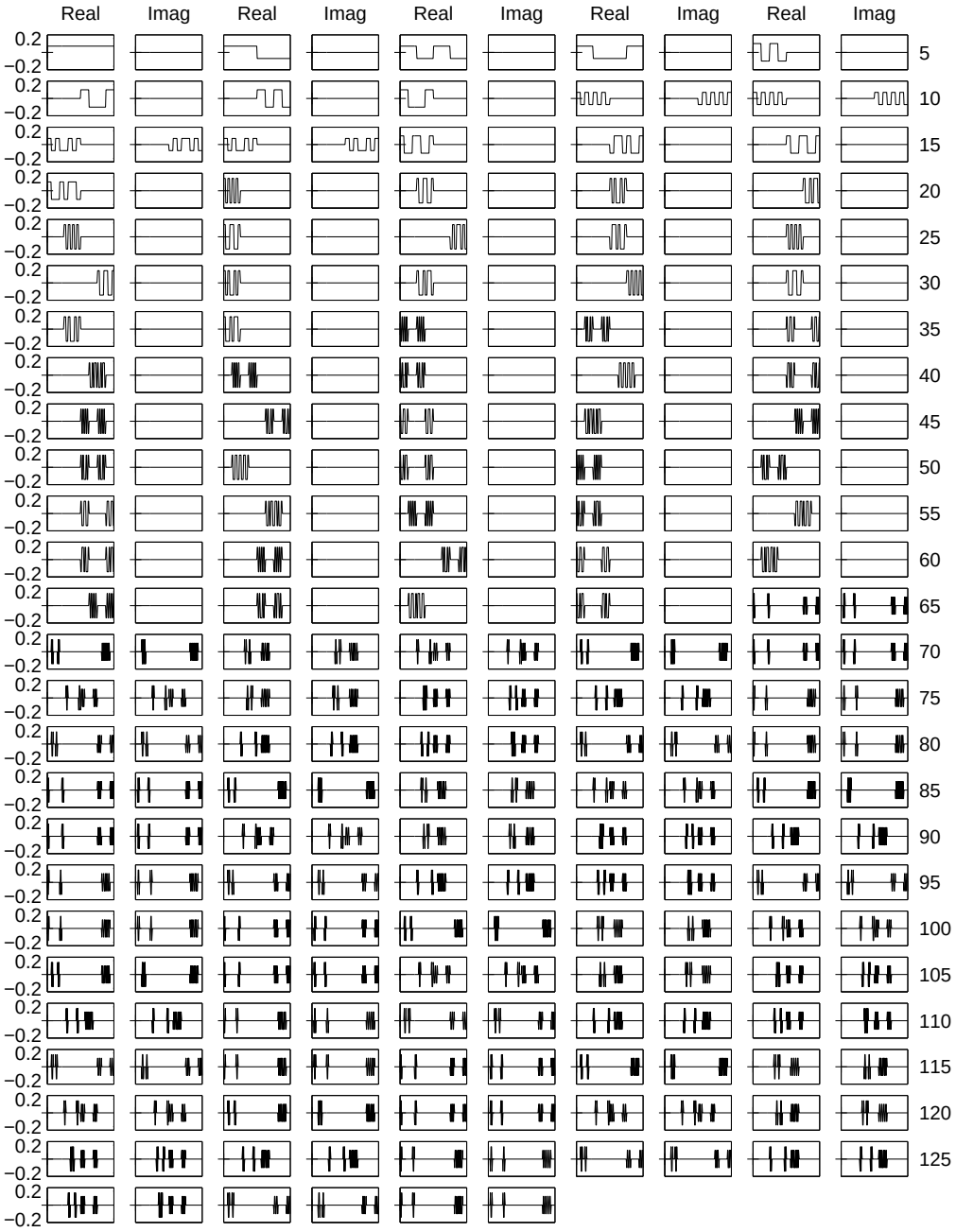


Abbildung 6.5: Koordinatenfunktionen $\Delta_{k\mu\nu}$ der Gruppe G_{128}.

Kapitel 7

DFT-basierte Signalverarbeitung

Momentan gibt es nur wenige Anwendungsgebiete verallgemeinerter diskreter Fouriertransformationen und deren Spektraltransformationen. Dies liegt unter anderem daran, daß es meist schwierig ist, die verallgemeinerten Spektralkoeffizienten sinnvoll zu interpretieren. Spezialfälle haben Anwendungen hauptsächlich im Bereich der Signalverarbeitung und Statistik gefunden (siehe u. a. [15, 24, 25]). Einen Überblick hierfür liefert auch der Artikel [51] von Rockmore (1995). In dem im letzten Jahr erschienenen Artikel [29] wird die Anwendung verallgemeinerter DFTs nicht-abelscher Gruppen (Kranzprodukte zyklischer Gruppen) im Bereich der Bildverarbeitung beschrieben. Diese DFTs umfassen viele bekannte Transformationen wie die Haar-Wavelet-Transformation und verschiedene Multikanal-Filterbankbäume als Spezialfälle.

Dieses Kapitel befaßt sich mit einigen Anwendungen der verallgemeinerten Spektralzerlegungen im Bereich der digitalen Signalverarbeitung zur Signalanalyse und Datenkompression. In Abschnitt 7.1 beschreiben wir, wie sich die Menge der verallgemeinerten Spektralkoeffizienten mit Hilfe der Multiskalenanalyse in Teilmengen zerlegen läßt, so daß die Koeffizienten jeder Teilmenge grob einem Frequenzband zugeordnet werden können. Eine Multiskalenanalyse entspricht dabei einer Folge von Partitionen der Eins der Gruppenalgebra $\mathbb{C}G$. In [15], Kapitel 11, deuten Clausen und Baum einen Algorithmus zur effizienten Datenkompression mittels verallgemeinerter DFTs an. Wir stellen nun in Abschnitt 7.2 ein Gesamtsystem vor, das diese Ideen erstmalig realisiert, und diskutieren einige der durchgeführten Experimente zur randomisierten Datenkompression.

7.1 Zerlegung von Signalräumen

In diesem Abschnitt widmen wir uns der Analyse des Signalraums $L^2(X) := \{f : X \to \mathbb{C}\}$, auf dessen endlicher Definitionsmenge X eine Gruppe G transitiv operiert. In Abschnitt 7.1.1 zeigen wir, wie solche Signale als Elemente der Gruppenalgebra $\mathbb{C}G$ aufgefaßt werden können. Untergruppenketten von G führen zu einer G-invarianten Filtration des Signalraums. Dies führt auf das Konzept der diskreten Multiskalenanalyse (MRA), das wir in Abschnitt 7.1.2 behandeln. Die durch die MRA definierte Zerlegung des Signalraums $L^2(X) = \oplus_{i=1}^{n+1} W_i$ erlaubt eine Frequenzinterpretation der G-invarianten Komponenten W_i (Abschnitt 7.1.3). In

Abschnitt 7.1.4 zeigen wir, daß die MRA-Zerlegung durch die Wedderburn-Zerlegung verfeinert wird. Dies erlaubt eine Zuordnung der Koordinatenfunktion zu den der MRA-Zerlegung korrespondierenden Frequenzbändern. Wir erläutern in Abschnitt 7.1.5 die Theorie am Beispiel der Gruppe G_{128}.

7.1.1 Signalräume

Der Anfang dieser Zusammenfassung folgt den Ausführungen von [29]. Nachfolgend wird beschrieben, wie sich die entsprechenden Aussagen mit Hilfe von Idempotenten formulieren lassen.

Sei X eine endliche Menge, bei der wir z. B. an eine diskrete Menge von Zeitpunkten oder an ein Gitter im Raum denken können. Ein *Signal* ist im folgenden eine komplexwertige Abbildung $f : X \to \mathbb{C}$. Der komplexe Vektorraum aller Signale wird zusammen mit dem analog zu 6.9 definierten Standardskalarprodukt $(\cdot\,|\,\cdot)$ zu einem Hilbertraum, den wir mit $L^2(X)$ bezeichnen. Fixiert man eine Ordnung auf X, so kann $L^2(X)$ mit dem $\mathbb{C}^N$, $N := |X|$, identifiziert werden.

Sei G eine endliche Gruppe, die auf X operiert. Für $g \in G$ und $x \in X$ schreiben wir dann gx für das Bild unter der von g definierten Permutation auf X und bezeichnen gx als das um g translatierte Element x. Mit der Operation von G auf X ist eine Darstellung von G auf $L^2(X)$ assoziiert, die durch

$$(g \cdot f)(x) := f(g^{-1}x), \quad \text{für alle } x \in X, \tag{7.1}$$

für $g \in G$ und $f \in L^2(X)$ definiert ist. Dies ist einfach die *Permutationsdarstellung* von G auf $L^2(X)$. Das Signal $g \cdot f$ heißt das *Translat* von f um g.

Sei $x_0 \in X$ ein ausgezeichneter Punkt und G_{x_0} der Stabilisator von x_0 in G. Operiert G transitiv auf X, dann gibt es für jedes $x \in X$ ein $g_x \in G$, so daß $g_x x_0 = x$ gilt. Die Zuordnung $x \mapsto g_x G_{x_0}$ definiert eine Bijektion, welche mit der G-Operation auf X und der durch Linksmultiplikation definierten G-Operation auf G/G_{x_0} verträglich ist. Wir können daher X und G/G_{x_0} als G-Mengen identifizieren. $L^2(X) \simeq L(G/G_{x_0})$ kann mit dem folgenden Unterraum von $L^2(G)$ identifiziert werden:

$$L^2(G, G_{x_0}) := \{f \in L^2(G) \mid f(gh) = f(g), \text{ für alle } g \in G, h \in G_{x_0}\} \tag{7.2}$$

sei der Unterraum des Signalraums $L^2(G)$, bestehend aus den auf den Linksnebenklassen von G_{x_0} konstanten Signalen. Die Abbildung

$$\pi_X : L^2(G) \to L^2(X), \quad \pi_X(f)(x) = \sum_{g \in g_x G_{x_0}} f(g), \tag{7.3}$$

$f \in L^2(G)$, $x \in X$, definiert eine G-lineare Abbildung. Anschaulich wird die Funktion $f \in L^2(G)$ durch π_X auf ihren Linksnebenklassen $g_x G_{x_0}$ gemittelt und mit $|G_{x_0}|$ multipliziert. Die Einschränkung von π_X auf $L^2(G, G_{x_0})$ ist ein G-linearer Isomorphismus mittels dem die Signalräume $L^2(X)$ und $L^2(G, G_{x_0})$ identifiziert werden. Zusammenfassend haben wir gezeigt, daß eine transitive Operation einer endlichen Gruppe G auf einer Menge X zu einer durch Linkstranslation definierten G-Operation auf der Menge G/H der Linksnebenklassen

äquivalent ist, wobei H der Stabilisator eines Punktes von X ist. Die durch die G-Operation auf $L^2(X) \simeq L^2(G,H)$ definierte Darstellung ist die nach G induzierte triviale Darstellung 1_H von H, also $1_H\uparrow G$ (siehe Abschnitt 3.3 von [52]).

Wir wollen nun beschreiben, wie sich die Projektionsabbildungen durch Idempotente ausdrücken lassen. Sei $H \subset G$ eine Untergruppe, dann ist $L^2(H) = \mathbb{C}H$ eine Unteralgebra von $L^2(G) = \mathbb{C}G$. Der analog zu (7.2) definierte Signalraum $L^2(G,H)$ ist ebenfalls eine Unteralgebra von $L^2(G)$, die invariant unter G-Links- und G-Rechtsmultiplikation ist. Man sieht leicht, daß

$$e_H := \frac{1}{|H|} \sum_{h \in H} h \in L^2(H) \tag{7.4}$$

ein Idempotent definiert. Die Faltung mit e_H in der Algebra $L^2(G)$ definiert eine Projektionsabbildung

$$\pi_{(G,H)} : L^2(G) \to L^2(G,H), \quad \pi_{(G,H)}(f) := f \cdot e_H. \tag{7.5}$$

Für $f \in L^2(G)$ und $x \in G$ zeigt die Rechnung

$$\begin{aligned} \pi_{(G,H)}(f)(x) &= (f \cdot e_H)(x) \\ &= \sum_{g \in G} f(g) \cdot e_H(g^{-1}x) \\ &= \frac{1}{|H|} \cdot \sum_{g \in xH} f(g), \end{aligned}$$

daß unter den angegebenen Identifikationen die Projektion π_X von (7.3) gerade der Projektion $\pi_{(G,H)}$ entspricht. Die Projektion $\pi_{(G,H)}$ wird auch *Radontransformation* genannt, die man sich als „Verklumpungsoperator“ vorstellen kann. Der Wert $\pi_{G,H}(f)(g_0)$ für ein $g_0 \in G$ ergibt sich als Mittelung der Werte $f(g)$ über die Elemente g der Linksnebenklasse g_0H. Analog zu (7.5) hat man für Untergruppen $H \subset K \subset G$ eine Radonabbildung $\pi : L^2(G,H) \to L^2(G,K)$. Wir verweisen auf [10] für eine allgemeine Definition der Radonabbildungen.

7.1.2 Multiskalenanalyse (MRA)

Ausgehend von zwei verschiedenen Richtungen - der reinen Mathematik und der Bildverarbeitung -, haben Yves Meyer und Stéphane Mallat die Theorie der sogenannten *Multiskalenanalyse* oder MRA (multiresolution analysis) begründet (siehe [41]). Eine MRA ermöglicht es, ein Signal in verschiedenen Auflösungen zu betrachten, die sich jeweils um einen Faktor zwei unterscheiden. Dabei kann die Differenz zweier solcher Auflösungen als Folge von Waveletkoeffizienten (einer diskreten orthogonalen Wavelettransformation) kodiert werden, für deren Berechnung es schnelle Algorithmen gibt. Für den späteren Vergleich mit einer diskreten Version der MRA geben wir zunächst die Definition der klassischen MRA an:

Definition 7.1.1 *Eine* Multiskalenanalyse (MRA) *des* $L^2(\mathbb{R})$ *ist eine aufsteigende Folge abgeschlossener Unterräume* $V_i \subset L^2(\mathbb{R})$ *mit den folgenden Eigenschaften:*

(1) $\{0\} \subset \ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \subset L^2(\mathbb{R})$ *mit* $\overline{\bigcup_{i\in\mathbb{Z}} V_i} = L^2(\mathbb{R})$ *und* $\bigcap_{i\in\mathbb{Z}} V_i = \{0\}$.

(2) *Die Räume V_i sind translationsinvariant: Sei $f \in V_i$, dann folgt $\forall t \in \mathbb{R} : f(\cdot - t) \in V_i$.*

(3) *Es existiert eine Funktion $\varphi \in L^2(\mathbb{R})$, die auch als* Skalierungsfunktion *bezeichnet wird, deren ganzzahligen Translate eine Orthonormalbasis von V_0 erzeugen, d. h. $V_0 = \overline{\mathrm{span}\{\varphi(\cdot - k) | k \in \mathbb{Z}\}}$.*

(4) *Der Raum V_i ist im folgenden Sinne eine skalierte Version des Raumes V_0: $f \in V_i \iff f(2^i \cdot) \in V_0$.*

Sei W_i das orthogonale Komplement von V_i in V_{i-1}. Das folgende wichtige Hauptergebnis stellt nun den Zusammenhang zwischen der Waveletttheorie und den MRAs her: Zu jeder MRA existiert ein Wavelet ψ, das sogenannte „Mutterwavelet", dessen translatierte und dilatierte Versionen $\psi_{i,k}(x) := 2^{-i/2}\psi(2^{-i}x - k)$, $k \in \mathbb{Z}$, für jedes $i \in \mathbb{Z}$ eine Orthonormalbasis des Raumes W_i bilden. Damit läßt sich der Signalraum $L^2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j$ bezüglich der Waveletbasis zerlegen. Wir verweisen auf [23] für weitere Einzelheiten.

Das Konzept der MRA wurde auf den Fall diskreter Signalräume übertragen (siehe z. B. [29, 57]). Hierbei wird der betrachtete Signalraum, auf dem eine Gruppe G von Symmetrien operiert, durch eine Kette G-invarianter Unterräume filtriert, die den verschiedenen Auflösungen entsprechen. Der „Differenzraum" zweier Auflösungen läßt sich weiter in irreduzible G-invariante Unterräume zerlegen. Dies entspricht der Zerlegung eines Signals in seine Spektralkomponenten bezüglich G (siehe z. B. [15, 25]).

Sei G zunächst eine beliebige endliche Gruppe, $H \subset G$ eine Untergruppe und $G = G_n \supset G_{n-1} \supset \ldots \supset G_0 = H$ eine Untergruppenkette relativ zu H. Analog zu (7.2) definieren wir die Signalräume $V_i := L^2(G, G_i) \subset L^2(G)$ für $0 \le i \le n$ und setzen $V_{n+1} := \{0\}$. Dann ist

$$\{0\} = V_{n+1} \subset V_n \subset V_{n-1} \subset \ldots \subset V_1 \subset V_0 = L^2(G, H) \tag{7.6}$$

eine G-invariante Filtration von $L^2(G, H)$. Der Raum V_n besteht aus den auf G konstanten Funktionen.

Die Radonabbildung $\pi_i : V_0 \to V_i$, $1 \le i \le n$, ist durch Rechtsmultiplikation in der Gruppenalgebra mit dem Idempotent $e_{G_i} := |G_i|^{-1} \sum_{g\in G_i} g$ definiert. Die Einschränkung von π_i auf V_{i-1}, die wir ebenfalls mit π_i bezeichnen, definiert ebenso eine G-lineare Projektionsabbildung auf V_i. Weiterhin sei π_{n+1} die Projektion auf den trivialen Raum $V_{n+1} = \{0\}$. Das G-invariante orthogonale Komplement zu $V_i \subset V_{i-1}$ ist durch den Kern $W_i := \{f \in V_{i-1} \mid \pi_i(f) = 0\}$ von π_i definiert. Damit gilt $V_{i-1} = V_i \oplus W_i$. Bezeichnen wir die Projektionen von V_{i-1} auf W_i mit ρ_i, dann gilt

$$\begin{array}{lccccccccc} L^2(G,H) = V_0 & \xrightarrow[\rho_1 \searrow]{\pi_1} & V_1 & \xrightarrow[\rho_2 \searrow]{\pi_2} & V_2 & \ldots & V_n & \xrightarrow[\rho_{n+1} \searrow]{\pi_{n+1}} & V_{n+1} = \{0\}, \\ & & W_1 & & W_2 & \ldots & & & W_{n+1} \end{array}$$

und $L^2(G, H) = V_0 = \oplus_{i=1}^{n+1} W_i$ ist eine orthogonale Zerlegung in G-invariante Unterräume, die wir im folgenden als *MRA-Zerlegung* bezeichnen. Analog zur Multiskalenanalyse für kontinuierliche Signalräume (siehe Definition 7.1.1) haben wir folgende diskrete Version:

Satz 7.1.2 *Sei G eine endliche Gruppe mit Untergruppenkette $G = G_n \supset \ldots \supset G_0 = H$, die auf dem Signalraum $L^2(X)$, $X := G/H$, operiert. Mit den zuvor eingeführten Bezeichnungen haben wir die folgende diskrete Version einer MRA für $L^2(X)$:*

(1) *Aufsteigende Kette von Unterräumen: $\{0\} = V_{n+1} \subset V_n \subset \ldots \subset V_1 \subset V_0 = L^2(X)$, also insbesondere $\bigcup_i V_i = L^2(X)$ und $\bigcap_i V_i = \{0\}$.*

(2) *Translationsinvarianz (G-Invarianz) der V_i: Sei $f \in V_i$, dann folgt $\forall g \in G : g \cdot f \in V_i$.*

(3) *Der Skalierungsfunktion entspricht das Idempotent $e_H := |H|^{-1} \sum_{h \in H} h \in V_0$. Sei $L \subset G$ ein Repräsentantensystem der Linksnebenklassen von H in G, dann ist $\{\ell \cdot e_H \mid \ell \in L\}$ eine Orthonormalbasis von V_0.*

(4) *Die Radonabbildung π_i stellt den Zusammenhang zwischen den verschiedenen Auflösungen her und entspricht in gewissem Sinne der Skalierbarkeitseigenschaft (4) von Definition 7.1.1: $f \in V_0 \Rightarrow \pi_i(f) \in V_i$.*

Aus Sicht der Idempotente ist die diskrete MRA nichts anderes als eine schrittweise Zerlegung der $1 \in \mathbb{C}G$ in paarweise orthogonale Idempotente. Der Zerlegung in Linksideale $V_0 = W_1 \oplus V_1$ entspricht die Partition $\vec{\epsilon}_1 := (1 - e_{G_1}, e_{G_1})$ (siehe Satz 6.1.3). Im nächsten Schritt wird nun V_1 weiter zerlegt, was mit einer weiteren Zerlegung von e_{G_1} korrespondiert. Man sieht leicht, daß zu der Zerlegung $V_0 = W_1 \oplus W_2 \oplus V_2$ die Partition $\vec{\epsilon}_2 := (1 - e_{G_1}, e_{G_1} - e_{G_2}, e_{G_2})$ gehört. Aus $e_{G_2} e_{G_1} = e_{G_2}$ folgt sofort, daß $\vec{\epsilon}_2$ tatsächlich eine Partition der Eins definiert. Induktiv erhalten wird damit eine Folge von Partitionen der Eins, die bezüglich der in (6.4) definierten Halbordnung $\prec$ eine linear angeordnete Kette bilden:

$$\vec{\epsilon}_1 \prec \vec{\epsilon}_2 \prec \ldots \prec \vec{\epsilon}_{n-1} \prec \vec{\epsilon}_n.$$

Dabei enspricht der feinsten Zerlegung $L^2(G, H) = V_0 = \oplus_{i=1}^{n+1} W_i$ die Partition

$$\vec{\epsilon}_n = (1 - e_{G_1},\ e_{G_1} - e_{G_2},\ \ldots,\ e_{G_{n-1}} - e_{G_n},\ e_{G_n}).$$

7.1.3 Frequenzinterpretation der MRA

Wie schon der Name der Multiskalenanalyse besagt, entsprechen die Räume V_i Auflösungen des Signalraums $L^2(X)$ auf verschiedenen „Skalen“. Mit wachsendem i ist die Projektion $\pi_i(f)$ eines Signals $f \in L^2(X)$ eine immer gröbere Approximation von f. Die i-te Auflösung $\pi_i(f)$ erhält man durch Mittelwertbildung über die Linksnebenklassen von G_i, was bei unserer Konvention für die Numerierung der Gruppenelemente (siehe Abschnitt 5.3) der Mittelwertbildung jeweils $|G_i|$ benachbarter Werte entspricht.

Das Idempotent e_{G_i} kann aus Sicht der Signaltheorie als *Filter* $h = (h_k)_{k \in \mathbb{Z}}$ interpretiert werden, dessen nicht verschwindende Filterkoeffizienten durch $h_0 = h_1 = \ldots = h_{|G_i|-1} = \frac{1}{|G_i|}$ gegeben sind. Die sogenannte *Frequenzantwort* ist durch die Fourierreihe

$$H(\omega) := \sum_{k \in \mathbb{Z}} h_k e^{-2\pi i k \omega} \tag{7.7}$$

definiert. Da die Filterkoeffizienten h_k reell sind, ist die durch den Betrag der Frequenzantwort definierte Funktion $\omega \mapsto |H(\omega)|$, die auch als *Amplitudengang* bezeichnet wird, eine gerade Funktion der Periode 1. Damit ist der Amplitudengang durch Angabe auf dem Bereich $[0, \frac{1}{2}]$ vollständig beschrieben. Entsprechend der Natur des Amplitudengangs werden Filter typischerweise in *Tiefpaßfilter* (tiefe Frequenzen werden durchgelassen, hohe gesperrt), *Hochpaßfilter* (hohe Frequenzen werden durchgelassen, tiefe gesperrt) und *Bandpaßfilter* (nur ein bestimmtes Frequenzband, das die Frequenz Null nicht enthält, wird durchgelassen, alles übrige gesperrt) eingeteilt (siehe Abbildung 7.1).

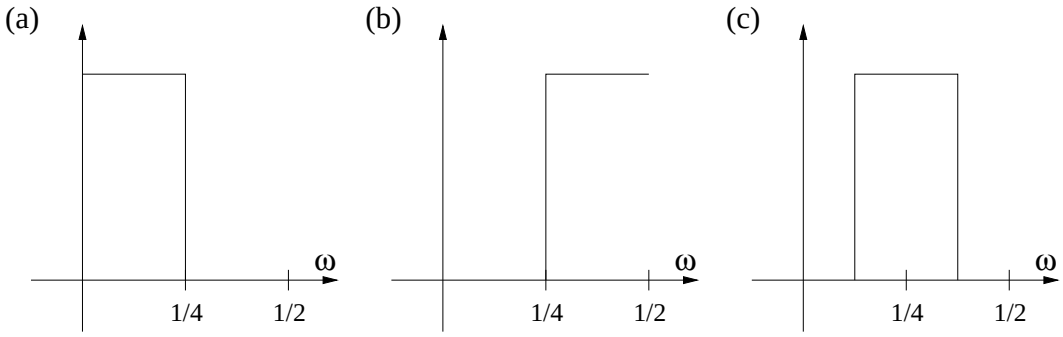


Abbildung 7.1: Amplitudengang eines idealen (a) Tief-, (b) Hoch- und (c) Bandpaßfilters.

Bei diskreten Signalen der Länge N lassen sich große positive Frequenzen nicht von kleinen negativen Frequenzen unterscheiden, d. h. die Zahl der Oszillationen der Exponentialfunktion der Frequenz ω stimmt mit der der Frequenz $N-\omega$ überein. Dieser Sachverhalt wird durch das *Nyquist-Theorem* ausgedrückt: Die maximal unterscheidbare Frequenz oder *Nyquist-Frequenz* für diese Abtastrate ist $N/2$ Oszillationen pro N Einheiten, also $\frac{1}{2}$. Wir wollen an dieser Stelle für weitere Einzelheiten auf die Standardliteratur der digitalen Signaltheorie (siehe z. B. [33, 64, 65]) verweisen. Untersucht man die Frequenzantworten der durch die Idempotente e_{G_i} und $e_{G_{i-1}} - e_{G_i}$ definierten Filter, so ergibt sich der folgende Satz:

Satz 7.1.3 *Sei $G = G_n \supset \ldots \supset G_0 = \{0\}$ eine Untergruppenkette und $f \in L^2(G)$. Die zugehörige diskrete MRA läßt sich auf folgende Weise signaltheoretisch interpretieren:*

- *Die Projektion $\pi_i(f)$ auf V_i, $1 \leq i \leq n$, entspricht einer Filterung von f mit dem durch e_{G_i} definierten Tiefpaßfilter. Der Durchlaßbereich des Filters (bei nicht-idealen Filtern nach Definition der Bereich, bei dem der Amplitudengang unter einem gewissen Schwellenwert liegt) entspricht dabei dem Intervall $\left[0, \frac{1}{2|G_i|}\right]$.*
- *Die Projektion $\rho_i(f)$ auf W_i, $1 \leq i \leq n$, entspricht einer Filterung von f mit dem durch $e_{G_{i-1}} - e_{G_i}$ definierten Bandpaßfilter. Hierbei setzen wir $e_{G_0} = 1$. Der Durchlaßbereich des Filters entspricht dabei dem Intervall $\left[\frac{1}{2|G_i|}, \frac{1}{2|G_{i-1}|}\right]$. Aus der Definition $V_{n+1} = \{0\}$ folgt weiterhin $\rho_{n+1}(f) = \pi_n(f)$.*

Die Abbildung 7.2 illustriert diese Frequenz-Interpretation am Beispiel der MRA bezüglich einer Kompositionsreihe einer 2-Gruppe. Es sind jeweils die Amplitudengänge der den Räumen V_i und W_i entprechenden Filter für $i = 1, \ldots, 4$ dargestellt.

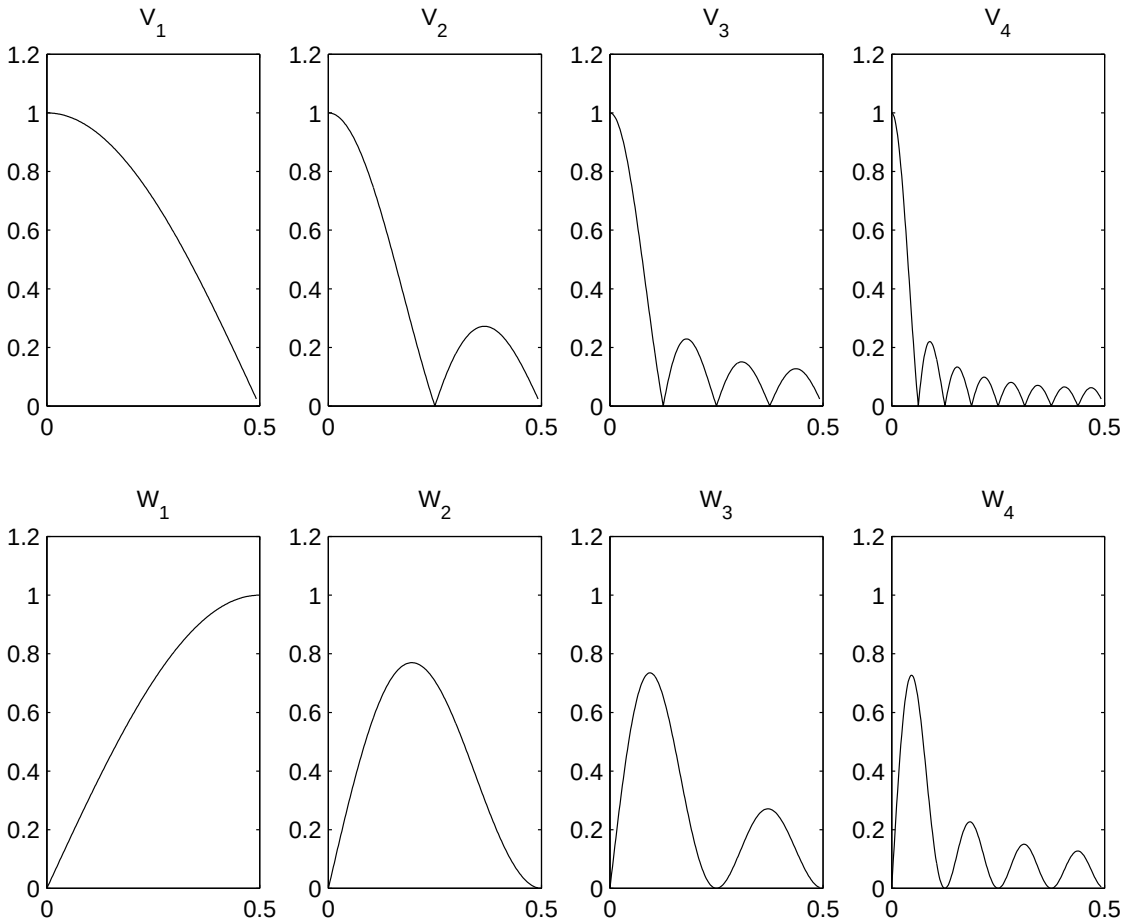


Abbildung 7.2: Amplitudengänge der mit den Räumen V_i und W_i korrespondierenden Filter.

In Abbildung 7.3 ist ein Beispiel für eine Multiskalenanalyse bezüglich einer Kompositionsreihe einer 2-Gruppe der Ordnung 256 dargestellt. Die MRA-Zerlegung ist in diesem Fall die Haar-Wavelet-Zerlegung. Beim Signal f handelt es sich um die Überlagerung zweier Sinusschwingungen der Frequenzen 5 bzw. 50 mit zwei zusätzlichen Impulsen an den Stellen 64 und 128.

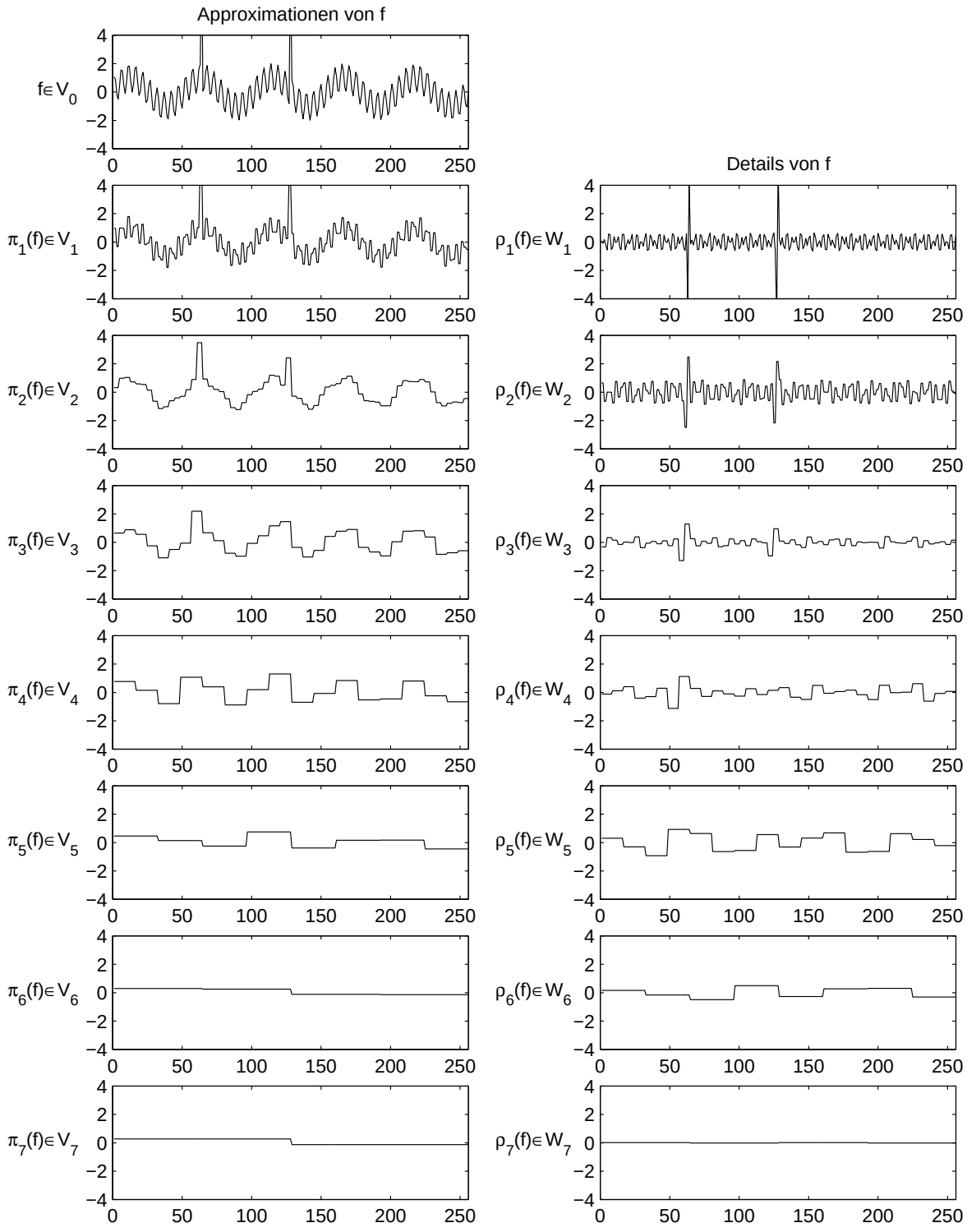


Abbildung 7.3: MRA bezüglich der Kompositionsreihe einer 2-Gruppe der Ordnung 256.

7.1.4 MRA und Spektralzerlegung

In diesem Abschnitt wollen wir die Idee der Multiskalenanalyse auf den Fall endlicher auflösbarer Gruppen G mit Kompositionsreihe $\mathcal{T} = (G = G_n \rhd \ldots \rhd G_1 \rhd G_0 = \{1\})$ anwenden. Wir zeigen zunächst, daß die Zerlegung von $L^2(G) = \mathbb{C}G$ in isotypische Komponenten (Wedderburn-Zerlegung, siehe Korollar 6.2.1) eine Verfeinerung der durch die MRA definierten Zerlegung ist. Dabei verwenden wir die in Abschnitt 6.2.1 eingeführte Notation.

Sei also $\mathrm{Irr}(G) = \{D_1, \ldots, D_h\}$ eine Transversale irreduzibler Darstellungen und $\mathrm{irr}(G)$ die Menge der zugehörigen irreduziblen Charaktere. Wir definieren für $0 \le i \le n$ die Teilmengen

$$\mathrm{Irr}(G, G_i) := \{D \in \mathrm{Irr}(G) \mid G_i \subset \ker(D)\} \tag{7.8}$$

der auf G_i trivialen Darstellungen von $\mathrm{Irr}(G)$ und die Teilmenge $\mathrm{irr}(G, G_i)$ der zugehörigen Charaktere. Auf G_i triviale Darstellungen von G entsprechen offensichtlich eineindeutig den Darstellungen von G/G_i und $\{\tilde{D} : gG_i \mapsto D(g) \mid D \in \mathrm{Irr}(G, G_i)\}$ definiert eine Transversale $\mathrm{Irr}(G/G_i)$ irreduzibler Darstellungen von G/G_i. Die Charaktere zu $\tilde{D}$ seien mit $\tilde{\chi}$ bezeichnet. Dann gilt für die durch (6.7) definierten Idempotente $e_{\tilde{\chi}}$ nach Korollar 6.2.1 die Gleichung

$$1 = \sum_{\tilde{\chi} \in \mathrm{irr}(G/G_i)} e_{\tilde{\chi}} = \sum_{\tilde{\chi}} \Big(\frac{\tilde{\chi}(1)}{|G/G_i|} \cdot \sum_{g \in G/G_i} \tilde{\chi}(g^{-1})g \Big) = \sum_{g \in G/G_i} \Big(\sum_{\tilde{\chi}} \frac{\tilde{\chi}(1)}{|G/G_i|} \cdot \tilde{\chi}(g^{-1}) \Big) g.$$

Hieraus folgt:

$$\sum_{\tilde{\chi} \in \mathrm{irr}(G/G_i)} \frac{\tilde{\chi}(1)}{|G/G_i|} \cdot \tilde{\chi}(g^{-1}) \;=\; \delta_{g,1}. \tag{7.9}$$

Es sei $L \subset G$ ein Vertretersystem der Linksnebenklassen von G_i in G mit $1 \in L$ als Vertreter der Linksnebenklasse G_i. Es bezeichne ℓG_i die Klasse von $\ell \in L$ in G/G_i. Da $\chi((\ell \cdot h)^{-1}) = \tilde{\chi}(\ell^{-1}G_i)$ für alle $\ell \in L$ und $h \in G_i$ gilt, folgt aus (7.9):

$$\begin{aligned}
\sum_{\chi \in \mathrm{irr}(G,G_i)} e_\chi &= \sum_{\chi \in \mathrm{irr}(G,G_i)} \Big(\frac{\chi(1)}{|G|} \cdot \sum_{g \in G} \chi(g^{-1})g \Big) \\
&= \sum_{\chi \in \mathrm{irr}(G,G_i)} \Big(\sum_{\ell \in L} \sum_{h \in G_i} \frac{1}{|G_i|} \cdot \frac{\chi(1)}{|G/G_i|} \cdot \chi(h^{-1} \cdot \ell^{-1})\ell \cdot h \Big) \\
&= \frac{1}{|G_i|} \cdot \sum_{h \in G_i} \Big(\sum_{\ell \in L} \Big[\sum_{\tilde{\chi} \in \mathrm{irr}(G/G_i)} \frac{\tilde{\chi}(1)}{|G/G_i|} \cdot \tilde{\chi}(\ell^{-1}G_i) \Big] \ell \Big) h \\
&\overset{(7.9)}{=} \frac{1}{|G_i|} \cdot \sum_{h \in G_i} \tilde{\ell} \cdot h \\
&= e_{G_i}
\end{aligned}$$

Mit anderen Worten, das Idempotent e_{G_i} läßt sich als Summe der Idempotente e_χ, $\chi \in \mathrm{irr}(G, G_i)$, schreiben. Da $\mathbb{C}Ge_\chi$ gerade die χ-isotypischen Komponenten der Wedderburn-Zerlegung definiert, ergibt sich der folgende Satz:

Satz 7.1.4 *Sei G eine endliche auflösbare Gruppe mit Kompositionsreihe $\mathcal{T}$. Dann ist die Zerlegung von $L^2(G) = \mathbb{C}G$ in seine isotypischen Komponenten (Wedderburn-Zerlegung) eine Verfeinerung der MRA-Zerlegung $L^2(G) = \oplus_{i=1}^{n+1} W_i$. Insbesondere gilt:*

$$\begin{aligned} e_{G_i} &= \sum_{\chi \in \mathrm{irr}(G,G_i)} e_\chi, && 0 \le i \le n, \\ V_i &= \bigoplus_{\chi \in \mathrm{irr}(G,G_i)} \mathbb{C}Ge_\chi, && 0 \le i \le n, \\ W_i &= \bigoplus_{\chi \in \mathrm{irr}(G,G_{i-1}) \setminus \mathrm{irr}(G,G_i)} \mathbb{C}Ge_\chi, && 1 \le i \le n, \\ W_{n+1} &= V_n. \end{aligned}$$

Wir kommen nun auf die in Abschnitt 6.3.3 definierte Spektraldarstellung zurück. Mit den dort eingeführten Bezeichnungen läßt sich ein Signal $f \in L^2(G)$ in seine Spektralkomponenten $f = \sum_{k\mu\nu} \hat{f}_{k\mu\nu} \Delta_{k\mu\nu}$ mit den Spektralkoeffizienten $\hat{f}_{k\mu\nu} = (f \,|\, \Delta_{k\mu\nu})$ zerlegen. Nach (3) von Satz 6.3.3 ist diese Zerlegung eine Verfeinerung der Wedderburn-Zerlegung und damit auch der MRA-Zerlegung. Mit anderen Worten, es gibt für jede Koordinatenfunktion $\Delta_{k\mu\nu}$ genau ein $i \in [1 : n+1]$, so daß $\Delta_{k\mu\nu} \in W_i$ gilt. Genauer gilt:

Korollar 7.1.5 *Sei $\Delta := \{\Delta_{k\mu\nu}, 1 \le k \le h,\ 1 \le \mu, \nu \le d_k\}$ die Menge aller Koordinatenfunktionen zu einer DFT $D = \oplus_{k=1}^{h} D_k$. Dann definiert*

$$\Delta^i := \{\Delta_{k\mu\nu} \in \mathrm{Irr}(G, G_{i-1}) \setminus \mathrm{Irr}(G, G_i)\}, \quad 1 \le i \le n+1,$$

eine Partition von Δ. Weiterhin bilden die Koordinatenfunktionen von Δ^i eine Orthonormalbasis für W_i.

Ist $f \in L^2(G)$ ein Signal, so ist die Größe des Spektralkoeffizienten $\hat{f}_{k\mu\nu}$ ein Maß dafür, mit welcher Energie die Koordinatenfunktion $\Delta_{k\mu\nu}$ in f „enthalten“ ist. Daher kann man aus den Spektralkoeffizienten zu den Koordinatenfunktionen von Δ^i ablesen, wie sich die Energie des Signals f auf die zu W_i korrespondierenden Frequenzbänder verteilt.

Abschließend sei angemerkt, daß die MRA-Zerlegung nur von den Ordnungen der Gruppen G_i, also dem Tupel $(|G_n|, |G_{n-1}|, \ldots, |G_1|)$, abhängt. So stimmen z. B. für p-Gruppen fester Ordnung die MRA-Zerlegungen bezüglich beliebiger Hauptreihen überein. Erst bei der Wedderburn-Zerlegung oder Spektralzerlegung treten dann die gruppenspezifischen Unterschiede auf.

7.1.5 Beispiel: 2-Gruppe G_{128}

Wir wollen die Überlegungen dieses Abschnitts am Beispiel der in Abschnitt 2.3 definierten Gruppe G_{128} veranschaulichen. Die Knoten der obersten Stufe des Charaktergraphen in Abbildung 7.4 entsprechen den irreduziblen Darstellungen in $\mathrm{Irr}(G_{128})$. Die Menge Δ^i der Koordinatenfunktionen läßt sich im Charaktergraph leicht anschaulich identifizieren: Die entsprechenden Darstellungen sind gerade diejenigen, die sich über der trivialen Darstellung von G_{i-1}, aber nicht über der trivialen Darstellung von G_i befinden. In Abbildung 7.4 sind dies gerade die Koordinatenfunktionen zu den Darstellungen der obersten Stufe 7, deren Einschänkung auf G_i die Darstellung $D_{i,1}$ enthalten.

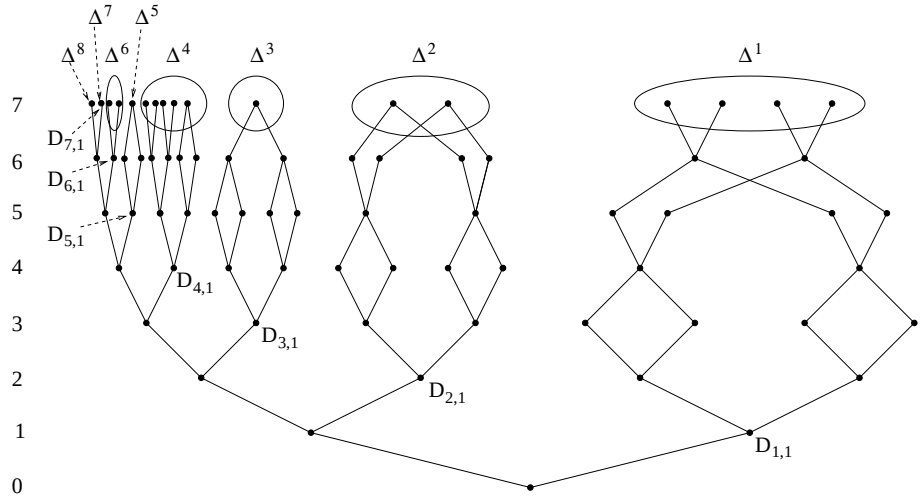


Abbildung 7.4: Partitionierung der Darstellungen von G_{128} gemäß der MRA.

In Abbildung 7.5 sind gemäß der Partitionierung $\Delta = \cup_{i=1}^{8} \Delta^i$ alle Koordinatenfunktionen von G_{128} dargestellt. Aus Gründen der Übersichtlichkeit wurden nur die Realteile der Koordinatenfunktionen abgebildet (vgl. mit Abbildung 6.5, wo sowohl die Real- also auch Imaginärteile dargestellt sind). Nach Satz 7.1.3 können die Koordinatenfunktionen von Δ^i, $1 \leq i \leq 7$, dem Frequenzband $\left[\frac{1}{2|G_i|}, \frac{1}{2|G_{i-1}|}\right]$ und die von Δ^8 dem Frequenzband $\left[0, \frac{1}{2|G_7|}\right]$ zugeordnet werden. Mit fallendem i beinhalten die entsprechenden Frequenzbänder immer größere Frequenzen. Diese Interpretation wird auch durch Abbildung 7.5 bestätigt. Für fallendes i werden die Oszillationen der Koordinatenfunktionen immer stärker.

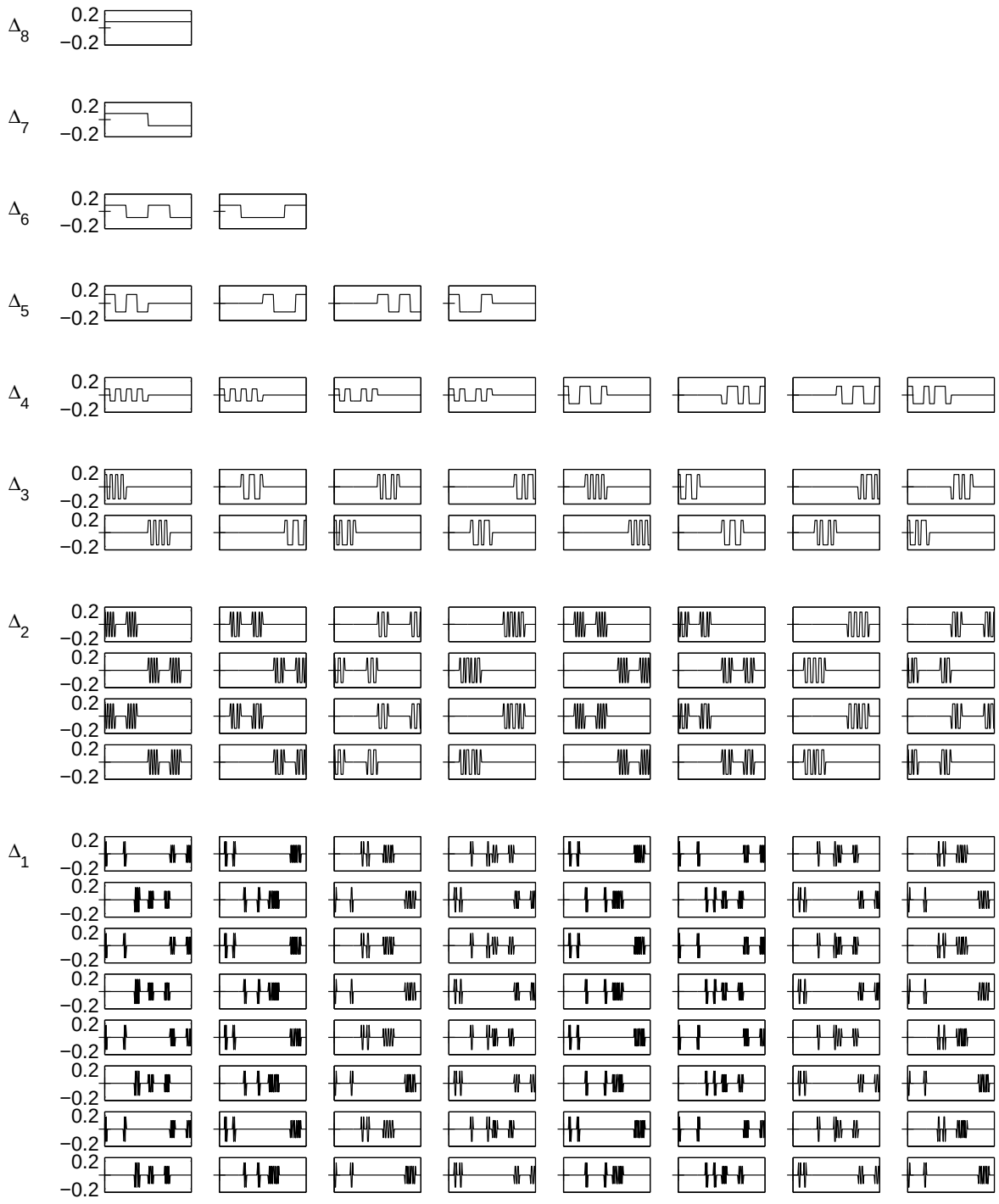


Abbildung 7.5: Realteile der in W_i enthaltenen Koordinatenfunktionen von G_{128}.

7.2 Datenkompression

Die grundlegende Idee der Fourieranalyse ist, daß die Spektraldarstellung einer Funktion Symmetrien offenbaren kann, die im Definitionsbereich des Signals („Zeitbereich“) versteckt sind. Wird das Fourierspektrum von einigen wenigen großen Koeffizienten dominiert, dann ist das Signal f annähernd eine Linearkombination der entsprechenden Koordinatenfunktionen. Diese Konzentration der Signalenergie kann z. B. für Kompressionszwecke ausgenutzt werden. In diesem Abschnitt besprechen wir, wie auf solche Weise verallgemeinerte Spektraltransformationen zur effizienten Datenkompression eingesetzt werden können. Grundzüge der Ideen wurden von Clausen und Baum in Kapitel 11 von [15] dargestellt. Wir stellen ein Gesamtsystem vor, das diese Ideen erstmalig realisiert.

7.2.1 Thresholding

Angenommen, wir wollen einen beliebigen komplexen Datenvektor $f \in \mathbb{C}^N$ der Länge $N \in \mathbb{N}$ komprimieren. Hierbei nehmen wir eine kleine kontrollierbare Störung der Daten in Kauf, die durch eine Genauigkeitsschranke vorgegeben ist. Folgende auf einer Spektralanalyse beruhende Methode zur Datenkompression ist denkbar:

(1) Wähle eine endliche Gruppe G der Ordnung N.

(2) Numeriere die Elemente von G, so daß der Datenvektor f als Element der Gruppenalgebra $\mathbb{C}G$ aufgefaßt werden kann.

(3) Berechne die verallgemeinerten Spektralkoeffizienten $\hat{f}_{k\mu\nu}$ von f bezüglich G.

(4) Speichere nur die großen Spektralkoeffizienten mit ihren jeweiligen Positionen. Die kleineren Spektralkoeffizienten werden einfach weggelassen.

Ist die Energie von f in wenigen großen Spektralkoeffizienten konzentriert, so können auf diese Weise hohe Kompressionsraten erzielt werden. Sei I die Indexmenge für die abgespeicherten Spektralkoeffizienten, dann ist

$$\tilde{f} := \sum_{(k\mu\nu)\in I} \hat{f}_{k\mu\nu}\Delta_{k\mu\nu} \tag{7.10}$$

die komprimierte, approximative Version des Originalsignals f. Der Fehler bezüglich der euklidischen Norm $\|\cdot\|_2$ ergibt sich wegen der Orthogonalität der Spektraltransformation aus der Formel von Plancherel:

$$\|f-\tilde{f}\|_2^2 \;=\; (f-\tilde{f}\,|\,f-\tilde{f}) \;=\; \sum_{(k\mu\nu)\notin I} |\hat{f}_{k\mu\nu}|^2. \tag{7.11}$$

Das Verfahren, bei dem Koeffizienten im Transformationsbereich entfernt werden, deren Betrag unter einem festgelegten *Schwellenwert* (Threshold) liegen, nennt man auch *Thresholding*. Es gibt verschiedene Strategien, die Anzahl der zu verbleibenden Spektralkoeffizienten zu bestimmen:

(1) Wähle eine globale Schranke δ und führe damit das Thresholding durch. Entferne also alle Spektralkoeffizienten $\hat{f}_{k\mu\nu}$ mit $|\hat{f}_{k\mu\nu}| \leq \delta$.

(2) Lege eine Kompressionsrate fest und wähle entsprechend dieser Rate die betragsmäßig größten Spektralkoeffizienten aus. Bei einer Kompressionsrate von 1 : 10 werden also 10% Prozent der betragsmäßig größten Koeffizienten ausgewählt und 90% der Koeffizienten auf Null gesetzt.

(3) Wähle bezüglich der euklidischen Norm eine Genauigkeitsschranke, innerhalb der das komprimierte Signal $\tilde{f}$ relativ zu f liegen soll. Die relative $\|\cdot\|_2$-Genauigkeit wird im folgenden in Prozent angegeben und durch

$$\frac{\|\tilde{f}\|_2}{\|f\|_2} \cdot 100\%.$$

berechnet. Mit der Genauigkeitsschranke in Prozent ergibt sich dann eine Bedingung für die Anzahl der notwendigen Spektralkoeffizienten.

Weiterhin können anstelle der komplexen Spektralkoeffizienten $\hat{f}_{k\mu\nu}$ auch die Real- und Imaginärteile getrennt betrachtet werden. Wir bezeichnen diese im folgenden einfach als die *reellen Koeffizienten* der Spektraltransformation.

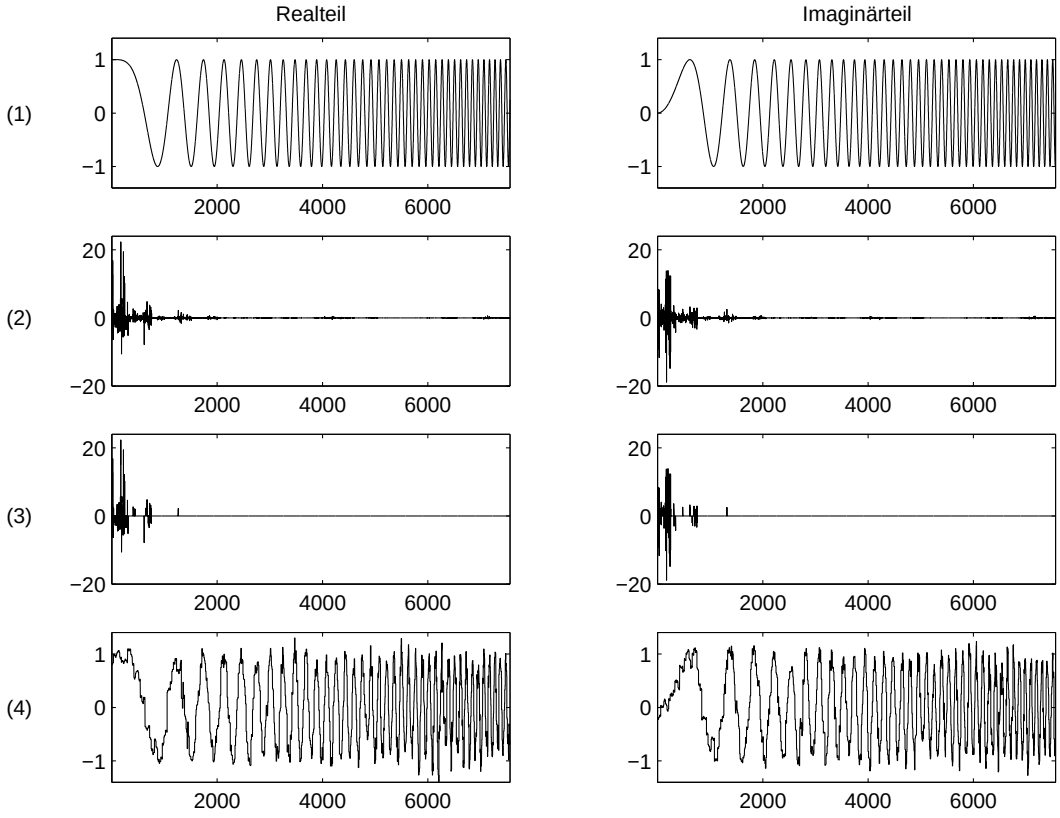


Abbildung 7.6: Thresholding eines Chirpsignals bezüglich der Gruppe $\mathrm{Lib}_{7560}(37)$. (1) Signal f, (2) Spektralkoeffizienten $\hat{f}_{k\mu\nu}$, (3) Spektralkoeffizienten mit $|\hat{f}_{k\mu\nu}| \leq \delta$, (4) Signal $\tilde{f}$.

Abbildung 7.6 illustriert die Vorgehensweise an einem Beispiel. In Zeile (1) sind Real- und Imaginärteil eines komplexen Chirpsignals f der Länge $N = 7560$ abgebildet. Dieses Signal wurde bezüglich der durch die Gruppe $G = \mathrm{Lib}_{7560}(37)$ (siehe Abschnitt 3.4.1) definierten DFT in den Fourierbereich transformiert. Zeile (2) zeigt den Real- bzw. den Imaginärteil der Spektralkoeffizienten, die lexikographisch bezüglich der Tripel (k, μ, ν) angeordnet sind (siehe (2.4)). Es handelt sich dabei um 15120 reelle Koeffizienten, von denen die 200 größten Koeffizienten ausgewählt und alle anderen Koeffizienten auf Null gesetzt wurden (Zeile (3) von Abbildung 7.6). Dies entspricht also einer Kompressionsrate von $200 : 15120 \approx 1 : 76$, wobei sich 96.3% der Signalenergie in den 200 Koeffizienten konzentriert. Aus der Bedingung für die Anzahl der Koeffizienten ergibt sich in diesem Beispiel der Schwellenwert $\delta = 1.846$, d. h. alle Real- und Imaginärteile der Spektralkoeffizienten, deren Betrag unter dieser Schranke liegen, wurden beim Thresholding auf Null gesetzt. Durch Rücktransformation erhält man das in Zeile (4) dargestellte approximative Signal $\tilde{f}$.

7.2.2 Gesamtsystem

Das im letzten Abschnitt beschriebene Verfahren zur Datenkompression mittels verallgemeinerter Spektraltransformationen wurde erstmals implementiert. Wir diskutieren in den folgenden Punkten die wesentlichen Merkmale des Gesamtsystems, das in Abbildung 7.7 dargestellt ist:

(1) Gegeben ist eine Bibliothek von konsistenten pc-Präsentationen, die überauflösbare Gruppen G verschiedener Ordnung N definieren (siehe Abschnitt 3.4.1). Die Daten, die für die Beschreibung einer pc-Präsentation für eine Gruppe der Ordnung N benötigt werden, haben einen Speicherplatzbedarf von $O(\log^3 N)$ (siehe (2.7) und (2.8)).

(2) Sei f der zu komprimierende komplexe Datenvektor der Länge N. Zur Speicherung dieses Vektors werden $2 \cdot N$ Gleitkommazahlen benötigt. Aus der Bibliothek wird eine konsistente pc-Präsentation ausgewählt, die eine Gruppe $G = \mathrm{Lib}_N(k)$ der Ordnung N mit Kompositionsreihe $\mathcal{T}$ definiert.

(3) Zu G wird eine Transversale $\mathcal{T}$-angepaßter $e(\mathcal{T})$-monomialer irreduzibler Darstellungen $\mathrm{Irr}(G, \mathcal{T})$ berechnet. Dies wird durch den BC-Algorithmus mit $O(N \log N)$ Additionen in $\mathbb{Z}_e$, wobei e den Exponenten von G bezeichnet, bewerkstelligt. Die Darstellungen werden in Form der DFT-Datenstruktur abgespeichert, die einen Speicherplatzbedarf von $O(N)$ Zahlen in $\mathbb{Z}_e$ hat. (Siehe Kapitel 3.)

(4) Die Berechnung der verallgemeinerten Spektralkoeffizienten $f_{k\mu\nu}$ wird auf eine DFT-Auswertung zurückgeführt (siehe (6.18)). Die DFT kann schnell mit Hilfe der FFT (Baum-Algorithmus) mit $O(N \log N)$ komplexen Operationen (also Gleitkommaoperationen) ausgewertet werden. Der Algorithmus benötigt dabei einen linearen Speicherplatzbedarf $O(N)$ an Gleitkommazahlen. (Siehe Kapitel 5.)

(5) Ist die vorgegebene Genauigkeitsschranke, die für die Kompression eingehalten werden soll, in Form der relativen $\|\cdot\|_2$-Genauigkeit angegeben, so müssen die Spektralkoeffizienten der Größe ihrer Beträge nach sortiert werden. Standard-Sortieralgorithmen, wie z. B. Mergesort, bewerkstelligen dies in $O(N \log N)$. Das Auswählen der größten Koeffizienten, bis die geforderte Energieschranke überschritten ist, geht dann in $O(N)$.

(6) Analog zu (4) kann unter Benutzung der IFFT das komprimierte Signal $\tilde{f}$ berechnet werden. Dies benötigt wie in (4) den Aufwand $O(N \log N)$.

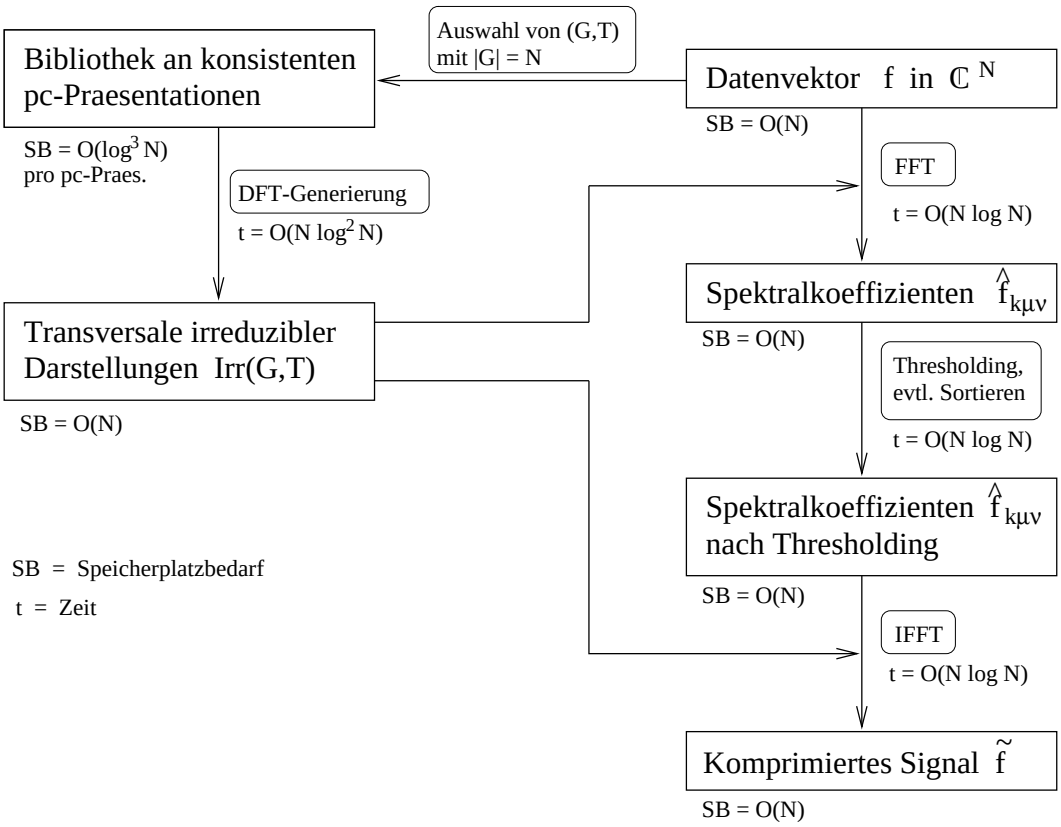


Abbildung 7.7: Gesamtsystem zur DFT-basierten Datenkompression.

Alle Berechnungen können mit Hilfe schneller Algorithmen durchgeführt werden, die jeweils eine Laufzeit von $O(N \log N)$ haben und mit einem Speicherplatz $O(N)$ auskommen. Das implementierte Gesamtsystem ermöglicht damit eine Datenkompression, die in *Echtzeit* durchgeführt werden kann.

Im allgemeinen gibt es für festes N eine große Anzahl überauflösbarer Gruppen. Zum Beispiel wächst die Anzahl der Isomorphieklassen von Gruppen der Ordnung p^n, wobei p eine Primzahl bezeichnet, exponentiell in n. Higman und Sims geben in [34] bzw. [54] asymptotische Abschätzungen, welche zeigen, daß die Anzahl der Isomorphieklassen von Gruppen der Ordnung p^n sich wie $p^{2n^3/27+O(n^{8/3})}$ verhält. Besche und Eick geben in [6] die Anzahl der Isomorphieklassen von Gruppen für alle Ordnungen bis 1000 mit Ausnahme der Ordnungen 512 und 768 an. Unter anderem gibt es 2328 Isomorphieklassen von Gruppen der Ordnung 128 und 56092 für die Ordnung 256. In [27] wird gezeigt, daß es für die Ordnung 512 bereits 10.494.213 Isomorphietypen gibt.

Jede Isomorphieklasse einer überauflösbaren Gruppe definiert eine grundsätzlich verschiedene Spektralzerlegung, die mit dem oben beschriebenen System schnell generiert und ausgewertet

werden kann. Aufgrund der großen Anzahl nicht-isomorpher überauflösbarer Gruppen, z. B. für Gruppen der Ordnungen p^n, bietet sich folgendes *randomisierte adaptive* Verfahren zur Datenkompression an (siehe auch [15]):

- Wähle für jeden zu komprimierenden Datenvektor f der Länge N randomisiert eine pc-Präsentation der Bibliothek zu einer Gruppe der Ordnung N und führe die zuvor beschriebene Datenkompression aus.
- Wiederhole dieses Verfahren für festes f mehrere Male und speichere das bezüglich der gewählten Qualitätsmaßes beste Kompressionsergebnis samt der zugehörigen pc-Präsentation ab.

Als Kriterium für die Wahl der „besten“ pc-Präsentation kann hierzu z. B. die minimale Anzahl der nicht auf Null gesetzten Spektralkoeffizienten oder die maximale Energie des komprimierten Signals bei fest vorgegebener Anzahl der zu speichernden Spektralkoeffizienten dienen.

7.3 Experimente

In diesem Abschnitt beschreiben wir einige der Experimente, die mit dem zuvor beschriebenen Gesamtsystem zur Datenkompression durchgeführt wurden. In den angeführten Beispielen wird diskutiert, inwieweit sich die durch die verschiedenen pc-Präsentationen definierten Spektraltransformationen unterscheiden. Unter anderem wurde bei Vorgabe der $\|\cdot\|_2$-Genauigkeitsschranke (siehe Abschnitt 7.3.1) und bei Vorgabe der Kompressionsrate (siehe Abschnitt 7.3.2) eine statistische Untersuchung zum Verhalten der verschiedenen Spektraltransformationen durchgeführt. Abschließend illustrieren wir in Abschnitt 7.3.3 anhand zweier Beispiele, wie man durch randomisierte Datenkompression eine gute Approximation der Wellenform der Originalsignale erhält.

7.3.1 Vorgabe der $\|\cdot\|_2$-Genauigkeitsschranke

Im folgenden Experiment wurde das in (1) von Abbildung 7.8 dargestellte Signal f, das aus Sinusschwingungen zusammengesetzt ist, mit Hilfe verschiedener Gruppen der Ordnung 128 komprimiert. Dabei wurde für das Thresholding eine relative $\|\cdot\|_2$-Genauigkeitsschranke von 99% vorgegeben, d. h. im Spektralbereich wurden so viele der betragsmäßig größten Real- und Imaginärteile der Spektralkoeffizienten gewählt, bis mindestens 99% der Signalenergie von f in diesen Koeffizienten konzentriert war. Das Thresholding wurde für insgesamt 2328 pc-Präsentationen durchgeführt, wobei die pc-Präsentationen paarweise nicht-isomorphe Gruppen definieren (es gibt 2328 Isomorphieklassen von Gruppen der Ordnung 128, siehe Abschnitt 3.4.1). Unter allen Gruppen der Bibliothek verhält sich $G = \mathrm{Lib}_{128}(40)$ am günstigsten: 99% der Signalenergie konzentrieren sich in 44 der 256 Real- und Imaginärteile der Spektralkoeffizienten. Das komprimierte, aus diesen Koeffizienten rücktransformierte Signal $\tilde{f}$ ist in (2) dargestellt. Im Gegensatz hierzu werden bei der Gruppe $G = \mathrm{Lib}_{128}(456)$ für dieselbe Aufgabe 171 reelle Koeffizienten benötigt (siehe (3)).

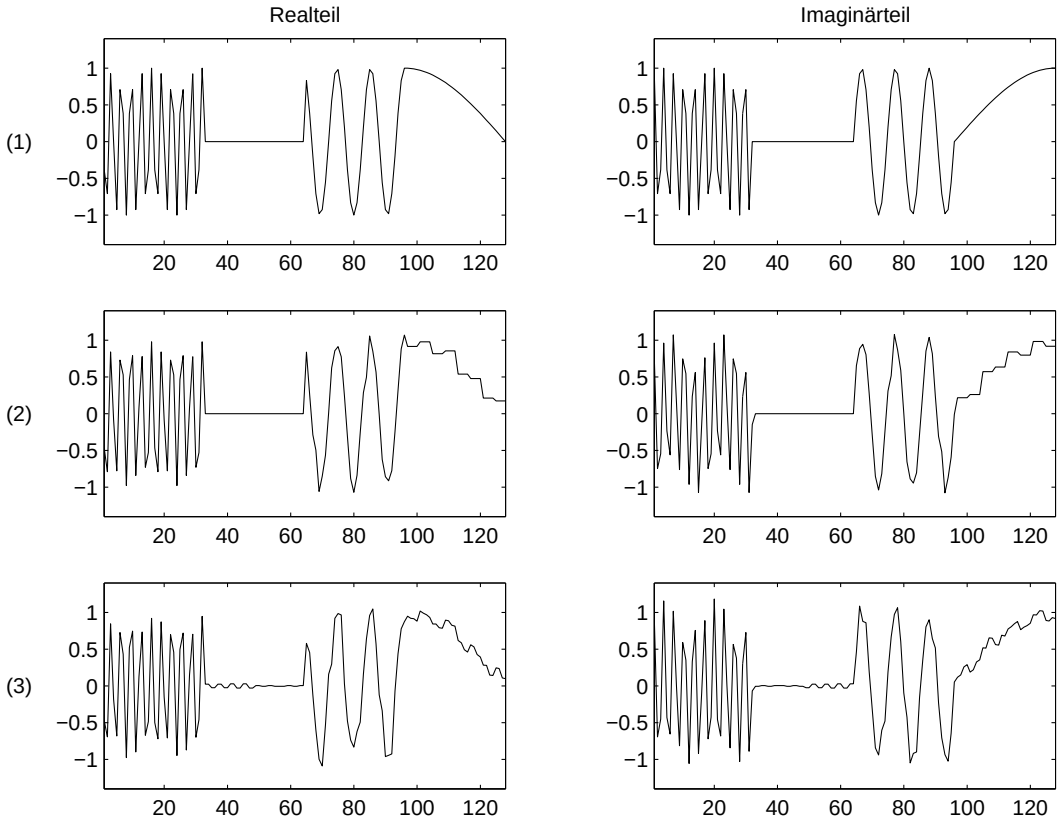


Abbildung 7.8: Thresholding bei relativer $\|\cdot\|_2$-Genauigkeit von 99%. (1) Originalsignal f, (2) $\tilde{f}$ für $G = \mathrm{Lib}_{128}(40)$ mit 44 Koeffizienten, (3) $\tilde{f}$ für $G = \mathrm{Lib}_{128}(456)$ mit 171 Koeffizienten.

In Abbildung 7.9 kann man ablesen, wieviele der 2328 Gruppen die oben gestellte Aufgabe mit jeweils welcher Anzahl an reellen Koeffizienten lösen können. Die in (2.9) definierte Gruppe G_{128} benötigt übrigens 92 Koeffizienten und liegt damit im Mittelfeld.

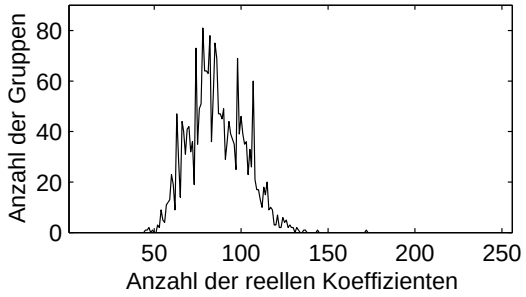


Abbildung 7.9: Anzahl der benötigten Koeffizienten bei relativer $\|\cdot\|_2$-Genauigkeit von 99%

Damit kann mit der Gruppe $G = \mathrm{Lib}_{128}(40)$ die höchste Kompressionsrate von $44 : 256 \approx 1 : 6$ erzielt werden. Natürlich ist die Wahl der „besten“ Gruppe vom jeweiligen Signal f abhängig, wie wir auch in den nächsten Abschnitten sehen werden.

7.3.2 Vorgabe der Anzahl der Koeffizienten

Unter Verwendung derselben Gruppen wie in Abschnitt 7.3.1 wurde in diesem Experiment die Kompressionsrate in Form der Anzahl der zu speichernden Real- und Imaginärteile der Spektralkoeffizienten vorgegeben. Beim Signal f handelt es sich um ein Peak an der Stelle 64. Dies ist bekanntlich für die klassische Fouriertransformation der „worst case“. Da die Koordinatenfunktionen der zyklischen DFT periodisch sind, müssen zur Darstellung von lokalen Phänomenen viele Koordinatenfunktionen herangezogen werden. Mit Hilfe verallgemeinerter Spektraltransformationen kann der Peak unter Umständen aus viel weniger Koordinatenfunktionen zusammengesetzt werden. Ist insbesondere die Gruppe nicht-abelsch und besitzt sie hochdimensionale monomiale Darstellungen, so haben die entsprechenden Koordinatenfunktionen einen schmalen Träger und eignen sich gut zur Darstellung lokaler Phänomene.

Bei der Gruppe $G = \mathrm{Lib}_{128}(138)$ sind z. B. in 25 reellen Koeffizienten 97% der Signalenergie enthalten. Dies entspricht einer Kompressionsrate von $1 : 10$. Der Peak, wie (2) in Abbildung 7.10 zeigt, kann gut durch wenige Koordinatenfunktionen approximiert werden. Bei der Gruppe $G = \mathrm{Lib}_{128}(128)$ hingegen bleibt bei Verwendung der 25 größten Koeffizienten nur 44% der Signalenergie erhalten. Dementsprechend schlecht ist die Approximation $\tilde{f}$ (siehe (3)).

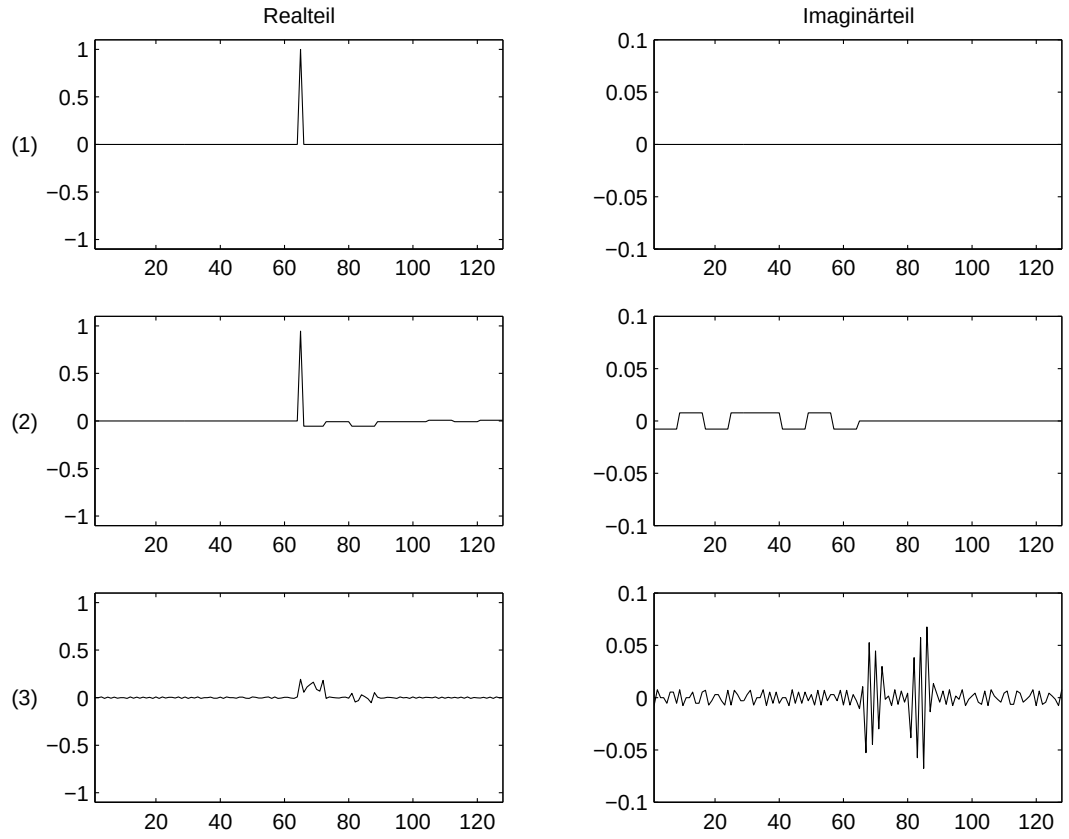


Abbildung 7.10: Kompression um den Faktor 10. (1) Originalsignal f, (2) $\tilde{f}$ für $G = \mathrm{Lib}_{128}(138)$, (3) $\tilde{f}$ für $G = \mathrm{Lib}_{128}(128)$.

Der Grund für das gute Approximationsverhalten der Gruppe $\mathrm{Lib}_{128}(138)$ liegt darin, daß diese eine irreduzible monomiale Darstellung vom Grad 8 hat. (Nebenbei sei angemerkt, daß genau 75 der 2328 Isomorphieklassen der Gruppen der Ordnung 128 eine solche Darstellung haben.) Die entsprechenden Koordinatenfunktionen haben wegen der Monomialität der Darstellung einen schmalen Träger, der einem Achtel der Gesamtlänge des Definitionsbereiches entspricht. Mit anderen Worten nehmen die entsprechenden Koordinatenfunktionen nur auf 16 der 128 Gruppenelemente einen von Null verschiedenen Wert an. Die Gruppe $\mathrm{Lib}_{128}(128)$ hingegen ist abelsch, d. h. alle Darstellungen haben den Grad 1. Damit haben die entsprechenden Koordinatenfunktionen, wie im Fall der klassischen Fouriertransformation, einen vollen Träger und lokale Phänomene lassen sich nur durch Überlagerung vieler verschiedener Koordinatenfunktionen zusammensetzen. In Abbildung 7.11 sind jeweils typische Koordinatenfunktionen für die beiden Gruppen dargestellt.

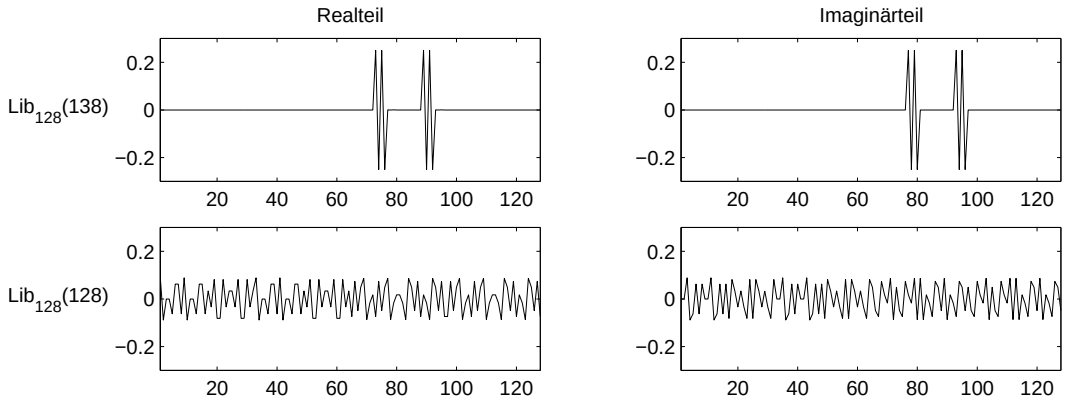


Abbildung 7.11: Typische Koordinatenfunktionen für eine Darstellung vom Grad 8 ($\mathrm{Lib}_{128}(138)$) und eine Darstellung vom Grad 1 ($\mathrm{Lib}_{128}(128)$).

In Abbildung 7.12 findet man wieder eine Statistik über das Verhalten der 2328 pc-Präsentationen der Bibliothek. Nun ist jeweils die Anzahl der Gruppen dargestellt, bei denen sich der entsprechende Energieanteil in Prozent in den 25 größten reellen Koeffizienten konzentriert. Die Gruppe G_{128} von (2.9) bewegt sich mit 76% wieder im Mittelfeld.

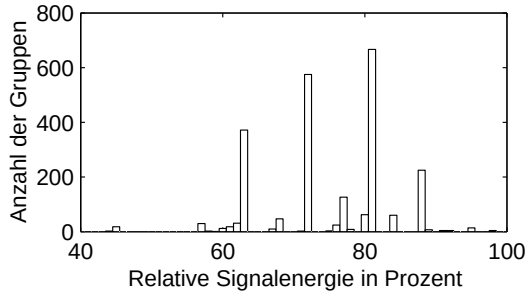


Abbildung 7.12: Relative $\|\cdot\|_2$-Genauigkeit bei fester Anzahl an Koeffizienten

7.3.3 Experimente zur randomisierten Datenkompression

Im Rahmen dieser Arbeit wurden erste Experimente zur randomisierten Datenkompression durchgeführt. Es hat sich dabei herausgestellt, daß das Konzentrationsverhalten der verschiedenen verallgemeinerten Spektraltransformationen sehr stark vom jeweiligen zu komprimierenden Signal abhängt. Z. B. können bei quasi-periodischen Signalen, die keine lokalen Phänomene wie Peaks aufweisen, Koordinatenfunktionen zu niederdimensionalen Darstellungen, deren Träger sich auf den gesamten Definitionsbereich erstrecken, gute Approximationen liefern. Demgegenüber schneiden in diesen Fällen Koordinatenfunktionen hochdimensionaler Darstellungen aufgrund ihrer schmalen Träger eher schlecht ab, die aber dafür um so besser abrupte Veränderungen und lokale Schwankungen im Signal erfassen können (siehe auch Abbildung 7.11). In Abhängigkeit von der Gruppe hat die aus den jeweiligen Koordinatenfunktionen bestehende Orthonormalbasis des Signalraums ein unterschiedliches Approximationsverhalten. Für die Kompression eines gegebenen Signals f kann damit aus einer ganzen Bibliothek an orthogonalen Transformationen adaptiv die für f „beste“ Transformation gewählt werden.

Zur Illustration der randomisierten Datenkompression geben wir zwei Beispiele zu den durchgeführten Experimenten, bei denen u. a. Audiosignale verwendet wurden.

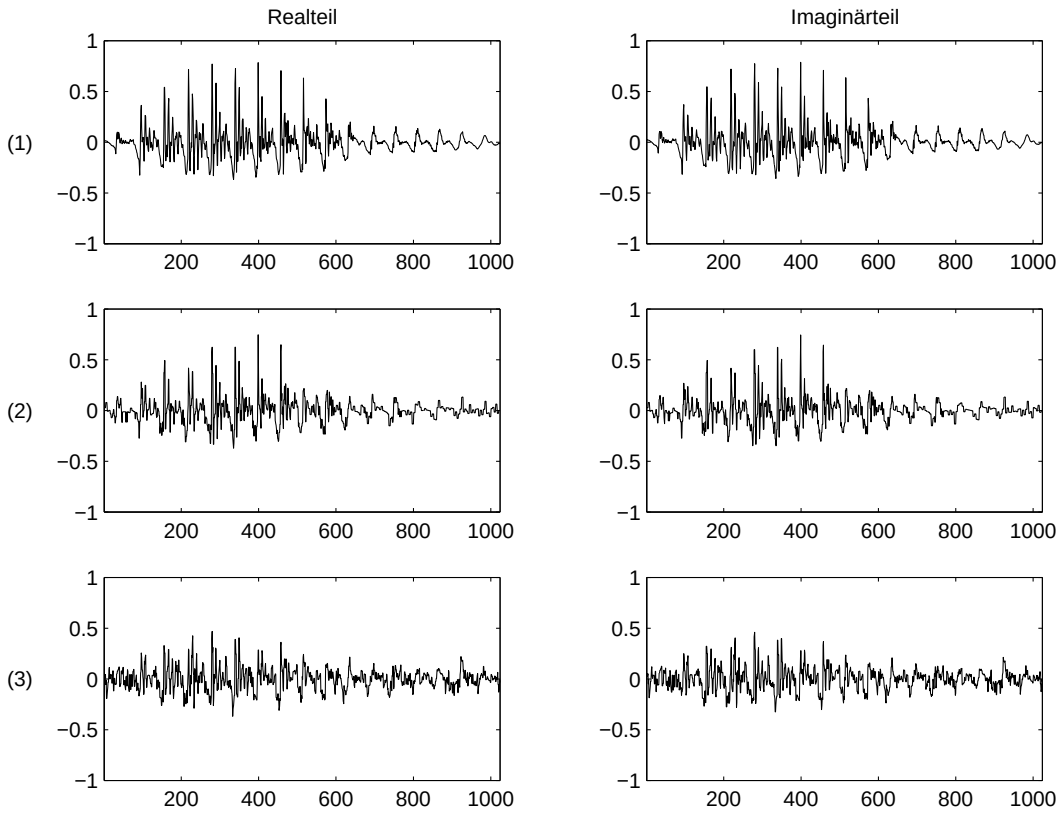


Abbildung 7.13: Kompression um den Faktor 10. (1) Originalsignal f, (2) $\tilde{f}$ für $G = \mathrm{Lib}_{1024}(232)$, (3) $\tilde{f}$ für $G = \mathrm{Lib}_{1024}(102)$.

Zeile (1) von Abbildung 7.13 zeigt den Ausschnitt eines Sprachsignals, welches in etwa dem Wortlaut des Wortes „wann“ entspricht. Hierbei stimmen linker und rechter Kanal überein, die den Real- bzw. Imaginärteil des komplexen Eingangssignals f der Länge $N = 1024$ definieren. Bei Vorgabe der Kompressionsrate $1 : 10$, welches der Wahl der 256 größten reellen Koeffizienten entspricht (siehe Abschnitt 7.3.2), wurden randomisiert konsistente pc-Präsentationen zu Gruppen der Größe 1024 ausgewählt und die Datenkompression durchgeführt. Zeile (2) bzw. Zeile (3) zeigen das dabei beste bzw. schlechteste erzielte Kompressionsergebnis: Bei der Gruppe $G = \mathrm{Lib}_{1024}(102)$ konzentrieren sich 87.5% der Signalenergie in den 256 Koeffizienten, während es bei der Gruppe $G = \mathrm{Lib}_{1024}(232)$ nur 79.5% sind.

Analog zum vorherigen Abschnitt ist in Abbildung 7.14 dargestellt, wie sich die Energie der verwendeten 302 Gruppen der Ordnung 1024 in den 256 größten reellen Koeffizienten konzentriert.

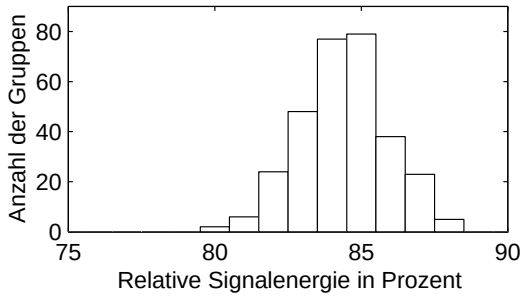


Abbildung 7.14: Relative $\|\cdot\|_2$-Genauigkeit bei fester Anzahl an Koeffizienten

Als Referenz gegenüber dem Heimatort „Böblingen“ des Autors wurde ein entsprechendes Sprachsignal der Länge $N = 7560$ mit 86 zufällig ausgewählten Gruppen derselben Ordnung komprimiert. Das Ergebnis ist in Abbildung 7.15 zu sehen. Während $G = \mathrm{Lib}_{7560}(6)$ immerhin 95.8% der Signalenergie in den 1000 größten reellen Koeffizienten, was einer Kompressionsrate von $1 : 15$ entspricht, zu konzentrieren vermag, bleiben in diesem Fall bei $G = \mathrm{Lib}_{7560}(77)$ nur 88.1% der Energie übrig.

Die schlechtere Approximation der Wellenform von f im Fall $G = \mathrm{Lib}_{7560}(77)$ ist in der Abbildung deutlich zu erkennen: Erstens sind markante Stellen des Originalsignals f (siehe (1)) in der Approximation (3) weniger stark ausgeprägt als in (2). Zweitens ist das rekonstruierte Signal in (3) wesentlich stärker verrauscht als im Fall (2), was auch akustisch nachvollziehbar ist.

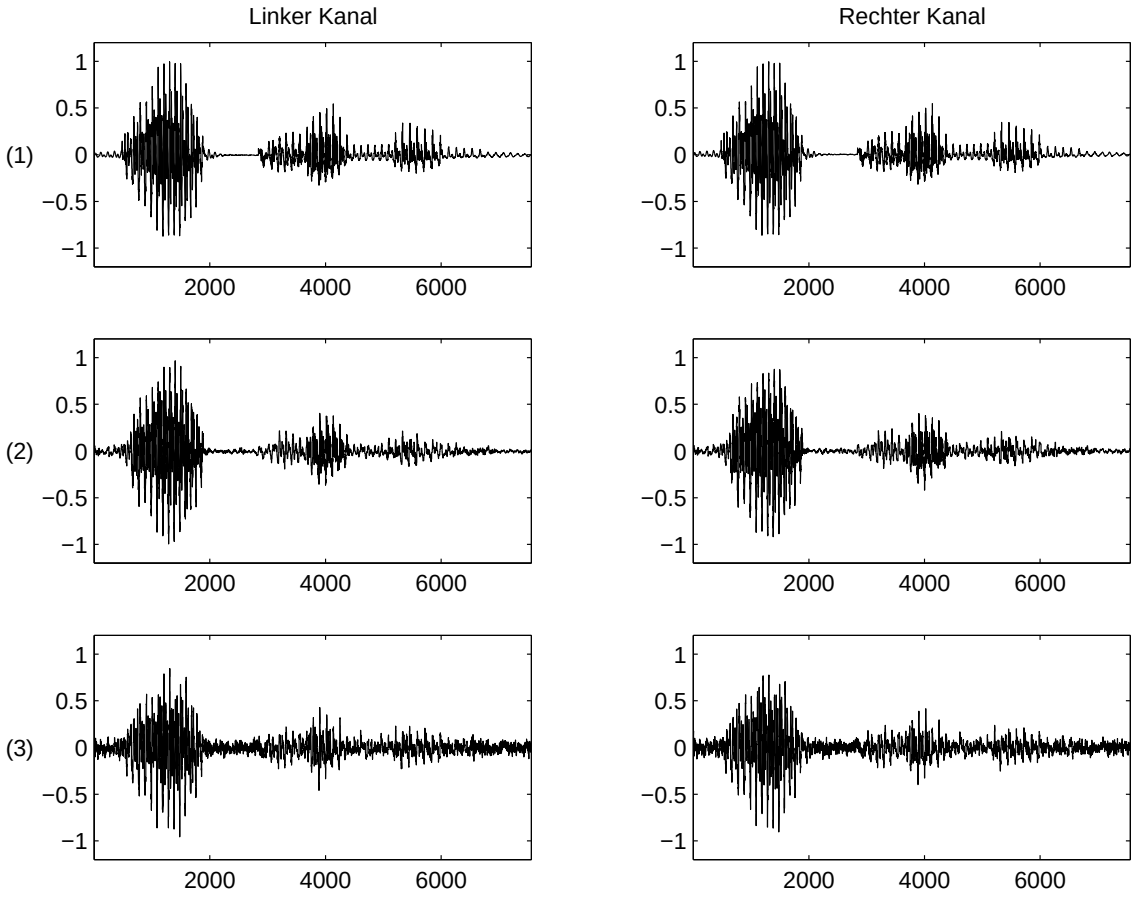


Abbildung 7.15: „Böblingen“ bei 1000 reellen Koeffizienten. (1) Originalsignal f, (2) $\tilde{f}$ für $G = \mathrm{Lib}_{7560}(6)$, (3) $\tilde{f}$ für $G = \mathrm{Lib}_{7560}(77)$.

7.4 Zusammenfassung und Ausblick

Das vorgestellte Verfahren zur Datenkompression wählt in Abhängigkeit vom zu komprimierenden Signal f eine verallgemeinerte Spektraltransformation (definiert durch die zugrundeliegende pc-Präsentation), die in möglichst wenigen Spektralkoeffizienten möglichst viel von der Signalenergie zu konzentrieren vermag. Das aus diesen Koeffizienten rekonstruierte Signal $\tilde{f}$ liefert eine Approximation des Originalsignals f bezüglich der euklidischen Norm $\|\cdot\|_2$. Dies entspricht der Approximation der Wellenform von f durch $\tilde{f}$. Damit eignet sich das Verfahren zur Kompression von Signalen, bei denen kleine Störungen in der Wellenform (typischerweise in Form von Rauschen) akzeptabel sind.

Bei Audiosignalen ist dies nicht der Fall. Hier können schon kleinste Veränderungen der Wellenform (bezüglich der euklidischen Norm) für das menschliche Ohr zu nicht mehr akzeptablen Verzerrungen des Signals führen. Heute gängige Audiokompressionsverfahren berücksichtigen das menschliche Hörempfinden, welches Inhalt der sogenannten *Psychoakustik* ist. Hierbei werden Maskierungseffekte zur Überdeckung von Quantisierungsrauschen, das bei der Kompression entsteht, ausgenutzt (siehe z. B. [66]). Ausgehend von dem vorgestellten Gesamtsystem wäre es interessant zu untersuchen, inwieweit sich verallgemeinerte Fouriertransformationen und randomisierte Kompressionsverfahren in Kombination mit modernen Audiocodierern einsetzen lassen. Weiterhin könnte man testen, ob Verfahren zur Kompression von digitalisierten Bildern, die durch Transformationscodierung bezüglich abelscher Gruppen Kompressionsraten bis zu $1:50$ erzielen (siehe z. B. [28]), bei Verwendung nicht-abelscher Gruppen noch erfolgreicher arbeiten.

Spezialfälle von Gruppen lassen sich, über die auf der MRA basierenden groben Frequenzinterpretationen von Abschitt 7.1.3 hinaus, signaltheoretisch deuten. So läßt sich die klassische Fourierdarstellung bezüglich der zyklischen Gruppen als Frequenzdarstellung eines Signals interpretieren und signaltheoretisch ausnutzen. Eine weitere Aufgabe wäre es zu untersuchen, ob und wie sich weitere Spezialfälle von Gruppen signaltheoretisch deuten lassen. Hierbei wird auch die Numerierung der Gruppenelemente von G (wir haben immer die lexikographische Ordnung von Abschnitt 5.3 vorausgesetzt) eine entscheidende Rolle spielen.

Kapitel 8

DFT-Generierung auflösbarer Gruppen

In Kapitel 3 wurde die DFT-Konstruktion endlicher pc-präsentierter *überauflösbarer* Gruppen behandelt. Der BC-Algorithmus konstruierte eine $\mathcal{T}$-angepaßte e-monomiale DFT für solche Gruppen in einer im wesentlichen optimalen Laufzeit (siehe Abschnitt 3.1). Für die Konstruktion war dabei ganz entscheidend, daß jede endliche überauflösbare Gruppe G eine Kompositionsreihe $\mathcal{T}$ hat, die zugleich eine Hauptreihe ist. Mit anderen Worten, jede Untergruppe G_i einer solchen Kompositionsreihe $\mathcal{T}$ ist nicht nur in der nächst höheren Untergrupppe G_{i+1}, sondern sogar in der gesamten Gruppe G normal. Damit konnten die Verkettungsräume in einer „bottom-up"-Konstruktion entlang $\mathcal{T}$ konstruiert werden, wobei hierfür die Invarianz der Räume $\mathrm{Irr}(G_i)$ unter Konjugation mit den Erzeugern g_j für $j > i$ entscheidend war.

Dies geht für endliche auflösbare Gruppen im allgemeinen nicht mehr. Ist

$$\mathcal{C} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_1 \rhd G_0 = \{1\})$$

eine Hauptreihe einer endlichen auflösbaren Gruppe G, dann kann diese zwar zu einer Kompositionsreihe $\mathcal{T}$ verfeinert werden, allerdings sind dann die Untergruppen im allgemeinen nicht mehr normal in G. Weiterhin besitzen auflösbare Gruppen im allgemeinen keine Isomorphielisten irreduzibler Darstellungen, die aus lauter monomialen Darstellungen besteht. Daher ist die DFT-Konstruktion in diesem Fall wesentlich aufwendiger, und ähnlich effiziente Algorithmen wie der BC-Algorithmus sind nicht mehr zu erwarten.

Püschel beschreibt in seiner Dissertation [47] einen Algorithmus zur Zerlegung der regulären Darstellung einer endlichen Gruppe G, was der Berechnung einer der Hauptreihe angepaßten DFT gleichkommt. Er gibt leider keine theoretische Laufzeitanalyse seines Algorithmus. Die Implementierung seines Algorithmus erfolgte unter Verwendung von GAP-Routinen und ist Teil des AREP-Pakets, eines Programmpakets zur konstruktiven Darstellungstheorie [26]. Seine experimentellen Resultate für kleine Gruppengrößen (bis zur Ordnung 500) lassen auf eine *durchschnittliche* Laufzeit schließen, die quadratisch in der Gruppenordnung ist. (Es sei an dieser Stelle bemerkt, daß die überwiegende Anzahl der Gruppen bis zur Ordnung 500 überauflösbar ist).

In diesem Kapitel stellen wir einen Algorithmus (M-Algorithmus für „Main Algorithm") vor,

der für eine endliche auflösbare Gruppe G eine $\mathcal{T}$-angepaßte DFT konstruiert, und geben die unseres Wissens erste „worst-case"-obere Komplexitätsschranke für den auflösbaren Fall im Hinblick auf die Anzahl der benötigten Körperoperationen an. Dabei muß G durch eine pc-Präsentation gegeben sein, die mit einer Kompositionsreihe $\mathcal{T}$ korrespondiert, die wiederum eine Hauptreihe $\mathcal{C}$ verfeinert. Der M-Algorithmus verfährt nach dem im Abschnitt 2.4 beschriebenen Grundprinzip der DFT-Generierung. Die Konstruktion geschieht wiederum „bottom-up", aber diesmal entlang der Hauptreihe $\mathcal{C}$. Innerhalb eines Hauptfaktors benutzen wir als Unterroutine eine relative Version des BC-Algorithmus zur Konstruktion der Darstellungen, den wir im folgenden als RBC-Algorithmus bezeichnen und in Abschnitt 8.1 beschreiben. Um die nötigen Daten von einer Untergruppe der Hauptreihe zur nächst höheren Untergruppe zu heben, benötigen wir den in Abschnitt 8.2 beschriebenen ET-Algorithmus („Equivalence Test"), der zwei Darstellungen auf Äquivalenz testet. Im Abschnitt 8.3 stellen wir dann den M-Algorithmus vor und analysieren diesen in Abschitt 8.4. Dabei leiten wir unter anderem die „worst-case"-obere Komplexitätsschranke $O(p \cdot |G|^2 \log(|G|))$ her, wobei p den größten Primteiler von $|G|$ bezeichnet.

8.1 RBC-Algorithmus

Für den M-Algorithmus benötigen wir als Unterroutine eine relative Version des BC-Algorithmus für überauflösbare Gruppen (RBC-Algorithmus) . Da die relative Version eine unmittelbare Verallgemeinerung des ursprünglichen Algorithmus ist, verweisen wir für Einzelheiten auf Abschnitt 3.1 und [4] und stellen hier nur das Ergebnis vor.

Sei H eine endliche auflösbare Gruppe mit Normalteiler N, so daß H/N überauflösbar ist. Dann gibt es eine Kette von Untergruppen

$$\mathcal{T} = (H = H_r \rhd H_{r-1} \rhd \ldots H_1 \rhd H_0 = N),$$

wobei $H_k \lhd H$ und $[H_k : H_{k-1}] =: p_k$ prim ist. Wir setzen die folgenden Daten als bekannt voraus:

(i) Eine pc-Präsentation von H relativ zu N, die mit $\mathcal{T}$ korrespondiert und Erzeuger $h_1, \ldots, h_r$ hat, so daß $h_k H_{k-1}$ für $k = 1, \ldots, r$ den Quotienten H_k/H_{k-1} erzeugt.

(ii) Eine Transversale $\mathrm{Irr}(N)$ von irreduziblen Darstellungen von N. Darüber hinaus einen Algorithmus, der jedes $F \in \mathrm{Irr}(N)$ vom Grad $f := \deg(F)$ an jedem in Normalform gegebenen $n \in N$ mit $O(f^3 \cdot \log |N|)$ Operationen auswerten kann.

(iii) Die h_k-Operation des Erzeugers h_k auf $\mathrm{Irr}(N)$, gegeben durch eine Permutation π_{h_k} auf der Menge $\mathrm{Irr}(N)$, so daß $\pi_{h_k}(F) \sim F^{h_k}$ für alle $F \in \mathrm{Irr}(N)$, $k = 1, \ldots, r$ gilt.

(iv) Die zugehörigen Verkettungsmatrizen $X_{h_k,F} \in \mathrm{Int}(F^{h_k}, \pi_{h_k}(F)) \setminus \{0\}$.

Der RBC-Algorithmus konstruiert dann eine Transversale $\mathrm{Irr}(H, \mathcal{T})$ von irreduziblen $\mathcal{T}$-angepaßten Darstellungen „bottom-up" entlang den Untergruppen H_k. Eine Analyse dieses Algorithmus analog zu [4] ergibt die obere Komplexitätsschranke

$$O\left(|H| \log^2(|H|) \sqrt{|N|}\right). \tag{8.1}$$

Außerdem gilt, daß jedes $D \in \mathrm{Irr}(H, \mathcal{T})$ eine f-blockmonomiale Darstellung ist, wobei f durch $f := \deg(F)$ mit einem $F \in \mathrm{Irr}(N)$, für das $\langle D|F\rangle > 0$ gilt, definiert ist. Offensichtlich reduziert sich der RBC-Algorithmus im Fall $N = \{1\}$ zum ursprünglichen BC-Algorithmus überauflösbarer Gruppen (siehe Satz 3.1.1).

8.2 ET-Algorithmus

Der ET-Algorithmus, welcher zwei Darstellungen auf Äquivalenz testet, wird als Unterroutine des M-Algorithmus verwendet. Darüber hinaus wird im Fall der Äquivalenz eine nicht-triviale Verkettungsmatrix konstruiert. Grundlage des ET-Algorithmus ist das folgende Lemma, das eine Idee von Plesken [46] verallgemeinert.

Lemma 8.2.1 *Seien G eine endliche Gruppe, H eine Untergruppe von G vom Index $s := [G : H]$ und $g_1, \ldots, g_s$ Vertreter der Rechtsnebenklassen von H, d. h. $G = Hg_1 \sqcup \ldots \sqcup Hg_s$. Sei weiterhin $\mathbb{K}$ ein Körper mit $\mathrm{char}(\mathbb{K}) \nmid s$ und seien D, Δ Darstellungen von G über $\mathbb{K}$. Dann definiert*

$$\psi : \mathrm{Int}(D{\downarrow}H, \Delta{\downarrow}H) \to \mathrm{Int}(D, \Delta), \quad Y \mapsto \frac{1}{s} \sum_{i=1}^{s} \Delta(g_i^{-1}) Y D(g_i)$$

eine $\mathbb{K}$-lineare Projektion von $\mathrm{Int}(D{\downarrow}H, \Delta{\downarrow}H)$ auf $\mathrm{Int}(D, \Delta)$.

Beweis: Die $\mathbb{K}$-Linearität von ψ ist offensichtlich, und aus der Definition (2.2) des Verkettungsraums folgt für alle $Y \in \mathrm{Int}(D, \Delta)$ sofort $\psi(Y) = Y$. Damit ist nur noch $\psi(Y) \in \mathrm{Int}(D, \Delta)$ für alle $Y \in \mathrm{Int}(D{\downarrow}H, \Delta{\downarrow}H)$ zu zeigen. Fixiere ein solches Y, dann gilt nach (2.2)

$$Y D(h) = \Delta(h) Y \tag{8.2}$$

für alle $h \in H$. Für jedes $g \in G$ gibt es offensichtlich $h_1 \ldots, h_s \in H$, so daß (als Mengen!)

$$\{g_1 g, \ldots, g_s g\} = \{h_1 g_1, \ldots, h_s g_s\} \tag{8.3}$$

gilt. Daher gilt für dieses g die folgende Gleichung:

$$\begin{aligned} \Delta(g^{-1})\psi(Y)D(g) &= \frac{1}{s} \sum_{i=1}^{s} \Delta((g_i g)^{-1}) Y D(g_i g) \overset{(8.3)}{=} \frac{1}{s} \sum_{i=1}^{s} \Delta((h_i g_i)^{-1}) Y D(h_i g_i) \\ &\overset{(8.2)}{=} \frac{1}{s} \sum_{i=1}^{s} \Delta(g_i^{-1}) Y D(g_i) = \psi(Y). \end{aligned}$$

□

Wir kommen nun zum ET-Algorithmus. Sei H eine endliche auflösbare Gruppe mit Normalteiler N und sei

$$\mathcal{T} = (H = H_r \rhd H_{r-1} \rhd \ldots \rhd H_1 \rhd H_0 = N)$$

eine Kette von Untergruppen mit Primzahlindizes $[H_k : H_{k-1}] =: p_k$, $k = 1, \ldots, r$. In diesem Abschnitt setzen wir nicht voraus, daß die H_k normal in der gesamten Gruppe H sind. Wie

zuvor seien $h_k \in H$ so gewählt, daß $h_k H_{k-1}$ den Quotienten H_k/H_{k-1} erzeugt. (Weil H_{k-1} normal in H_k ist, stimmen die Rechtsnebenklassen von H_{k-1} in H_k mit den Linksnebenklassen überein.) Wir definieren für zwei Darstellungen D, Δ von H die Abbildung

$$\psi_k : \mathrm{Int}(D{\downarrow}H_{k-1}, \Delta{\downarrow}H_{k-1}) \to \mathrm{Int}(D{\downarrow}H_k, \Delta{\downarrow}H_k), \quad Y \mapsto \frac{1}{p_k} \sum_{t=0}^{p_k-1} \Delta(h_k^{-t}) Y D(h_k^t).$$

Dann definiert $\psi := \psi_r \circ \psi_{r-1} \circ \cdots \circ \psi_1$ eine Projektion von $\mathrm{Int}(D{\downarrow}N, \Delta{\downarrow}N)$ auf $\mathrm{Int}(D, \Delta)$. Sind D, Δ irreduzible Darstellungen, so folgt aus dem Lemma von Schur, daß für $Y \in \mathrm{Int}(D{\downarrow}N, \Delta{\downarrow}N)$ die Abbildung $\psi(Y)$ entweder die Nullabbildung oder invertierbar ist. Sei nun $\mathcal{B}$ eine Basis von $\mathrm{Int}(D{\downarrow}N, \Delta{\downarrow}N)$. Zwei irreduzible Darstellungen D, Δ können durch Berechnung aller Bilder $\psi(E)$, $E \in \mathcal{B}$, auf Äquivalenz getestet werden. Gibt es ein $\psi(E) \neq 0$, dann gilt $D \sim \Delta$, und $\psi(E)$ erzeugt den eindimensionalen Verkettungsraum $\mathrm{Int}(D, \Delta)$. Im anderen Fall gilt $\psi(E) = 0$ für alle $E \in \mathcal{B}$, und aus der Surjektivität von ψ folgt $\mathrm{Int}(D, \Delta) = \{0\}$, was $D \not\sim \Delta$ impliziert.

Sei $d = \deg(D) = \deg(\Delta)$ und $Y \in \mathrm{Int}(D{\downarrow}N, \Delta{\downarrow}N)$. Wegen

$$\Delta(h_k^{-t}) Y D(h_k^t) = \Delta(h_k)^{-1} \left(\Delta(h_k^{-(t-1)}) Y D(h_k^{t-1}) \right) D(h_k),$$

$t = 1, \ldots, p_k - 1$, kann $\psi_k(Y)$ für $k \in [1 : r]$, mit $O(p_k d^3)$ Operationen berechnet werden. Daher läßt sich $\psi(Y)$ mit

$$\sum_{k=1}^{r} O\left(p_k d^3\right) = O\left(d^3 \sum_{k=1}^{r} p_k\right)$$

Operationen berechnen. Für den Äquivalenztest muß $\psi(E)$ für alle $E \in \mathcal{B}$ berechnet werden. Dies ist mit $O(|\mathcal{B}| \cdot d^3 \sum_{k=1}^{r} p_k)$ Operationen zu bewerkstelligen. Sind die irreduziblen Darstellungen D und Δ darüber hinaus f-blockmonomial, $f \mid d = \deg(D) = \deg(\Delta)$, dann ist die Berechnung von $\psi_k(Y)$ billiger. Sie benötigt unter Benutzung von (2.11) nur $O(p_k \cdot (\frac{d}{f})^2 \cdot f^3) = O(p_k \cdot d^2 \cdot f)$ Operationen. Dies führt zu einem Gesamtaufwand von

$$O\left(|\mathcal{B}| \cdot d^2 \cdot f \sum_{k=1}^{r} p_k\right) \tag{8.4}$$

Operationen für den ET-Algorithmus zum Test von $D \sim \Delta$.

8.3 M-Algorithmus

In diesem Abschnitt stellen wir den M-Algorithmus zur Konstruktion einer $\mathcal{T}$-angepaßten DFT einer endlichen auflösbaren Gruppe G vor. Sei

$$\mathcal{C} = (G = G_n \rhd G_{n-1} \rhd \ldots \rhd G_1 \rhd G_0 = \{1\})$$

eine Hauptreihe von G, dann gilt $G_i \lhd G$. Darüber hinaus sind die Hauptfaktoren elementarabelsch. Mit anderen Worten, es existieren $r_i \in \mathbb{N}$ und Primzahlen p_i, so daß $G_i/G_{i-1} \simeq C_{p_i}^{r_i}$

gilt. (Für einen Beweis verweisen wir auf Theorem (9.13) von [35].) Wir verfeinern die Hauptreihe zu einer Kompositionsreihe $\mathcal{T}$ von G mit geeigneten Untergruppen

$$G_i = G_{ir_i} \rhd G_{ir_i-1} \rhd \ldots \rhd G_{i1} \rhd G_{i0} = G_{i-1}.$$

Im allgemeinen sind die Gruppen G_{ik}, $1 \leq k < r_i$, nicht normal in G. Sei weiterhin G durch eine pc-Präsentation mit Erzeugern $\{g_{ik} \in G \mid 1 \leq i \leq n, 1 \leq k \leq r_i\}$ gegeben, welche mit $\mathcal{T}$ korrespondiert: $g_{ik}G_{ik-1}$ erzeuge also $G_{ik}/G_{ik-1} \simeq C_{p_i}$.

Der M-Algorithmus verfährt „bottom-up“ entlang der Hauptreihe $\mathcal{C}$. Wie in Abschnitt 2.4 sei $\mathcal{T}_i$ durch $\mathcal{T}_i := (G_i \supset \ldots \supset G_0)$, $1 \leq i \leq n$, definiert. Dann stehen in Schritt i folgende Daten als Eingabe zur Verfügung:

(1) $\mathcal{F} := \mathrm{Irr}(G_{i-1}, \mathcal{T}_{i-1})$, eine Transversale von $\mathcal{T}_{i-1}$-angepaßten irreduziblen Darstellungen von G_{i-1} und der zugehörige Charaktergraph von G_{i-1}.

(2) Für jedes $i-1 < j \leq n$ und $1 \leq k \leq r_j$ die g-Operation, $g := g_{jk}$, auf $\mathcal{F}$ durch eine Permutation π_g von $\mathcal{F}$, so daß $F^g \sim \pi_g F$ für alle $F \in \mathcal{F}$. Darüber hinaus die Verkettungsmatrizen $X_{gF} \in \mathrm{Int}(F^g, \pi_g F)$ für jedes $F \in \mathcal{F}$.

Die folgenden Daten werden berechnet:

(1) $\mathcal{D} := \mathrm{Irr}(G_i, \mathcal{T}_i)$, eine Transversale von $\mathcal{T}_i$-angepaßten irreduziblen Darstellungen von G_i und der zugehörige Charaktergraph von G_i.

(2) Für jedes $i < j \leq n$ und $1 \leq k \leq r_j$ die g-Operation, $g := g_{jk}$, von $\mathcal{D}$ gegeben durch eine Permutation τ_g von $\mathcal{D}$, so daß $D^g \sim \tau_g D$ für alle $D \in \mathcal{D}$. Darüber hinaus die Verkettungsmatrizen $Y_{gD} \in \mathrm{Int}(D^g, \tau_g D)$ für jedes $D \in \mathcal{D}$.

Dabei ist die Eingabe für Schritt 0 trivial. Der Schritt i des M-Algorithmus besteht aus zwei Phasen:

Phase 1. Sei $H := G_i$, $N := G_{i-1}$, $r := r_i$ und $p := p_i$. Dann ist N normal in H und H/N ist elementar-abelsch, insbesondere überauflösbar. Setze $H_k := G_{ik}$, $k = 0, \ldots, r$, und $h_k := g_{ik}$, $k = 1, \ldots, r$, dann ist Voraussetzung (i) des RBC-Algorithmus erfüllt. Voraussetzung (ii) gilt, da nach Induktionshypothese (1) von Schritt $i-1$ die Menge $\mathcal{F} := \mathrm{Irr}(N, \mathcal{T}_{i-1})$ bereits konstruiert ist, wobei $F \in \mathcal{F}$ auf den Erzeugern von N gegeben ist. Daher kann nach (2.13) $F(n)$ in $O(f^3 \cdot \log|N|)$, $f := \deg(F)$, für jedes $n \in N$ in Normalform berechnet werden. Die Voraussetzungen (iii) und (iv) sind nach Induktionshypothese (2) von Schritt $i-1$ erfüllt. Daher kann der RBC-Algorithmus zur Konstruktion von $\mathcal{D} := \mathrm{Irr}(H, \mathcal{T}_i)$ verwendet werden, welches die Ausgabe (1) von Schritt i ist.
Darüber hinaus werden im RBC-Algorithmus alle Daten berechnet, die zur Erweiterung des Charaktergraphen von G_{i-1} auf G_i benötigt werden.

Phase 2. Sei $g := g_{jk}$, $i < j \leq n$, $1 \leq k \leq r_j$, fest und $D \in \mathcal{D}$, $d := d(D) := \deg(D)$. Zur Bestimmung von $\tau_g D$ benötigen wir die Darstellung $\Delta \in \mathcal{D}$ mit $D^g \sim \Delta$. Wir verringern die Anzahl der möglichen Kandidaten in $\mathcal{D}$ unter Verwendung der Induktionshypothese

(2) aus Schritt $i-1$ und des Charaktergraphen von G_i aus Phase 1.
Die Zerlegung der Einschränkung $D{\downarrow}N$ in irreduzible Darstellungen von $\mathcal{F}$ kann aus dem Charaktergraph von G_i abgelesen werden. Sei $F \in \mathcal{F}$, $f := \deg(F)$, mit $m := m(D) := \langle D|F\rangle > 0$ und $\{F_1 = F, F_2, \ldots, F_q\} \subset \mathcal{F}$, $q := q(D) \in \mathbb{N}$, die Bahn von F unter der H-Operation auf $\mathcal{F}$. Nach dem Satz von Clifford 2.1.2 gilt

$$D{\downarrow}N \sim m \cdot \bigoplus_{k=1}^{q} F_k.$$

Da D eine $\mathcal{T}_i$-angepaßte Darstellung ist, gibt es eine Permutationsmatrix P der Form $P = P_\sigma \otimes \mathrm{Id}_f$ mit einer Permutation $\sigma \in S_{d/f}$, so daß

$$D{\downarrow}N = P \left(\bigoplus_{k=1}^{q} m \cdot F_k \right) P^{-1}.$$

Aus $D^g \sim m \cdot \bigoplus_{k=1}^{q} F_k^g \sim m \cdot \bigoplus_{k=1}^{q} \pi_g F_k$ folgt, daß

$$\Delta \in \{\Delta_1, \Delta_2, \ldots, \Delta_\ell\} \subset \mathcal{D}, \quad \ell := \ell(D) \in \mathbb{N},$$

wobei nach Definition diese Menge aus genau denjenigen Darstellungen in $\mathcal{D}$ besteht, deren Einschränkung auf N äquivalent zu $m \cdot \bigoplus_{k=1}^{q} \pi_g F_k$ ist. Diese Information kann wiederum dem Charaktergraphen von G_i entnommen werden. Wir benutzen nun den ET-Algorithmus, um die Darstellungen Δ_λ, $1 \le \lambda \le \ell$ auf Äquivalenz mit D^g zu testen. Hierfür benötigen wir eine Basis $\mathcal{B}$ von $\mathrm{Int}(D^g{\downarrow}N, \Delta_\lambda{\downarrow}N)$. Aus der $\mathcal{T}_i$-Angepaßtheit von Δ_λ folgt die Existenz einer Permutationsmatrix Q_λ der Form $Q_\lambda = \sigma_\lambda \otimes \mathrm{Id}_f$ mit einer Permutation $\sigma_\lambda \in S_{d/f}$, so daß

$$\Delta_\lambda{\downarrow}N = Q_\lambda \left(\bigoplus_{k=1}^{q} m \cdot \pi_g F_k \right) Q_\lambda^{-1}.$$

Aus Lemma 2.1.1 folgt nun

$$\begin{aligned}
\mathrm{Int}\left(D^g{\downarrow}N, \Delta_\lambda{\downarrow}N\right) &= \mathrm{Int}\left(P \left(\bigoplus_{k=1}^{q} m \cdot F_k^g \right) P^{-1}, Q_\lambda \left(\bigoplus_{k=1}^{q} m \cdot \pi_g F_k \right) Q_\lambda^{-1} \right) \\
&= Q_\lambda \mathrm{Int}\left(\bigoplus_{k=1}^{q} m \cdot F_k^g \,,\, \bigoplus_{k=1}^{q} m \cdot \pi_g F_k \right) P^{-1} \\
&= Q_\lambda \left[\bigoplus_{k=1}^{q} \mathbb{C}^{m\times m} \otimes \mathrm{Int}\left(F_k^g, \pi_g F_k\right) \right] P^{-1}.
\end{aligned}$$

Alle $X_{gF_k} \in \mathrm{Int}(F_k^g, \pi_g F_k)$ sind nach Induktionshypothese (2) von Schritt $i-1$ bekannt, und

$$\mathcal{B} = \left\{ E_{abc} := Q_\lambda \left[\bigoplus_{k=1}^{q} \delta_{kc} \cdot (E_{ab} \otimes X_{gF_k}) \right] P^{-1}, 1 \le a, b \le m, 1 \le c \le q \right\} \tag{8.5}$$

definiert eine Basis von $\mathrm{Int}(D^g{\downarrow}N, \Delta_\lambda{\downarrow}N)$, wobei E_{ab} die $m\times m$-Matrix mit genau einem von Null verschiedenen Eintrag 1 an der Position (a, b) bezeichnet. Offensichtlich gilt

$$|\mathcal{B}| = q \cdot m^2.$$

Der Rest von Phase 2 ist nun eine unmittelbare Anwendung des ET-Algorithmus. Unter Benutzung der Basis $\mathcal{B}$ können wir entscheiden, ob D^g und Δ_λ äquivalent sind oder nicht. Im Fall der Äquivalenz gilt $\Delta = \Delta_\lambda$, und wir setzen $\tau_g D := \Delta_\lambda$. Darüber hinaus liefert in diesem Fall der ET-Algorithmus eine von Null verschiedene Verkettungsmatrix $Y_{gD} \in \mathrm{Int}(D^g, \tau_g D)$. Dies sind genau die Daten (2), die in Schritt i zu bestimmen waren.

8.4 Analyse des M-Algorithmus

In diesem Abschnitt analysieren wir den M-Algorithmus zur Bestimmung des asymptotischen Verhaltens. Im folgenden sei dabei eine arithmetische Operation eine Addition, Subtraktion, Inversion, Multiplikation oder Kopieroperation in unserem Grundkörper K, welche alle als in $O(1)$ berechenbar vorausgesetzt werden. Für eine Diskussion der Probleme, die beim exakten Rechnen über Kreisteilungskörpern $K = \mathbb{Q}^{(e)}$ (anstelle über $K = \mathbb{C}$) entstehen, verweisen wir auf Abschnitt 8.5.

Für unsere Analyse benötigen wir einige Abschätzungen. Nach (2.15) gilt mit der Notation des letzten Abschnitts $\sum_{D\in\mathcal{D}} \deg(D)^2 = |H|$. Aus $\{\Delta_1, \Delta_2, \ldots, \Delta_\ell\} \subset \mathcal{D}$ und $d = \deg(\Delta_\lambda)$ für alle $\lambda = 1, \ldots, \ell(D)$ folgt dann mit $\ell = \ell(D)$

$$\ell(D) \cdot d^2 = \sum_{\lambda=1}^{\ell(D)} \deg(\Delta_\lambda)^2 \leq |H|. \tag{8.6}$$

Nach 2.17 gilt für eine beliebige reelle Zahl $s \geq 2$ die Abschätzung

$$d^s(G) := \sum_{D\in\mathcal{D}} \deg(D)^s \leq d^{s-2} \sum_{D\in\mathcal{D}} \deg(D)^2 \leq |G|^{\frac{s}{2}}. \tag{8.7}$$

Wir analysieren die Anzahl der Operationen in Phase 1 und Phase 2 des M-Algorithmus im i-ten Schritt, $1 \leq i \leq n$.

Phase 1. In Schritt i des M-Algorithmus wird der RBC-Algorithmus für $H = G_i$ und $N = G_{i-1}$ benutzt, welcher nach (8.1)

$$O(|H| \log^2(|H|)\sqrt{|N|}) \tag{8.8}$$

Operationen benötigt. Die Erweiterung des Charaktergraphen von G_{i-1} auf G_i erfolgt ohne weitere wesentliche Kosten.

Phase 2. In Schritt i wurde ein $g = g_{jk}$ und ein $D \in \mathcal{D}$ fixiert. Die Zahlen $m = m(D), q = q(D), \ell = \ell(D)$, die Darstellungen F_k, $k = 1, \ldots, q(D)$, und die Darstellungen Δ_λ, $\lambda = 1, \ldots, \ell(D)$ ergeben sich unmittelbar aus dem Charaktergraphen. Ebenso können die Permutationsmatrizen P und Q_λ leicht mit Hilfe des Charaktergraphen bestimmt werden, was mit einer vernachlässigbaren Anzahl von Operationen bewerkstelligt werden kann (und sich nicht auf die Gesamtkomplexität auswirkt).
Der teure Teil ist der ET-Algorithmus. Abhängig von λ müssen Basen $\mathcal{B} = \mathcal{B}_\lambda$ konstruiert werden, deren Elemente f-blockmonomiale Matrizen mit nur einem von Null verschiedenen f-Block sind. Da die Basen $\mathcal{B}_\lambda$ bis auf die Permutationsmatrizen Q_λ

identisch sind (siehe (8.5)) und die Verkettungsmatrizen X_{gF_k} aus Schritt $i-1$ bekannt sind, können simultan alle Basen $\mathcal{B}_\lambda$ ausschließlich durch Kopieroperationen konstruiert werden, welche sich bei einer geeigneten Datenstruktur durch

$$O\left(|\mathcal{B}| \cdot f^3\right) = O\left(m^2 \cdot q \cdot f^3\right) = O\left(d^3\right), \tag{8.9}$$

abschätzen lassen. Hierbei wurde $|\mathcal{B}| = q \cdot m^2$ und $d = m \cdot q \cdot f$ benutzt. Weiterhin müssen $D^g(h_k) = D(g^{-1}h_k g)$ für $k = 1, \ldots, r$ berechnet werden. Da das Wort $g^{-1}h_k g$ ohne Kosten der pc-Präsentation entnommen werden kann und als normalisiertes Wort in $H_k \subset H$ vorliegt, können unter Benutzung der f-Blockmonomialität von D alle $D^g(h_k)$ nach (2.14) in

$$O\left(\sum_{k=1}^{r}(d \cdot f^2 \log(|H_k|)\right) = O\left(r \cdot d^3 \cdot \log(|H|)\right). \tag{8.10}$$

berechnet werden. Für D müssen insgesamt höchstens $\ell(D)$ Äquivalenztests (mit den Δ_λ, $\lambda = 1, \ldots \ell(D)$) durchgeführt werden. Aufgrund der f-Blockmonomialität von D und der Δ_λ benötigt der ET-Algorithmus für alle $\ell = \ell(D)$ Tests nach (8.4)

$$O\left(\ell \cdot |\mathcal{B}| \cdot d^2 \cdot f \cdot r \cdot p\right) \overset{(8.6)}{=} O\left(|H| \cdot r \cdot p \cdot |\mathcal{B}| \cdot f\right) = O\left(|H| \cdot r \cdot p \cdot d^2\right) \tag{8.11}$$

Operationen. Summation über alle $D \in \mathcal{D}$ ergibt aus (8.9), (8.10) und (8.11) eine Komplexitätsschranke für Phase 2 des i-ten Schritts für ein festes $g = g_{jk}$:

$$\begin{aligned} &\sum_{D \in \mathcal{D}} \left(O\left(d^3\right) + O\left(r \cdot d^3 \cdot \log(|H|)\right) + O\left(|H| \cdot r \cdot p \cdot d^2\right)\right) \\ \overset{(8.7)}{=}\ & O\left(|H|^{\frac{3}{2}} + r \cdot |H|^{\frac{3}{2}} \log(|H|) + |H|^2 \cdot r \cdot p\right) = O\left(|H|^2 \cdot r \cdot p\right). \end{aligned}$$

Da es höchstens $\log([G:H])$ Erzeuger $g = g_{jk}$, $i < j \leq n$ und $1 \leq k \leq r_j$, gibt, ergibt sich hieraus die folgende Komplexitätsschranke für Phase 2 des i-ten Schritts:

$$O\left(\log([G:H]) \cdot |H|^2 \cdot r \cdot p\right). \tag{8.12}$$

Wir fassen das bisherige Ergebnis im folgenden Lemma zusammen:

Lemma 8.4.1 *Die Anzahl der Operationen des M-Algorithmus in Schritt i, $1 \leq i \leq n$, zur Berechnung der Daten (1) und (2) zu G_i aus den Daten (1) und (2) zu G_{i-1} kann in Phase 1 durch*

$$O\left(|G_i| \log^2(|G_i|) \sqrt{|G_{i-1}|}\right)$$

und in Phase 2 durch

$$O\left(\log([G:G_i]) \cdot |G_i|^2 \cdot r_i \cdot p_i\right)$$

abgeschätzt werden. (Insbesondere ist für $i = n$ der Aufwand in Phase 2 Null.)

Summation über alle Schritte i, $1 \leq i \leq n$, ergibt die im folgenden berechnete obere Schranke für die Anzahl der Operationen des M-Algorithmus mit einer geeigneten Konstanten $\gamma \in \mathbb{R}$.

Dabei benutzen wir $[G : G_i] \cdot |G_i| = |G|$, $|G_i| \leq |G_n| \cdot 2^{i-n}$ und $\log([G : G_i]) \leq [G : G_i]$:

$$\begin{aligned}
& \sum_{i=1}^{n} \gamma \cdot \left(\log([G : G_i]) \cdot |G_i|^2 \cdot r_i \cdot p_i + |G_i| \log^2(|G_i|) \sqrt{|G_{i-1}|} \right) \\
\leq \; & \gamma \sum_{i=1}^{n} [G : G_i] |G_i| |G_n| \cdot 2^{i-n} \cdot r_i \cdot p_i + \gamma \cdot \log^2(|G_n|) \sum_{i=1}^{n} |G_n| \cdot 2^{i-n} \cdot |G_n|^{\frac{1}{2}} \cdot (2^{i-n})^{\frac{1}{2}} \\
\leq \; & \gamma |G|^2 \max\{r_i \cdot p_i | 1 \leq i \leq n\} \sum_{i=1}^{n} 2^{i-n} + \gamma |G|^{\frac{3}{2}} \log^2(|G|) \sum_{i=1}^{n} 2^{i-n} \\
\leq \; & 2\gamma \left(|G|^2 \max\{r_i \cdot p_i | 1 \leq i \leq n\} + |G|^{\frac{3}{2}} \log^2(|G|) \right).
\end{aligned}$$

Unter anderem erkennt man, daß die Komplexität der Phase 2 asymptotisch größer ist als die der Phase 1. Das Endergebnis ist im folgenden Satz zusammengefaßt, wobei wie oben erwähnt eine Operation eine Körperoperation in $\mathbb{Q}^{(e)}$ ist.

Satz 8.4.2 *Sei G eine durch eine pc-Präsentation gegebene endliche auflösbare Gruppe. Die Präsentation korrespondiere mit einer Kompositionsreihe, welche wiederum eine Hauptreihe verfeinere. Dann kann eine Transversale von gewöhnlichen irreduziblen Darstellungen von G mit*

$$O(\max\{r_i \cdot p_i | 1 \leq i \leq n\} \cdot |G|^2)$$

Operationen berechnet werden. Unter Benutzung von $r_i \leq \log(|G|)$ $1 \leq i \leq n$, erhält man die Komplexitätsschranke

$$O(p \cdot |G|^2 \log(|G|)),$$

wobei p den größten Primteiler von $|G|$ bezeichnet.

Die durch die O-Notation unterdrückten Konstanten sind nicht sehr groß, so daß die Abschätzungen auch praxisrelevante „a-priori"-Schranken liefern. Zusammenfassend gehen wir noch einmal auf zwei Punkte ein, die für die Effizienz des M-Algorithmus von entscheidender Bedeutung sind:

(1) Innerhalb zweier Untergruppen G_{i-1} und G_i der Hauptreihe sind alle auftretenden Matrizen und Darstellungen blockmonomial, wobei die Blockgrößen durch die Größen der Matrizen aus Schritt $i-1$ bestimmt sind. Berechnungen mit diesen blockmonomialen Matrizen sind wesentlich billiger als mit voll besetzten Matrizen.

(2) Da die Untergruppen G_i der Hauptreihe normal in der gesamten Gruppe G sind, operiert G auf den jeweiligen Mengen $\mathrm{Irr}(G_i)$. Dies erlaubt eine „bottom-up"-Konstruktion der jeweiligen Verkettungsmatrizen entlang der Untergruppen G_i der Hauptreihe $\mathcal{C}$, statt beispielsweise lineare Gleichungssysteme in jedem Schritt separat zu lösen.

8.5 Ausblick

In diesem Kapitel haben wir bisher nur Darstellungen über dem komplexen Zahlenkörper $\mathbb{C}$ betrachtet. Nach dem Satz von R. Brauer über Zerfällungskörper (siehe z. B. Kapitel 12

in [52]) können die gewöhnlichen Darstellungen einer endlichen Gruppe G über dem Kreisteilungskörper $\mathbb{Q}^{(e)}$ realisiert werden, wobei e den Exponenten von G bezeichnet. Obwohl $\mathbb{Q}^{(e)} = \mathbb{Q}[X]/(\Phi_e(X))$, wobei $\Phi_e(X)$ das e-te Kreisteilungspolynom bezeichnet, eine exakte Arithmetik erlaubt, können die Berechnungen in $\mathbb{Q}^{(e)}$ sehr teuer sein, da man keine Kontrolle über die Größe der Koeffizienten der Polynome hat.

Dieses Problem stellt sich nicht, wenn man über endlichen Körpern rechnet. Ist $\mathbb{K}$ ein endlicher Körper, der eine e-te primitive Einheitswurzel enthält und für den $\mathrm{char}(\mathbb{K}) \nmid |G|$ gilt, dann ist $\mathbb{K}$ ein Zerfällungskörper von G. Es sei betont, daß der M-Algorithmus über jedem solchen Körper korrekt arbeitet. Aufgrund der engen Beziehung zwischen den gewöhnlichen irreduziblen Darstellungen und den irreduziblen $\mathbb{K}$-Darstellungen kann über dem endlichen Körper $\mathbb{K}$ gearbeitet werden, um strukturelle Informationen, wie z. B. den Charaktergraph und Äquivalenz bezüglich der gewöhnlichen Darstellungen zu erhalten. Eine weiterführende Fragestellung wäre, wie man konstruktiv Darstellungen über $\mathbb{Q}^{(e)}$ aus Darstellungen über endlichen Körpern $\mathbb{K}$ gewinnen kann. Hierbei könnten „Lifting“ -Techniken als Verallgemeinerung des Henselschen Lemmas grundlegend sein.

Literaturverzeichnis

[1] Atkinson, M. D. (ed.): Computational Group Theory. Academic Press, London, 1984.

[2] Baum, U.: Existence and efficient construction of fast Fourier transforms for supersolvable groups. Computational Complexity **1** (1991), 235–256.

[3] Baum, U., Clausen, M.: Some lower and upper complexity bounds for generalized Fourier transforms and their inverses. SIAM J. Comput. **20** (1991), 451–459.

[4] Baum, U., Clausen, M.: Computing irreducible representations of supersolvable groups. Mathematics of Computation, Volume **63**, Number 207 (1994), 351–359.

[5] Bergman, G.: The Diamond Lemma for Ring Theory. Advances in Mathematics **29** (1978), 178–218.

[6] Besche, H.U., Eick, B.: The groups of order at most 1000 except 512 and 768. J. Symbolic Compututation **27** (4), (1999), 405 - 413.

[7] Beth, T.: Verfahren der schnellen Fourier Transformation. Teubner, 1984.

[8] Beth, T.: On the Computational Complexity of the General Discrete Fourier Transform. Theor. Comp. Sci. **51** (1987), 331–339.

[9] Bluestein, L.I.: A Linear Filtering Approach to the Computation of the Discrete Fourier Transform. IEEE Trans. AU-18 (1970), 451–455.

[10] Bolker, E.: The finite Radon transform. Contemporary Math. **63** (1987), 27–50.

[11] Buchberger, B., Collins, G.E., Loos, R. (ed.): Computer Algebra: Symbolic and Algebraic Computation. Springer-Verlag, 1982.

[12] Bürgisser, P., Clausen, M., Shokrollahi, M.A.: Algebraic Complexity Theory. Grundlehren der mathematischen Wissenschaften, Volume **315**, Springer Verlag, Berlin, 1997.

[13] Clausen, M.: Beiträge zum Entwurf schneller Spektraltransformationen. Habilitationsschrift, Universität Karlsruhe, 1988.

[14] Clausen, M.: Fast Generalized Fourier Transforms. Theor. Comp. Sci. **67** (1989), 55–63.

[15] Clausen, M., Baum, U.: Fast Fourier Transforms. BI-Wissenschaftsverlag, Mannheim, 1993.

[16] Clausen, M., Baum, U.: Ein kombinatorischer Zugang zur Darstellungstheorie überauflösbarer Gruppen. Bayreuther Mathematische Schriften **44** (1993), 99–107.

[17] Clausen, M., Müller, M.: A Fast Program Generator of FFTs. Proceedings AAECC-13, Honolulu, LNCS **1719** (1999), 29–42.

[18] Cooley, J.W., Tukey, J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Math. Comp. **19** (1965), 297–301

[19] Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. 2nd edition, Springer-Verlag, New York, 1996.

[20] Cox, D., Little, J., O'Shea, D.: Using Algebraic Geometry. Springer-Verlag, New York, 1998.

[21] Curtis, C., Reiner, I.: Representation Theory of Finite Groups and Associative Algebras. Interscience Publishers, Wiley & Sons, 1962.

[22] Curtis, C., Reiner, I.: Methods of Representation Theory. Volume I, Wiley & Sons, 1990.

[23] Daubechies, I.: Ten lectures on wavelets. CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, 1992.

[24] Diaconis, P.: Group Representations in Probability and Statistics. IMS, Hayward, CA, 1988.

[25] Diaconis, P.: A generalization of spectral analysis with applications to ranked data. Ann. Stat. **17** (1989), 949–979.

[26] Egner, S., Püschel, M.: AREP - A Package for Constructive Representation Theory and Fast Signal Transforms. GAP share package, 1998.
`http://avalon.ira.uka.de/home/pueschel/arep/arep.html`.

[27] Eick, B., O'Brien, E.A.: Enumerating p-groups. J. Austral. Math. Soc. (Series A) **67** (1999), 191–205.

[28] Farrelle, P.M.: Recursive Block Coding for Data Compression. Springer, 1990.

[29] Foote, R., Mirchandani, G., Rockmore, D., Healy, D., Olson, T.: A Wreath Product Group Approach to Signal and Image Processing: Part I - Multiresolution Analysis. IEEE Transaction on Signal Processing, Volume **48**, No.1 (January 2000), 102–132.

[30] Fröberg, R.: An Introduction to Gröbner Bases. John Wiley & Sons, 1997.

[31] GAP- Groups, Algorithms and Programming. GAP4, Reference Manual. Lehrstuhl D für Mathematik, RWTH Aachen, Germany, 1998.
`http://mirrors.ccs.neu.edu/GAP/NEU/`.

[32] Gauss, C.F.: Theoria Interpolationis Methodo Nova Tractata. Königliche Gesellschaft der Wissenschaften, Werke Band III (1866), Göttingen, 265–330.

[33] Hess, W.: Digitale Filter. Teubner, 1993.

[34] Higman, G.: Enumerating p-groups. Proc. London Math. Soc. (3) **10** (1960), 24–30.

[35] Huppert, B.: Endliche Gruppen I. Grundlehren der mathematischen Wissenschaften, Volume **134**, Springer Verlag, 1967.

[36] Isaacs, I.: Character Theory of Finite Groups. Academic Press, Inc., 1976.

[37] Knuth, D., Bendix, P.: Simple word problems in universal algebras. Computational Problems in Abstract Algebra, J. Leech (ed.), Pergamon Press, Oxford (1970), 263–297.

[38] Lang, S.: Algebra. Addison-Wesley, 1993.

[39] Leedham-Green, C.R., Soicher, L.H.: Collection from the left and other strategies. J. Symbolic Computation **9** (1990), 665–675.

[40] James, G., Kerber, A.: The Representation Theory of the Symmetric Group. Cambridge University Press, 1989.

[41] Mallat, S.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume **11**, No. 7 (July 1989), 674–693.

[42] Maslen, D., Rockmore, D.: Generalized FFTs - a survey of some recent results. Proceedings of DIMACS Workshop in Groups and Computation **28** (1995), 182–238.

[43] Morgenstern, J.: Complexité linéaire de calcul. Thèse, Univ. de Nice, 1978.

[44] Mora, F.: Gröbner Bases for Non-Commutative Polynomial Rings. Proceedings AAECC-3, Grenoble, LNCS **229** (1985), 353–362.

[45] Omrani, A., Shokrollahi, M.A.: Computing irreducible representations of supersolvable groups over small finite fields. Mathematics of Computation, Volume **66**, Number 218 (1997) 779–786.

[46] Plesken, W.: Towards a Soluble Quotient Algorithm. J. Symbolic Computation **4** (1987), 111–122.

[47] Püschel, M.: Konstruktive Darstellungstheorie und Algorithmengenerierung. PhD thesis, Universität Karlsruhe, Fakultät für Informatik, 1998.

[48] Püschel, M.: Decomposing Monomial Representations of Solvable Groups. Technical Report Drexel-MCS-1999-2, Dept. of Mathematics and Computer Science, Philadelphia, 1999.

[49] Rader, C.M.: Discrete Fourier Transforms when the Number of Data Samples is Prime. Proceedings of the IEEE **56** (1968), 1107–1108.

[50] Ramos, G. U.: Roundoff Error Analysis of the Fast Fourier Transform. Mathematics of Computation **25**, Number 116 (1971), 757–768.

[51] Rockmore, D.: Some applications of generalized FFTs. Proceedings of DIMACS Workshop in Groups and Computation **28** (1995), 329–370.

[52] Serre, J.P.: Linear Representations of Finite Groups. Graduate Texts in Mathematics, Springer, 1986.

[53] Shokrollahi, M.A.: Some Practical Aspects of Computational Coding and Number Theory. Habilitationsschrift, Universität Bonn, 1997.

[54] Sims, C.C.: Enumerating p-groups. Proc. London Math. Soc. (3) **15** (1965), 151–166.

[55] Sims, C.C.: Computation with finitely presented groups. Cambridge University Press, 1994.

[56] Sims, C.C.: Fast multiplication and growth in groups. ISSAC'98, Rostock, Germany (1998), 165–170.

[57] Steidl, G.: Spline Wavelets over $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{R}/N\mathbb{Z}$, and $\mathbb{Z}/N\mathbb{Z}$. Wavelets: Theory, Algorithms, and Applications. Edited by Chui, C., Montefuso, L. Puccio, L. Academic Press, 1994, 155–177.

[58] Stoer, J.: Numerische Mathematik. Springer Verlag, 1993.

[59] Strassen, V.: Gaussian elimination is not optimal. Num. Math. **13** (1969), 354–356.

[60] Strang, G.: Wavelet Transforms versus Fourier Transforms. Bulletin of the AMS **28**, No. 2 (1993), 288–305.

[61] Sturmfels, B., Wiesmantel, R., Ziegler, G.: Gröbner Bases of Lattices, Corner Polyhedra and Integer Programming. Beiträge zur Algebra und Geometrie **39** (1995), 281–298.

[62] Thümmel, A.: Computing character tables of p-groups. Proceedings ISSAC'96, Zürich, Switzerland (1996), 150–154.

[63] Vaughan-Lee, M.R.: Collection from the left. J. Symbolic Computation **9** (1990), 725–733.

[64] Vaidyanathan, P.P.: Multirate Systems and Filter Banks. Prentice-Hall, 1993.

[65] Wickerhauser, M.V.: Adaptive Wavelet-Analysis. Vieweg, 1993.

[66] Zwicker, E., Fastl, H. Psychoacoustics. Springer, 1990.

Index

Lebenslauf

Name:	Meinard Müller	
Geburtsdatum:	19. August 1969	
Geburtsort:	Esslingen am Neckar	
Eltern:	Dr. Irmin Müller geb. Imhof	
	Hans-Georg Müller	
Familienstand:	ledig	
Schulausbildung:	1975-1979	Grundschule Dagersheim in Böblingen
	1980-1988	Otto-Hahn-Gymnasium in Böblingen
	1988	Abitur
Zivildienst:	1988-1990	Verein für Körperbehinderte in Sindelfingen
Studium:	1990-1993	Mathematik mit Nebenfach Informatik an der Universität Bonn
	1993-1994	Einjähriges Auslandsstudium an der Louisiana State University in Baton Rouge/USA
	1994-1997	Fortsetzung des Studiums an der Universität Bonn
	1997	Diplom in Mathematik
	1997-1998	Einjähriges Japanischstudium an der Keio Universität in Tokyo/Japan
	seit 1998	Promotionsstudium in Informatik an der Universität Bonn