Zur algorithmischen Zerlegungstheorie linearer Transformationen mit Symmetrie

Zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften der Fakultät für Informatik der Universität Karlsruhe (Technische Hochschule)

vorgelegte

Dissertation

von

Sebastian Egner

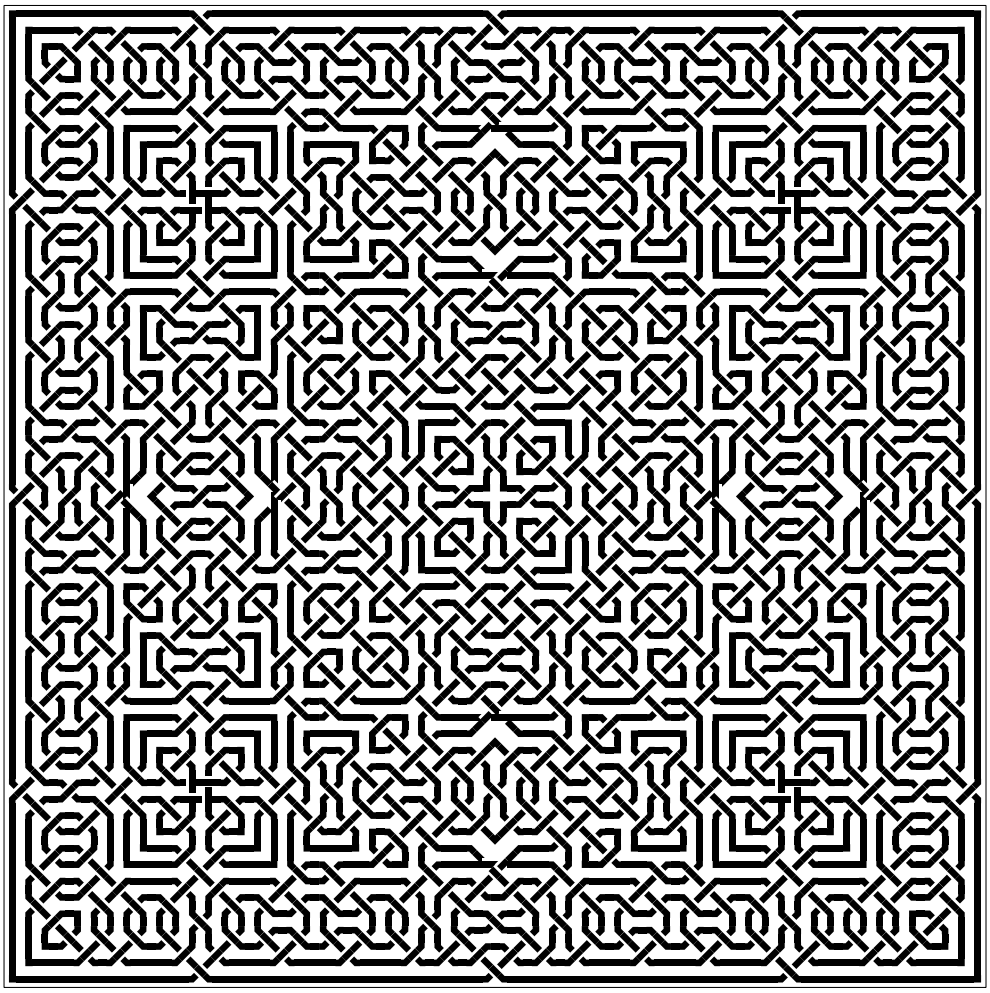
aus Hamburg

Tag der mündlichen Prüfung:	11. Juli 1997
Erster Gutachter:	Prof. Dr. Thomas Beth
Zweiter Gutachter:	Prof. Dr. Roland Vollmar

22. Dezember 1997

Diese Dissertation entstand im Rahmen des mir gewährten Stipendiums im Graduiertenkolleg „Beherrschbarkeit komplexer Systeme“ an der Fakultät für Informatik der Universität Karlsruhe. Mein herzlicher Dank gilt Herrn Professor Dr. Thomas Beth für die mir gewährte Unterstützung und für viele inspirierende Diskussionen zur Lösung komplexer wissenschaflicher Probleme und Herrn Professor Dr. Roland Vollmar, der es durch sein Verständnis und Interesse schafft, dem Graduiertenkolleg eine Atmosphäre des gegenseitigen Austauschs zu geben. Weiterhin gilt mein Dank Herrn Dipl.-Math. Markus Püschel und Herrn Dr. Torsten Minkwitz für die freundschaftliche und effektive Zusammenarbeit, ohne die diese Arbeit nicht in der vorliegenden Form existieren könnte. Weiterer Dank gilt den Herren Dipl.-Inf. Jörn Müller-Quade, Dr. Peter Sanders und Martin Rötteler für Anregungen und interessante Diskussionen.

Struktur und Komplexität



Wandornament in der „Halle der zwei Schwestern“ aus der Alhambra in Granada. Das Ornament bedeckt eine Fläche von etwa zwei Quadratmetern und entstand vermutlich in der zweiten Hälfte des 14. Jahrhunderts zur Zeit der letzten maurischen Dynastie, dem Nasridenreich.

Das Muster entstammt der Dissertation „Gruppentheoretische und Strukturanalytische Untersuchungen der Maurischen Ornamente aus der Alhambra in Granada“ von Edith Müller bei Andreas Speiser in Zürich, 1944. Die hier dargestellte Abbildung wurde von einem kleinen Programm des Autors erzeugt. Das Ornament entsteht durch Symmetrisierung eines Fundamentalbereichs unter der Diedergruppe der Ordnung 8 bei gleichzeitiger Inversion der topologischen Relation zwischen den Bändern.

Inhalt

1

Weshalb Symmetrieanalyse?

Bei vielen wichtigen Operationen der digitalen Signalverarbeitung, wie etwa der diskreten Fourier-Transformation, finden seit langem schnelle Algorithmen Verwendung. Wie aber sind deren Entdecker zu diesen Algorithmen gelangt? Hätte man diese auch automatisch finden können?

Die vorliegende Arbeit ist einem systematischen Zugang zu diesen Fragen gewidmet: *Wie kann die Struktur einer linearen Transformation aus einer definierenden Beschreibung automatisch bestimmt werden?* Das Ziel hierbei ist, für eine bestimmte Klasse von Signaltransformationen schnelle Algorithmen automatisch zu konstruieren und eine wesentliche Struktureinsicht zu gewinnen. Die Kenntnis einer solchen Struktur kann nicht nur zur Algorithmengenerierung verwendet werden, sondern auch zur Vereinfachung nichtlinearer Gleichungen. Ein frühes Beispiel für diese Art symmetriebasierter Vereinfachung stammt von C. E. Shannon: In seiner Diplomarbeit definiert Shannon (1938) den Begriff der symmetrischen booleschen Funktion und zeigt, daß sich die in diesem Sinne symmetrischen Funktionen geschickt durch elementare Bausteine (Relais) realisieren lassen, [84]. In heutiger Ausdrucksweise gesagt, hat Shannon den Invariantenring $\mathbb{F}_2[X_1, \ldots, X_n]^{\mathsf{S}_n}$ betrachtet.

Der Ansatz zur Konstruktion schneller Algorithmen, der in dieser Arbeit verfolgt wird, ist die „symmetriebasierte Algorithmengenerierung“ (Beth): Zu einer gegebenen linearen Transformation, spezifiziert durch eine Matrix, wird zunächst eine Symmetriegruppe bestimmt. Dann wird zu der Gruppe eine angepaßte Spektraltransformation synthetisiert. Im letzten Schritt wird die gegebene Transformation durch die konstruierte Spektraltransformation ausgedrückt. Der Informationsfluß der symmetriebasierten Algorithmengenerierung ist in der Abbildung 1.1 dargestellt. Die einzelnen Schritte sind

Suche Gegeben ist die lineare Transformation M. Zu dieser wird eine endliche Symmetriegruppe G durch kombinatorische Suche bestimmt. Die Symmetriegruppe erfaßt einen Teil der in M vorhandenen Redundanz. Die Bestimmung von Symmetriegruppen aus einer Matrix M ist das zentrale Thema dieser Arbeit.

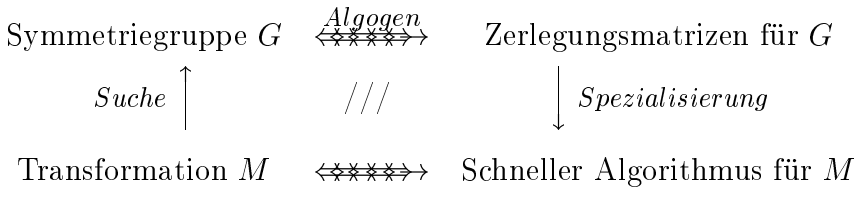


Abbildung 1.1: Symmetriebasierte Algorithmengenerierung

Algogen Die Symmetriegruppe G wird aufgefaßt als Paar von Darstellungen ihrer selbst. (Zum Begriff der Darstellung einer Gruppe siehe Anhang A.) Die beiden Darstellungen werden schrittweise in irreduzible Komponenten zerlegt, wobei jeder einzelne Zerlegungsschritt mit einer dünn besetzten Matrix ausgeführt wird. Dieser Prozeß des schrittweisen Ausreduzierens mit Hilfe dünn besetzter Matrizen soll *Algogen* heißen. (Diese Bezeichnung wurde von Minkwitz (1993) eingeführt.) Für folgende Klassen von Darstellungen sind effiziente *Algogen*-Verfahren bekannt: Reguläre Darstellungen, beliebige Permutationdarstellungen und monomiale Darstellungen. Das Ergebnis des „*Algogen*"-Schritts sind zwei Produkte dünn besetzter Matrizen. Diese beiden Matrizen zerlegen G in irreduzible Komponenten und machen damit die zuvor versteckte Symmetrie explizit.

Spezialisierung Die Matrix M wird durch die Symmetriegruppe G nicht vollständig bestimmt, und im „Algogen"-Schritt wurde nur G zerlegt. Es ist daher notwendig, eine Spezialisierung auf die gegebene Matrix M vorzunehmen. Aus algebraischen Gründen ist die dabei auftretende Matrix B blockdiagonal (eine Folge des Lemmas von Schur), und die Blöcke sind um so kleiner, je geringer die Grade der irreduziblen Komponenten von G sind und je verschiedener die Komponenten sind. Die Matrix B enthält den nicht durch die Symmetriegruppe G beschriebenen Anteil der Information in M. Insgesamt wird die Matrix M faktorisiert in

$$M = A_L^{(1)} \cdots A_L^{(r)} \cdot B \cdot (A_R^{(s)})^{-1} \cdots (A_R^{(1)})^{-1},$$

wobei A_L und A_R die im „Algogen"-Schritt konstruierten Zerlegungsmatrizen sind. Eine solche Matrix-Zerlegung von M ist der schnelle Algorithmus für M. Eine Veranschaulichung des schnellen Algorithmus ist der Signalflußgraph. Ein Beispiel für einen Signalflußgraphen ist in Abbildung 1.2 auf Seite 15 dargestellt.

Das zuvor vorgestellte Verfahren zur symmetriebasierten Algorithmengenerierung wird in Abschnitt 1.6 ausführlich beschrieben. Doch zunächst zur Entstehungsgeschichte der „symmetriebasierten Algorithmengenerierung". Die Verwendung von Symmetrie zur Lösung von Gleichungen hat eine lange Tradition in der Physik. Der unmittelbare Vorläufer der Algorithmengenerierung mit Hilfe von Symmetrie ist die schnelle Fourier-Transformation (FFT).

1.1 Die Entwicklung der verallgemeinerten FFT

∼1800	Gauss	Verwendet FFT in der Himmelsmechanik
1965	Cooley Tukey	FFT für Produktlängen, speziell Zweierpotenzen
1968	Rader	FFT auf Primzahllänge
1970	Bluestein	FFT durch Einbettung und Faltung
1970	Apple Wintz	Definition einer FT auf abelschen Gruppen
1971	Good Thomas	FFT durch multidimensionale FT
1975	Pichler	FFT auf abelschen Gruppen
1977	Karpovsky Trachtenberg	Definition einer FT auf nichtabelschen Gruppen; FFT für direkte Produkte
1978	Winograd	FFT durch Interpolation
1982	Nussbaumer	Umfassende Aufwandsanalyse der FFT
1984	Beth	Darstellungstheoretische Sichtweise der FT; schnelle Verfahren für auflösbare Gruppen; FT bei Wechsel des Grundkörpers (ADFT)
1988	Clausen	Komplexitätsanalyse der FFT; FFT auf metabelschen und symmetrischen Gruppen; Symmetrieadaption
1988	Baum	FFT für überauflösbare Gruppen
1990	Diaconis Rockmore	Neue Anwendungen der FFT in Stochastik und Statistik; FFT symmetrischer Gruppen (unabhängig von Clausen)
1992	Oberst Walch	Multiplikationsoptimale FFT abelscher Gruppen
1993	Linton Michler Olsson	FFT für M-Gruppen
1993	Minkwitz	Übergang von der FT zu beliebigen Matrizen mit Symmetrie; schnelle Algorithmen für Permutationsdarstellungen auflösbarer Gruppen
1994	Coppersmith	FFT für zyklische Gruppen von Zweierpotenzordnung im Maschinenmodell des Quantenregisters
1995	Maslen Rockmore	FFT auf kompakten unendlichen Gruppen
1997	Beals	FFT für symmetrische Gruppen im Maschinenmodell der Quanten-Turingmaschine

Im Rahmen seiner Habilitationsschrift [10] erkannte Th. Beth (1984), daß sich fast alle klassischen Algorithmen der FFT als Zerlegungsoperationen der Gruppenalgebra $\mathbb{C}[\mathsf{Z}_n]$ auffassen lassen. Durch diese fundamentale Einsicht konnte er die schnellen Algorithmen von Cooley-Tukey, Rader, Winograd und Good-Thomas durch universelle algebraische Konstruktionen neu begründen. Damit war eine Korrespondenz etabliert zwischen der regulären Darstellung einer zyklischen Gruppe und den klassischen FFT-Algorithmen: Der Signalflußgraph einer schnellen Transformation entspricht, *cum grano salis*, der Untergruppenstruktur einer endlichen Gruppe.

Die neue Sichtweise der klassischen numerischen FFT-Methoden reicht zudem weit über deren algebraische Begründung hinaus. Insbesondere ist es möglich geworden, umgekehrt vorzugehen: Aus der Struktur der Gruppenalgebra $\mathbb{C}[\mathsf{Z}_n]$ kann ein schneller Algorithmus zur FFT synthetisiert werden. Wie Beth (1984) gezeigt hat, kann die FFT auf einer auflösbaren Gruppe G in $O(|G| \log |G|)$ vielen Körperoperationen ausgewertet werden, [10]. Die Darstellungstheorie endlicher Gruppen erweist sich damit als die natürliche Beschreibungsform für Algorithmen der schnellen Fourier-Transformation.

Die bereits von Karpovsky und Trachtenberg (1977) in [49] definierte Fourier-Transformation auf einer endlichen Gruppe konnte nun systematisch untersucht werden. Sei ρ eine Darstellung der endlichen Gruppe G und $f : G \to \mathbb{C}$ eine komplexwertige Funktion auf G. Dann ist das Bild der Fourier-Transformierten $\hat{f}$ an der Stelle ρ gegeben durch

$$\hat{f}(\rho) = \sum_{g \in G} \rho(g) f(g).$$

Die Fourier-Transformation ordnet also einer Funktion f und einer Darstellung ρ die $(\deg(\rho) \times \deg(\rho))$-Matrix $\hat{f}(\rho)$ zu. Ist das Bild von $\hat{f}$ auf einem Repräsentantensystem $\rho_1, \ldots, \rho_h$ der irreduziblen Darstellungen von G bekannt, so kann f daraus rekonstruiert werden als

$$f(g) = \frac{1}{|G|} \sum_{i=1}^{h} \deg(\rho_i) \operatorname{tr}\left(\rho_i(g^{-1}) \hat{f}(\rho_i)\right) \quad \text{für } g \in G.$$

In dem Spezialfall der zyklischen Gruppe $G = \mathsf{Z}_n$ ergibt sich die klassische Fourier-Transformation, denn die irreduziblen Darstellungen von Z_n sind genau die harmonischen Funktionen auf n Punkten.

Aufbauend auf dem neuen Begriff der Fourier-Transformation entstanden in der Zeit zwischen 1984 und 1993 eine große Anzahl von Ergebnissen bezüglich des Berechnungsaufwandes der FFT auf einer endlichen Gruppe. Inbesondere beschäftigte sich Clausen (1988) in seiner Habilitationsschrift an der Universität Karlsruhe mit der Komplexität der FFT, [17]. Als zentrale Erkenntnis definierte Clausen in seiner Arbeit den wichtigen Begriff der Symmetrieadaption. Unabhängig von Clausen und Beth publizierten Diaconis und Rockmore (1990) in den

USA ähnliche Ergebnisse, [26]. Diaconis und Rockmore wandten ihre Ergebnisse zur FFT konsequent auf Probleme der Stochastik und Statistik an. So zeigten sie unter anderem, daß ein Markov-Prozeß auf einem Cayley-Graphen einer endlichen Gruppe durch Verwendung der FFT analysiert werden kann. Durch die Arbeiten von Diaconis und Rockmore sowie der von Clausen und Beth wurde die algebraische Begriffsbildung der Fourier-Transformation und die Existenz schneller Algorithmen zu ihrer Berechnung einer breiten Öffentlichkeit bekannt.

Besonders intensiv wurde der Spezialfall der FFT auf symmetrischen Gruppen untersucht. Die von Clausen (1989), von Diaconis und Rockmore (1990), sowie von Linton, Michler und Olsson (1991) publizierten Verfahren ermöglichen die schnelle Zerlegung eines Elements aus $\mathbb{C}[\mathsf{S}_n]$ bezüglich irreduzibler Komponenten, [11, 18, 26, 58, 48, 27]. Die Eingabe dieser FFT ist ein Vektor mit $n!$ vielen komplexen Zahlen. Kürzlich hat Beals (1997) nachgewiesen, daß die FFT auf einer symmetrischen Gruppe auch im Maschinenmodell der Quanten-Turingmaschine schnell berechnet werden kann. Die Anzahl der benötigten Quantenoperationen für die FFT auf der Gruppe G ist dabei beschränkt durch ein Polynom in $\log|G|$, [6]. Diese Arbeit erweitert das Ergebnis von Coppersmith (1994), der die Zerlegung der FFT von Beth (1984) im Aufwandsmaß eines Quantenregisters neu interpretiert hat. Dadurch wurde nachgewiesen, daß die FFT auf einer zyklischen Gruppe der Länge n in $O(\log(n)^2)$ Operationen auf einer Quanten-Registermaschine ausgeführt werden kann, [23], [10]. Die FFT auf einem Quantenrechner ist ein wesentlicher Bestandteil des Faktorisierungsalgorithmus von Shor (1994), [85].

Der andere besonders gut untersuchte Spezialfall ist die FFT auf einer M-Gruppe. Eine Gruppe G heißt M-Gruppe, wenn jede Darstellung von G ähnlich ist zu einer monomialen Darstellung. Jede überauflösbare Gruppe ist eine M-Gruppe und jede M-Gruppe ist auflösbar (Huppert, Bd. I, Kap. V, §18, Satz 18.5a, Satz 18.6b). Da sich monomiale Darstellungen im Rechner besonders kompakt abspeichern lassen, bestand die Hoffnung, alle irreduziblen Darstellungen einer M-Gruppe effizient zu konstruieren. Dies gelang schließlich Baum (1988) für den Spezialfall der überauflösbaren Gruppen durch Anwendung der Clifford-Theorie, [5]. Ein ähnliches Verfahren schlugen Linton, Michler und Olsson (1993) vor, um monomiale Darstellungen für die FFT zu nutzen, [59].

Zum Thema der FFT auf einer endlichen Gruppe und ihrer Anwendungen sind inzwischen einige zusammenfassende Arbeiten erschienen, allen voran das Buch von Clausen und Baum (1993), [19]. Eine andere Sicht der Dinge stellten Maslen und Rockmore (1995) in einem Übersichtsartikel dar, [61, 81].

1.2 „Beyond FFT“ — eine andere Sichtweise

Eine völlig neue Qualität der Konstruktivität und Allgemeinheit bei der Algorithmengenerierung mit Hilfe von Symmetrie ist mit der Dissertation von Minkwitz

(1993) an der Universität Karlsruhe erreicht worden, [64]. Statt „die“ FFT einer Gruppe zu betrachten, verwendete Minkwitz die gruppentheoretischen Methoden, um eine vorgegebene Matrix zu zerlegen. Er schaffte damit den Schritt von einem methodenorientierten Ansatz zu einem problemorientierten. Die eingangs erläuterte Methode der „symmetriebasierten Algorithmengenerierung“ geht von einer Matrix aus und zerlegt *genau diese* Matrix. Die früheren Arbeiten von Beth, Clausen, Diaconis, Rockmore und anderen beginnen bei einer Gruppe und definieren erst aus der Gruppe eine Matrix, die dann geeignet zerlegt wird. Durch die Arbeit von Minkwitz wird eine weitaus größere Klasse von Matrizen einer effizienten Behandlung mit algebraischen Methoden zugänglich. Insbesondere gelang es Minkwitz und Beth (1993), einen schnellen Algorithmus für die diskrete Cosinus-Transformation, Typ IV, anzugeben.

In algebraischer Sprechweise ausgedrückt, hat Minkwitz den Übergang von der Zerlegung der Gruppenalgebra zur Zerlegung einer *Darstellung* vollzogen. Vor Minkwitz wurde einer Gruppe G die Fourier-Transformation $f \mapsto \hat{f}$ zugeordnet. Dieses Prinzip wurde im vorigen Abschnitt erläutert. Minkwitz dagegen ordnet einer Matrix $M \in \mathsf{GL}_n(\mathbb{C})$ ein Paar von Darstellungen Π_L und Π_R der gleichen Gruppe G zu, so daß

$$\Pi_L(g) \cdot M = M \cdot \Pi_R(g) \quad \text{für alle } g \in G.$$

Eine Matrix mit dieser Eigenschaft wird (Π_L, Π_R)-symmetrisch genannt. Ist G auflösbar, Π_L eine Permutationsdarstellung und Π_R eine direkte Summe irreduzibler Darstellungen, so kann zu jeder (Π_L, Π_R)-symmetrischen Matrix ein schneller Algorithmus konstruiert werden (Minkwitz). Die Konstruktion des schnellen Algorithmus basiert bei Minkwitz auf der Clifford-Theorie.

Durch die Verallgemeinerung auf *Darstellungen* der Gruppenalgebra wird die nutzbare Struktur einer Transformation viel stärker: Im Extremfall $G = \mathsf{S}_n$ hat das Signal nur n Komponenten, aber es gibt $n!$ Gruppenelemente. Die verallgemeinerte FFT einer endlichen Gruppe entspricht dagegen dem Spezialfall der regulären Darstellung von G. In diesem Fall besteht das Signal immer aus $|G| = n!$ vielen Komponenten.

Das symmetriebasierte Verfahren zur Algorithmenkonstruktion ist immer dann erfolgreich, wenn die auftretenden irreduziblen Komponenten hinreichend kleinen Grades sind und wenn sie hinreichend verschieden sind. Kombinatorisch gesehen ist das nie der Fall: Die $(n \times n)$-Matrizen bilden einen Vektorraum der Dimension n^2. Ein schneller Algorithmus ist ein Produkt mit $O(\log n)$ vielen Faktoren, von denen jeder $O(n)$-dünn besetzt ist. Ein schneller Algorithmus besitzt daher nur $O(n \log n)$ viele Freiheitsgrade. Folglich gilt: *Fast alle Matrizen können keine nutzbare Symmetrie besitzen.* Es hat sich jedoch gezeigt, daß viele wichtige Klassen von Transformationen eine starke, nutzbare Struktur besitzen — diese muß allerdings explizit bekannt sein, obwohl sie im Prinzip in der Spezifikation inhärent enthalten ist. Dies ist der Ausgangspunkt der vorliegenden Arbeit:

Für die symmetriebasierte Algorithmengenerierung ist die explizite Kenntnis der Gruppen und der Darstellungen notwendig. Zu einer gegebenen linearen Transformation soll daher mit automatischen Methoden eine Symmetriegruppe konstruiert werden, die geeignet ist, die Transformation strukturell zu zerlegen.

1.3 Symmetrie-Operation, -Gruppe und -Typ

Der Begriff der *Symmetrie* steht im Mittelpunkt der vorliegenden Arbeit. Bisher wurde er nur im informellen Sinne gebraucht, um die nutzbare Strukturinformation einer linearen Transformation zu bezeichnen. Das Konzept der Symmetrie soll nun formalisiert werden. Dazu ist es gute Tradition, die folgenden drei Begriffe sorgfältig getrennt im Gedächtnis zu verwahren:

- **Symmetrieoperation** Eine Symmetrieoperation s überführt das Objekt X in sich selbst. Ist X zum Beispiel ein Quadrat in der Ebene, so ist die Drehung um $90°$ eine Symmetrieoperation, ebenso wie die Spiegelung an der y-Achse. Ein anderes Beispiel: X sei die Cosinus-Funktion und s die Abbildung, die durch $s(f) = x \mapsto \Leftrightarrow f(x + \pi)$ definiert ist. In diesem Fall ist s eine Symmetrieoperation von X, denn $\cos(x + \pi) = \Leftrightarrow\cos(x)$.

- **Symmetriegruppe** Die Symmetriegruppe eines Objektes X ist die Menge aller möglichen Symmetrieoperationen von einem bestimmten Symmetrietyp. Die Multiplikation der Gruppe ist die Nacheinanderausführung der Operationen. Die Symmetriegruppe wird manchmal auch kurz die Symmetrie genannt. Ist X ein Quadrat in der Ebene, so enthält die Symmetriegruppe acht Symmetrieoperationen (die Identität, drei echte Drehungen und vier Spiegelungen). Das andere Beispiel: Zu $X = \cos$ und s wie oben ist die Symmetriegruppe unendlich groß, weil $s^z(f) = x \mapsto (\Leftrightarrow 1)^z f(x + z\pi)$ für jedes $z \in \mathbb{Z}$ eine andere Symmetrieoperation ist.

- **Symmetrietyp** Ein Symmetrietyp ist eine vorgegebene Klasse von Operationen ähnlicher Beschaffenheit. Beispielsweise sind alle Symmetrieoperationen des Quadrats Isometrien der Ebene. Man könnte daher sagen, die Symmetrie des Quadrats sei vom Typ „Isometrie“. Das andere Beispiel: Die Operation s ist innen eine Translation um π und außen eine Skalierung mit $\Leftrightarrow 1$. Man könnte s daher einem Translations-Skalierungs-Typ zuordnen.

Nach dieser allgemeinen Begriffsbestimmung geht es im kommenden Abschnitt um die für diese Arbeit relevanten Formen von Symmetrie. Da in erster Linie schnelle Algorithmen für lineare Transformationen gefunden werden sollen, werden lineare Symmetrien von Matrizen betrachtet.

1.4 Symmetrie linearer Transformationen

In diesem Abschnitt wird ein allgemeiner Rahmen für die in dieser Arbeit betrachteten Typen von Symmetrie entwickelt. Gegeben ist eine Matrix $M \in \mathrm{K}^{n\times m}$. Die Grundmenge K ist ein effektiver Körper, also ein Körper, in dem die Arithmetik und der Vergleich $x = 0$ berechenbar sind. Für manche speziellen Typen von Symmetrien ist eine geringere algebraische Struktur ausreichend. Die folgenden allgemeinen Überlegungen bleiben dabei jedoch weiterhin gültig. Um die Symmetriegruppe von M zu definieren, muß ein Symmetrietyp vorgegeben werden. Dies stellt eine vorherige Einschränkung der möglichen Operationen auf der Matrix M dar. In der Folge kann die Symmetriegruppe von M in dem vorgegebenen Symmetrietyp definiert werden.

1.1 Definition *Gegeben sei eine Matrix $M \in \mathrm{K}^{n\times m}$ sowie eine Untergruppe $T \leq \mathsf{GL}_n(\mathrm{K}) \times \mathsf{GL}_m(\mathrm{K})$. Die **Symmetrie** von M vom Typ T ist die Menge*

$$\mathsf{Sym}_T(M) = \left\{ (L,R) \in T \;\middle|\; L \cdot M = M \cdot R \right\}.$$

Die Matrizen L beschreiben Zeilenoperationen auf der Matrix M, die Matrizen R operieren auf den Spalten. Ein Paar $(L,R) \in \mathsf{Sym}_T(M)$ beschreibt daher Zeilenoperationen L, die durch Spaltenoperationen R^{-1} wieder rückgängig gemacht werden können, denn $LMR^{-1} = M$.

Aus der allgemein gehaltenen Definition von $\mathsf{Sym}_T(M)$ werden durch Spezialisierung des Symmetrietyps T alle besonderen Typen von Symmetrie abgeleitet. Zum Beispiel ist $\mathsf{Sym}_{\mathsf{S}_n\times\mathsf{S}_m}(M)$ die Perm-Perm-Symmetrie von M, denn es werden nur noch Permutationsmatrizen zugelassen. Die Konvention für L und R ist so gewählt worden, daß sie zur Definition des Intertwining-Raumes (Definition 1.6) paßt und sich leicht merken läßt. Fundamental für alles Weitere ist die folgende Aussage.

1.2 Lemma Es seien M und T wie in Definition 1.1. Dann ist $\mathsf{Sym}_T(M)$ eine Untergruppe von T mit der folgendermaßen definierten Multiplikation

$$(L_1,R_1)\cdot(L_2,R_2) = (L_1L_2, R_1R_2).$$

Beweis Die Menge $\mathsf{Sym}_T(M)$ ist abgeschlossen unter der Multiplikation, wie die folgende Rechnung zeigt

$$L_1M = MR_1 \text{ und } L_2M = MR_2 \Rightarrow L_1L_2M = L_1MR_2 = MR_1R_2.$$

Analog kann die Abgeschlossenheit unter Inversion gezeigt werden und die Tatsache, daß die Menge nicht leer ist. ∎

Es liegt nahe, zuerst die abstrakte Struktur der Symmetriegruppen zu klären. Konkret: Wie setzt sich die Symmetriegruppe aus dem linken und dem rechten

Teil zusammen? Für die Klärung dieser Frage wird der Begriff des subdirekten Produkts mit vereinigter Faktorgruppe benötigt (Huppert (1983), [43], Bd. I, Kap. I, §9).

1.3 Definition *Seien $N_L \trianglelefteq G_L$ und $N_R \trianglelefteq G_R$ Gruppen mit den isomorphen Faktorgruppen $\varphi : G_L/N_L \overset{\cong}{\Leftrightarrow} G_R/N_R$. Dann ist*

$$G_L \curlywedge G_R = \left\{ (g_L, g_R) \in G_L \times G_R \;\middle|\; \varphi(g_L N_L) = g_R N_R \right\}$$

eine Untergruppe von $G_L \times G_R$. Sie wird als ***subdirektes Produkt*** *von G_L und G_R mit vereinigter Faktorgruppe $G_L/N_L \cong G_R/N_R$ bezeichnet. Im Falle endlicher Gruppen ist die Ordnung des subdirekten Produkts gegeben durch*

$$|G_L \curlywedge G_R| = \frac{|G_L| \cdot |G_R|}{|G_L/N_L|}.$$

Mit dieser Begriffsbildung kann die abstrakte Struktur einer Symmetriegruppe vollständig geklärt werden.

1.4 Satz *Es seien M und T wie in Definition 1.1 und $G = \mathsf{Sym}_T(M)$. Die Abbildung $\Pi_L = (L, R) \mapsto L$ bezeichne die Projektion von G auf die linke Komponente. Analog sei $\Pi_R = (L, R) \mapsto R$. Dann gilt*

$$G = \Pi_L(G) \curlywedge \Pi_R(G) \quad \textit{mit} \quad \Pi_L(G)/\Pi_L(\ker \Pi_R) \cong \Pi_R(G)/\Pi_R(\ker \Pi_L).$$

Beweis Die Projektionen Π_L und Π_R sind Gruppenhomomorphismen. Daher sind ihre Kerne Normalteiler von G. Daraus folgt, daß auch ihre Projektionen auf die jeweils andere Komponente Normalteiler sind. Das sind die Gruppen

$$\begin{aligned} \Pi_L(\ker \Pi_R) &= \{L \mid (L, \mathbf{1}) \in G\} \quad \text{und} \\ \Pi_R(\ker \Pi_L) &= \{R \mid (\mathbf{1}, R) \in G\}. \end{aligned}$$

Es bleibt zu zeigen, daß die Faktorgruppen isomorph sind. Zur Abkürzung sei $N_L = \Pi_L(\ker \Pi_R)$ und $N_R = \Pi_R(\ker \Pi_L)$. Wird nun ein Gruppenisomorphismus φ definiert durch

$$\varphi(LN_L) = RN_R \quad \text{für } (L, R) \in G,$$

dann gilt

1. φ ist wohldefiniert: Man betrachte $(L_1, R_1), (L_2, R_2) \in G$. Dann ist auch $(L_1^{-1}L_2, R_1^{-1}R_2) \in G$, das heißt $\varphi(L_1^{-1}L_2 N_L) = R_1^{-1}R_2 N_R$. Ist nun zusätzlich $L_1 N_L = L_2 N_L$, so folgt $L_1^{-1}L_2 \in N_L$, das heißt $(L_1^{-1}L_2, \mathbf{1}) \in G$ und daher $\varphi(L_1^{-1}L_2 N_L) = \mathbf{1}N_R$. Dann gilt $R_1 N_R = R_2 N_R$. Es kommt also auf die Wahl des Vertreters nicht an.

2. φ ist ein Gruppenhomomorphismus: Man betrachte $(L_1, R_1), (L_2, R_2) \in G$. Dann ist auch $(L_1L_2, R_1R_2) \in G$ und es gilt

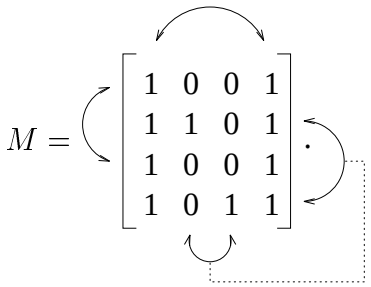
$$\varphi(L_1N_L \cdot L_2N_L) = \varphi(L_1L_2N_L) = R_1R_2N_R = \varphi(L_1N_L) \cdot \varphi(L_2N_L).$$

3. φ ist surjektiv und injektiv: Die Surjektivität folgt sofort aus der Definition von φ. Für die Injektivität betrachte man ein $(L, R) \in G$ mit $LN_L \in \ker \varphi$. Daraus folgt

$$\begin{aligned} RN_R = \mathbf{1}N_R \quad &\Leftrightarrow \quad R \in N_R \Leftrightarrow (\mathbf{1}, R) \in G \\ &\Rightarrow \quad (\mathbf{1}, R)^{-1} \cdot (L, R) = (L, \mathbf{1}) \in G \\ &\Rightarrow \quad L \in N_L \Leftrightarrow LN_L = \mathbf{1}N_L. \end{aligned}$$

Der Kern von φ ist also trivial und damit ist φ injektiv. ∎

1.5 Beispiel Bei der folgenden (4×4)-Matrix sind drei mögliche Zeilen- und Spaltenpermutationen eingezeichnet (gestrichelt = simultan).



Die dargestellten Permutationen sind Erzeuger der Perm-Perm-Symmetrie $G = \mathsf{Sym}_{\mathsf{S}_4 \times \mathsf{S}_4}(M)$. Die Gruppe besitzt 8 Elemente. Notiert als Paare von Permutationen lauten die Erzeuger

$$G = \Big\langle \Big((1\,3), \mathrm{id}\Big), \Big(\mathrm{id}, (1\,4)\Big), \Big((2\,4), (2\,3)\Big) \Big\rangle \cong \mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{Z}_2.$$

Die Projektionen von G und von den Normalteilern sind

$$\begin{aligned} \Pi_L(G) &= \langle (1\,3), (2\,4) \rangle \cong \mathsf{Z}_2 \times \mathsf{Z}_2, \\ \Pi_R(G) &= \langle (1\,4), (2\,3) \rangle \cong \mathsf{Z}_2 \times \mathsf{Z}_2, \\ \Pi_L(\ker \Pi_R) &= \langle (1\,3) \rangle \quad \text{und} \\ \Pi_R(\ker \Pi_L) &= \langle (1\,4) \rangle. \end{aligned}$$

Die Faktorgruppen sind

$$\langle (1\,3), (2\,4) \rangle / \langle (1\,3) \rangle \cong \mathsf{Z}_2 \cong \langle (1\,4), (2\,3) \rangle / \langle (1\,4) \rangle.$$

Die Elemente von (L, R) von G vertauschen simultan die Zeilen und die Spalten der Matrix M. Durch die Projektionen Π_L und Π_R wird eine Vertauschung getrennt in ihre Wirkung auf die Zeilen und in ihre Wirkung auf die Spalten. Die Normalteiler in den Projektionen sind genau diejenigen einseitigen Vertauschungen, die ohne Beachtung der anderen Seite eine Symmetrieoperation bewirken. Satz 1.4 besagt nun, daß die zugehörigen Faktorgruppen isomorph sind. ∎

1.5 Transformationen gegebener Symmetrie

Mit Definition 1.1 wird einer Matrix M eine Symmetriegruppe G zugeordnet. Es liegt daher nahe, umgekehrt zu fragen: Welche Matrizen besitzen die gegebene Symmetrie G?

1.6 Definition *Sei G eine endliche Gruppe, K ein Körper und $\rho : G \to \mathsf{GL}_n(\mathrm{K})$ sowie $\varphi : G \to \mathsf{GL}_m(\mathrm{K})$ zwei Darstellungen von G. Der* ***Intertwining-Raum***[1] *zwischen ρ und φ ist*

$$\mathsf{Int}(\rho, \varphi) = \Big\{ M \in \mathrm{K}^{n\times m} \;\Big|\; \rho(g) \cdot M = M \cdot \varphi(g), \text{ für alle } g \in G \Big\}.$$

Der Intertwining-Raum ist ein K-Vektorraum; die ***Intertwining-Zahl*** *ist seine Dimension. In der für diese Arbeit relevanten Situation sei $G \leq \mathsf{GL}_n(\mathrm{K}) \times \mathsf{GL}_m(\mathrm{K})$ eine Symmetriegruppe im Sinne von Definition 1.1. Dann ist der Intertwining-Raum von G*

$$\mathsf{Int}(G) = \Big\{ M \in \mathrm{K}^{n\times m} \;\Big|\; L \cdot M = M \cdot R,\ \forall (L, R) \in G \Big\} = \mathsf{Int}(\Pi_L, \Pi_R),$$

wobei Π_L bzw. Π_R die Projektion von G auf die linke bzw. rechte Komponente bezeichnet.

Der Begriff des Intertwining-Raums ist ein gut untersuchtes Konzept in der Darstellungstheorie endlicher Gruppen. Der Intertwining-Raum wird zum Beispiel in den Büchern von Curtis und Reiner definiert ([24], §43.11 und [25], §10C). Curtis und Reiner betrachten jedoch statt des Vektorraums von Matrizen $\mathsf{Int}(\rho, \varphi)$ den KG-Modul $\mathrm{Hom}_{KG}(L_1, L_2)$. Der Intertwining-Raum wird auch in dem Buch von Clausen und Baum verwendet ([19], Ch. 2.2 und 7.2). In der von Clausen und Baum gewählten Konvention sind ρ und φ gegenüber Definition 1.6 vertauscht, was für die Theorie keinen Unterschied macht.

Mit der Definition von $\mathsf{Int}(G)$ wird eine zu $\mathsf{Int}(\rho, \varphi)$ äquivalente Sichtweise eingeführt: Statt zwei Darstellungen ρ und φ einer abstrakten Gruppe G zu betrachten, kann auch eine Gruppe G aus Paaren von Matrizen betrachtet werden. Die Projektionen Π_L und Π_R sind dann Darstellungen von G.

1.7 Beispiel Seien G und M wie in Beispiel 1.5. Dann ist $\mathsf{Int}(G)$ ein $\mathbb{C}$-Vektorraum der Dimension 5 mit Basis

$$\mathsf{Int}(G) = \left\langle \begin{bmatrix} 1&0&0&1\\0&0&0&0\\1&0&0&1\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&1&1&0\\0&0&0&0\\0&1&1&0\\0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0\\1&0&0&1\\0&0&0&0\\1&0&0&1 \end{bmatrix}, \begin{bmatrix} 0&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&1&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0\\0&0&1&0\\0&0&0&0\\0&1&0&0 \end{bmatrix} \right\rangle.$$

Der Koeffizientenvektor von M bezüglich dieser Basis ist $[1, 0, 1, 1, 0]$. ∎

[1] Eine freie deutsche Übersetzung wäre „der Raum der verflechtenden Matrizen“.

Eine elementare und universelle Methode zur Berechnung des Intertwining-Raums $\mathsf{Int}(\rho, \varphi)$ ist die folgende: Sei $g_1, \ldots, g_r$ ein Erzeugendensystem der Gruppe G. Dann ist $\mathsf{Int}(\rho, \varphi)$ genau der Lösungsraum des homogenen linearen Gleichungssystems

$$\begin{aligned} 0 &= \rho(g_1) \cdot M \Leftrightarrow M \cdot \varphi(g_1) \\ &\vdots \\ 0 &= \rho(g_r) \cdot M \Leftrightarrow M \cdot \varphi(g_r) \end{aligned}$$

in den Variablen $M_{11}, \ldots, M_{nm}$. Da die Dimension des Intertwining-Raums zwischen 0 und nm liegt, gibt es im allgemeinen kein Gleichungssystem mit weniger Variablen. Für den Spezialfall von induzierten Darstellungen, wie etwa transitiven Permutationsdarstellungen, kann die Dimension von $\mathsf{Int}(\rho, \varphi)$ effizienter mit Hilfe des Intertwining-Number-Theorems (Curtis/Reiner, [25], §10C, Th. 10.24) konstruiert werden. Im Fall von Permutationsdarstellungen kann zudem eine Basis von $\mathsf{Int}(\rho, \varphi)$ durch Symmetrisierung einer Basis von $\mathbb{Q}^{n \times m}$ konstruiert werden.

1.6 Algorithmengenerierung

Am Anfang der Einführung wurde das Prinzip der symmetriebasierten Algorithmengenerierung informell beschrieben. In diesem Abschnitt wird die Algorithmengenerierung nun formaler erläutert, basierend auf den Begriffen der vorangegangenen beiden Abschnitte. Abschließend werden zwei Beispiele ausführlich erläutert: ein Faltungssystem und eine Spektraltransformation. Doch zunächst zu dem in Abbildung 1.1 auf Seite 2 dargestellten Prinzip der symmetriebasierten Algorithmengenerierung.

Suche Gegeben ist eine Matrix $M \in \mathrm{K}^{n \times m}$. Sie ist nicht notwendigerweise quadratisch oder invertierbar. Die Multiplikation $x \mapsto xM$ soll mit einem schnellen Algorithmus realisiert werden. Dies geschieht mit Hilfe der Symmetrie. Dazu wird ein Symmetrietyp T gewählt, und es wird die Symmetriegruppe $G = \mathsf{Sym}_T(M)$ vom Typ T, im Sinne von Definition 1.1, bestimmt. Die Symmetriebestimmung ist das Thema dieser Arbeit.

Algogen Nach Satz 1.4 sind die beiden Projektionen Π_L und Π_R Darstellungen der Gruppe G vom Grad n und vom Grad m und $M \in \mathsf{Int}(\Pi_L, \Pi_R)$ nach Definition 1.6. Es ist günstig, für eine solche Situation eine intuitive Notation zu haben: Seien α und β zwei Darstellungen von G, so definieren wir

$$\alpha \xrightarrow{T} \beta \quad \Leftrightarrow \quad T \in \mathsf{Int}(\alpha, \beta),$$

also $\alpha(g) \cdot T = T \cdot \beta(g)$ für alle $g \in G$. Es gilt dann

$$\alpha \xrightarrow{T} \beta \text{ und } \beta \xrightarrow{U} \gamma \quad \Rightarrow \quad \alpha \xrightarrow{T \cdot U} \gamma.$$

Falls T invertierbar ist, gilt zudem

$$\alpha \xrightarrow{T} \beta \quad \Leftrightarrow \quad \beta \xrightarrow{T^{-1}} \alpha.$$

Nun zurück zu *Algogen*. Die Darstellung Π_L wird mit der *Algogen*-Prozedur ausreduziert, wie dies in der Dissertation von T. Minkwitz (1993), [64], beschrieben wird. Die reduzierende Matrix A_L wird als Produkt dünn besetzter Matrizen $A_L^{(1)}, \ldots, A_L^{(r)}$ konstruiert. Die ausreduzierte Darstellung ist ρ_L. Analog wird die Darstellung Π_R durch das Produkt dünn besetzter Matrizen $A_R = A_R^{(1)} \cdots A_R^{(s)}$ in die Darstellung ρ_R ausreduziert. Dies benötigt einen zweiten Aufruf der *Algogen*-Prozedur. Durch die Zerlegung von Π_L und Π_R in ausreduzierte Darstellungen ρ_L und ρ_R ist der „*Algogen*"-Schritt gemacht.

Spezialisierung Die *Algogen*-Prozedur hat die Freiheit, für ρ_L jede beliebige ausreduzierte Darstellung von Π_L zu wählen, gleiches gilt für ρ_R. Um nun die spezielle Matrix M zu realisieren, wird eine Matrix B so gewählt, daß das folgende Diagramm kommutiert:

$$\begin{array}{ccccc} \Pi_L & \overset{A_L^{(1)}}{\Longrightarrow} & \cdots & \overset{A_L^{(r)}}{\Longrightarrow} & \rho_L \\ M\downarrow & & /\!/\!/ & & \downarrow B \\ \Pi_R & \underset{A_R^{(1)}}{\Longrightarrow} & \cdots & \underset{A_R^{(s)}}{\Longrightarrow} & \rho_R \end{array}$$

Offensichtlich ist $B = A_L^{-1} \cdot M \cdot A_R$. Damit lautet der schnelle Algorithmus

$$M = A_L^{(1)} \cdot \cdots \cdot A_L^{(r)} \cdot B \cdot (A_R^{(s)})^{-1} \cdot \cdots \cdot (A_R^{(1)})^{-1}.$$

Eine übersichtliche Darstellung für eine solche Zerlegung ist der Signalflußgraph. Ein Beispiel ist in Abbildung 1.2 auf Seite 15 gezeigt.

Der Aufwand des „schnellen" Algorithmus wird zum Teil durch die Kosten für $x \mapsto xB$ bestimmt. Die Matrix B erfaßt den Anteil der zu realisierenden Matrix M, der *nicht* durch die Symmetrie G, bzw. Π_L und Π_R, beschrieben wird. Wie das kommutierende Diagramm oben zeigt, ist B ein Element von $\mathsf{Int}(\rho_L, \rho_R)$. Der Aufwand für B hängt also sehr eng mit der Struktur des K-Vektorraums $\mathsf{Int}(\rho_L, \rho_R)$ zusammen; er soll daher genauer beschrieben werden.

Angenommen die Charakteristik des Grundkörpers K teilt die Gruppenordnung $|G|$ nicht (Maschke-Bedingung), wie dies zum Beispiel für $\mathbb{Q} \leq \mathrm{K} \leq \mathbb{C}$ und beliebiges G der Fall ist. Dann ist die Gruppenalgebra $\mathrm{K}[G]$ halb-einfach und der Intertwining-Raum zweier Darstellungen von G besitzt eine besonders schöne Struktur. Um diese zu beschreiben, werde ein Repräsentantensystem $\rho_1, \ldots, \rho_h$

der irreduziblen Darstellungen von G gewählt. O.B.d.A. seien nun die Darstellungen ρ_L und ρ_R von der Form

$$\begin{aligned}\rho_L &= (\mathbf{1}_{n_1} \otimes \rho_1) \oplus \cdots \oplus (\mathbf{1}_{n_h} \otimes \rho_h) \quad \text{und} \\ \rho_R &= (\mathbf{1}_{m_1} \otimes \rho_1) \oplus \cdots \oplus (\mathbf{1}_{m_h} \otimes \rho_h),\end{aligned}$$

mit $n_k, m_k \geq 0$ für alle $k \in \{1..h\}$. Mit diesen Bezeichnungen folgt aus dem Lemma von Schur

$$\mathsf{Int}(\rho_L, \rho_R) = (\mathrm{K}^{n_1 \times m_1} \otimes \mathbf{1}_{\deg(\rho_1)}) \oplus \cdots \oplus (\mathrm{K}^{n_h \times m_h} \otimes \mathbf{1}_{\deg(\rho_h)}).$$

Die Dimension dieses K-Vektorraums ist offensichtlich $\sum_k n_k m_k$, was gerade das Skalarprodukt der Charaktere von Π_L und Π_R ist. Für die Algorithmengenerierung ist es also günstig, unterschiedliche irreduzible Komponenten in Π_L und Π_R zu haben. Dies ist nicht unbedingt bei der größten möglichen Symmetriegruppe der Fall. Ist M invertierbar, dann sind Π_L und Π_R ähnlich und es gilt $m_k = n_k$ für alle k. Ist M invertierbar und ist Π_L sogar eine reguläre Darstellung, dann ist $n_k = \deg(\rho_k)$ und daher $\dim \mathsf{Int}(\rho_L, \rho_R) = |G|$ aufgrund der Schur'schen Relationen. Nun zu den Beispielen.

1.8 Beispiel (Faltung auf der D_6) Es soll ein schneller Algorithmus für die Multiplikation mit folgender Matrix konstruiert werden:

$$M = \begin{bmatrix} a & d & b & f & c & e \\ b & f & a & d & e & c \\ e & c & f & b & d & a \\ f & b & e & c & a & d \\ d & a & c & e & b & f \\ c & e & d & a & f & b \end{bmatrix}.$$

Dabei sind a, b, c, d, e, f nicht notwendigerweise verschiedene komplexwertige Konstanten. Auf die Werte der Konstanten kommt es für diesen Typ der Algorithmengenerierung nicht an. Die Matrix M ist im wesentlichen eine Faltung mit dem Vektor $[a, b, c, d, e, f]$ in der Gruppenalgebra der Diedergruppe D_6. Die Matrix M ist daher eine Gruppenzirkulante, allerdings nur bezüglich zweier *bestimmter* Darstellungen der abstrakten Gruppe D_6.

Suche Die Matrix M besitzt genau sechs verschiedene Einträge und in jeder Zeile und jeder Spalte tritt jeder genau einmal auf. Dies deutet auf eine Perm-Perm-Symmetrie hin. Diese Form von Symmetrie wird in Kapitel 3 behandelt. Die Symmetriegruppe der Matrix M ist abstrakt gegeben durch

$$G = \mathsf{PermPerm}(M) = \mathsf{Sym}_{\mathsf{S}_6 \times \mathsf{S}_6}(M) \cong \mathsf{D}_6$$

und als Gruppe von Paaren von Permutationsmatrizen

$$G = \Big\langle \big((1\,2)(3\,4)(5\,6), (1\,3)(2\,4)(5\,6)\big), \big((1\,3)(2\,6)(4\,5), (1\,6)(2\,5)(3\,4)\big) \Big\rangle.$$

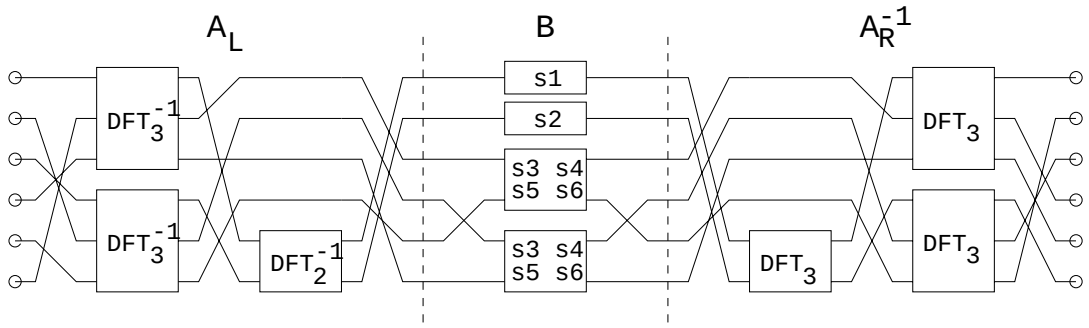


Abbildung 1.2: Signalflußgraph für $x \mapsto xM$ zur D_6.

Dieselbe Perm-Perm-Symmetriegruppe besitzt jede Matrix des Vektorraums

$$\mathsf{Int}(G) = \Big\langle \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \end{bmatrix}, \begin{bmatrix} \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \end{bmatrix}, \begin{bmatrix} \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}, \begin{bmatrix} \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \end{bmatrix}, \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}, \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \Big\rangle.$$

Die spezielle Matrix M besitzt den Koeffizientenvektor $[a, d, b, f, c, e]$ bezüglich der angegebenen Basis von $\mathsf{Int}(G)$. Da die Konstanten a bis f nicht angegeben wurden, kann die Matrix M auch als generisches Element von $\mathsf{Int}(G)$ aufgefaßt werden.

Algogen Die beiden Permutationsdarstellungen $\Pi_L = (L, R) \mapsto L$ und $\Pi_R = (L, R) \mapsto R$ der Gruppe G werden nun mit der *Algogen*-Prozedur ausreduziert. Da die Darstellungen ähnlich sind, können sie zur selben Darstellung ρ reduziert werden. Die reduzierenden Matrizen A_L und A_R sind

$$\begin{aligned} A_L &= (2\,5\,6)(3\,4) \cdot (\mathbf{1}_2 \otimes \mathrm{DFT}_3^{-1}) \cdot (1\,5\,2)(4\,6) \cdot (\mathbf{1}_4 \oplus \mathrm{DFT}_2^{-1}) \cdot (1\,3\,6\,2\,4\,5), \\ A_R &= (2\,6\,4)(3\,5) \cdot (\mathbf{1}_2 \otimes \mathrm{DFT}_3^{-1}) \cdot (1\,5\,2)(4\,6) \cdot (\mathbf{1}_4 \oplus \mathrm{DFT}_2^{-1}) \cdot (1\,3\,6\,2\,4\,5). \end{aligned}$$

Die einzelnen Faktoren der Produkte wurden mit einem MAGMA-Programm von T. Minkwitz (1997) konstruiert und mit den Methoden aus Kapitel 2 dieser Arbeit in die angegebene kompakte Form überführt. Die Funktionsweise der *Algogen*-Prozedur für Permutationsdarstellungen ist in [64] beschrieben. Sie soll an dieser Stelle nicht im Detail erklärt werden. Die Bilder der beiden oben angegebenen Erzeuger der Gruppe G unter der ausreduzierten Darstellung ρ sind

$$\rho(G) = \Big\langle \mathbf{1}_1 \oplus \Leftrightarrow\mathbf{1}_1 \oplus (\mathbf{1}_2 \otimes \begin{bmatrix} 0 & \zeta^2 \\ \zeta & 0 \end{bmatrix}),\ \mathbf{1}_1 \oplus \Leftrightarrow\mathbf{1}_1 \oplus (\mathbf{1}_2 \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}) \Big\rangle.$$

Dabei ist $\zeta = (\Leftrightarrow 1 + \mathrm{i}\sqrt{3})/2$ eine primitive dritte Einheitswurzel. Man erkennt die $(1 + 1 + 2 \times 2)$-Blockstruktur der irreduziblen Komponenten. Dies paßt zur Dimension des oben angebenen Intertwining-Raums, er hat die Dimension 6.

Spezialisierung Als dritten und letzten Schritt der Algorithmengenerierung wird die zerlegte Symmetrie spezialisiert auf die Matrix M. Dies geschieht durch

die Matrix $B = A_L^{-1} \cdot M \cdot A_R$, konkret

$$B = \begin{bmatrix} s_1 & & & & & \\ & s_2 & & & & \\ & & s_3 & 0 & s_4 & 0 \\ & & 0 & s_3 & 0 & s_4 \\ & & s_5 & 0 & s_6 & 0 \\ & & 0 & s_5 & 0 & s_6 \end{bmatrix} \quad \text{mit} \quad \begin{cases} s_1 &= a+b+f+c+d+e \\ s_2 &= a+b+f-c-d-e \\ s_3 &= a+f\zeta+b\zeta^2 \\ s_4 &= d+e\zeta+c\zeta^2 \\ s_5 &= d+c\zeta+e\zeta^2 \\ s_6 &= a+b\zeta+f\zeta^2. \end{cases}$$

Die Form von B zeigt deutlich die $(1+1+2\times 2)$-Struktur der irreduziblen Komponenten von ρ, denn es ist $\mathsf{Int}(\rho,\rho) = \mathbb{C} \oplus \mathbb{C} \oplus (\mathbb{C}^{2\times 2} \otimes \mathbf{1}_2)$. Mit der Berechnung von B ist das kommutative Diagramm komplett. Es lautet hier

$$\begin{array}{ccc} \Pi_L & \overset{A_L}{\longleftarrow} & \rho \\ M\big\downarrow & /\!/\!/ & \big\downarrow B \\ \Pi_R & \underset{A_R^{-1}}{\longleftarrow} & \rho. \end{array}$$

Der schnelle Algorithmus $M = A_L \cdot B \cdot A_R^{-1}$ ist dargestellt in Abbildung 1.2. ∎

1.9 Beispiel (Spektraltransformation der D_{12}) Es soll ein schneller Algorithmus für die Multiplikation mit folgender Matrix konstruiert werden:

$$M = \begin{bmatrix} 1 & 1 & 1 & \zeta & 1 & \zeta^2 \\ 1 & -1 & \zeta & 1 & -\zeta^2 & -1 \\ 1 & 1 & \zeta^2 & \zeta^2 & \zeta & \zeta \\ 1 & -1 & 1 & \zeta & -1 & -\zeta^2 \\ 1 & 1 & \zeta & 1 & \zeta^2 & 1 \\ 1 & -1 & \zeta^2 & \zeta^2 & -\zeta & -\zeta \end{bmatrix},$$

wobei $\zeta = (-1+\mathrm{i}\sqrt{3})/2$ eine primitive dritte Einheitswurzel ist. Die Matrix M ist eine Spektraltransformation zu einer *bestimmten* Darstellung der Diedergruppe D_{12} auf sechs Punkten. Diese wurde als Beispiel konstruiert, um die Methoden der Algorithmengenerierung zu erläutern.

Suche Die Matrix M besitzt die Eigenschaft, manche Permutationen simultan in Blöcke gleicher Blockstruktur zu zerlegen. Die Perm-Block-Symmetrie betrachtet systematisch alle derartigen Blockzerlegungen mit der Matrix M. Die Perm-Block-Symmetrie wird Kapitel 5 behandelt. Mit den dort beschriebenen Methoden kann der Verband der möglichen Blockstrukturen p und der zugehörigen maximalen Permutationsgruppen $\mathrm{G}_M(G)$ erstellt werden. Die Elemente des Verbands sind in der folgenden Tabelle zusammengetragen, der Übersichtlichkeit halber jeweils nur als Isomorphietyp. (Die Bezeichnungen für Gruppen sind in

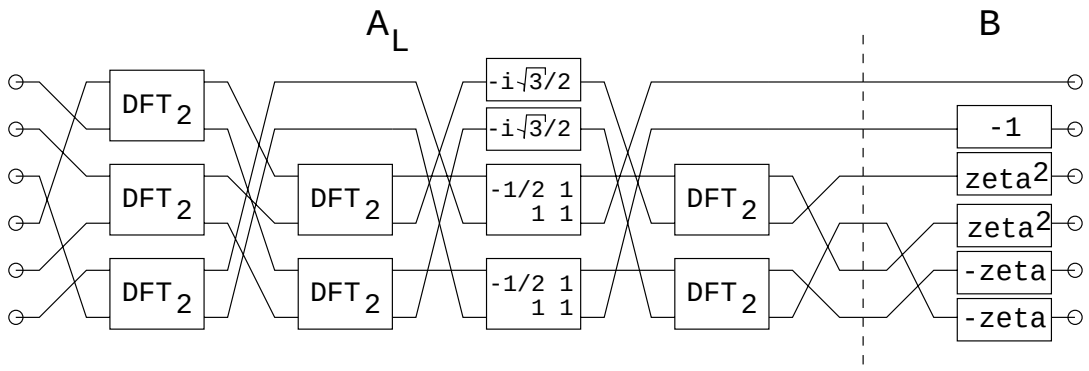


Abbildung 1.3: Signalflußgraph für $x \mapsto xM$ zur D_{12}.

Anhang A auf Seite 114 erläutert.)

p	$\mathrm{G}_M(p)$	$\lvert \mathrm{G}_M(p) \rvert$	$\dim \mathsf{Int}(\mathrm{G}_M(p))$
$(1\vert 2\,3\,4\,5\,6)$	S_6	120	2
$(1\vert 2\vert 3\,4\,5\,6)$	$\mathsf{S}_3 \wr \mathsf{Z}_2$	72	3
$(1\vert 2\,5\,6\vert 3\,4)$	$\mathsf{Z}_2 \times \mathsf{S}_4$	48	3
$(1\vert 2\vert 3\,6\vert 4\,5)$	$\mathsf{Z}_3 \times \mathsf{S}_3$	36	4
$(1\vert 2\,5\,6\vert 3\vert 4)$	$\mathsf{Z}_2 \times \mathsf{A}_4$	24	4
$(1\vert 2\vert 3\,4\vert 5\,6)$	D_{12}	12	4
$(1\vert 2\vert 3\vert 4\vert 5\vert 6)$	Z_6	6	6

Zur Algorithmengenerierung soll hier die Gruppe $G = \mathsf{D}_{12}$ verwendet werden. Die Gruppe G ist eine Untergruppe von $\mathsf{S}_6 \times \mathsf{GL}_6(\mathbb{C})$ gegeben durch

$$G = \langle \Big((1\,2\,3\,4\,5\,6), \begin{bmatrix} 1 & & & & & \\ & -1 & & & & \\ & & \zeta & 0 & & \\ & & 0 & \zeta^2 & & \\ & & & & -\zeta^2 & 0 \\ & & & & 0 & -\zeta \end{bmatrix} \Big), \Big((1\,4)(2\,3)(5\,6), \begin{bmatrix} 1 & & & & & \\ & -1 & & & & \\ & & 0 & \zeta & & \\ & & \zeta^2 & 0 & & \\ & & & & 0 & -\zeta^2 \\ & & & & -\zeta & 0 \end{bmatrix} \Big) \rangle.$$

Die Gruppe G ist eine Perm-Block-Symmetrie von M. Dieselbe Symmetriegruppe besitzt jede Matrix des Vektorraums

$$\mathsf{Int}(G) = \langle \begin{bmatrix} 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&1&0&0&0&0\\ 0&-1&0&0&0&0\\ 0&1&0&0&0&0\\ 0&-1&0&0&0&0\\ 0&1&0&0&0&0\\ 0&-1&0&0&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&1&\zeta&0&0\\ 0&0&\zeta&1&0&0\\ 0&0&\zeta^2&\zeta^2&0&0\\ 0&0&1&\zeta&0&0\\ 0&0&\zeta&1&0&0\\ 0&0&\zeta^2&\zeta^2&0&0 \end{bmatrix}, \begin{bmatrix} 0&0&0&0&1&\zeta^2\\ 0&0&0&0&-\zeta^2&-1\\ 0&0&0&0&\zeta&\zeta\\ 0&0&0&0&-1&-\zeta^2\\ 0&0&0&0&\zeta^2&1\\ 0&0&0&0&-\zeta&-\zeta \end{bmatrix} \rangle.$$

Der Koeffizientenvektor der speziellen Matrix M bezüglich der angegebenen Basis von $\mathsf{Int}(G)$ ist $[1,1,1,1]$. Die Erzeuger von G zeigen die $(1+1+2+2)$-Struktur der irreduziblen Komponenten. (Dies ist *nicht* identisch zu $(1+1+2\times 2)$, denn die beiden 2-dim. Darstellungen sind inäquivalent.)

Algogen Die beiden Projektionen $\Pi_L = (L,R) \mapsto L$ und $\Pi_R = (L,R) \mapsto R$ sind ähnliche Darstellungen der Gruppe G. Dabei ist Π_L eine Permutationsdarstellung und Π_R ist eine direkte Summe Irreduzibler. Mit der *Algogen*-Prozedur wird nun die Permutationsdarstellung ausreduziert zur Darstellung ρ_L. Die reduzierende Matrix A_L wird als Produkt dünn besetzter Matrizen konstruiert. Die folgenden Zerlegungsmatrizen wurde mit einem MATHEMATICA/GAP-Programm von T. Minkwitz (1993) numerisch konstruiert und mit den Methoden aus Kapitel 2 dieser Arbeit in die kompakte Schreibweise gebracht.

$$\begin{aligned} A_L \;=\; & (1\,2\,3\,6\,5\,4)\cdot(\mathbf{1}_3 \otimes \mathrm{DFT}_2)\cdot(1\,3\,4\,6\,2\,5)\cdot\Big(\mathbf{1}_2 \oplus (\mathbf{1}_2 \otimes \mathrm{DFT}_2)\Big)\cdot \\ & \cdot(1\,4)(2\,6)\cdot\Big(\Leftrightarrow\frac{\mathrm{i}\sqrt{3}}{2}\mathbf{1}_2 \oplus (\mathbf{1}_2 \otimes \begin{bmatrix} -1/2 & 1 \\ 1 & 1 \end{bmatrix})\Big)\cdot(1\,4)(2\,6)\cdot \\ & \cdot\Big(\mathbf{1}_2 \oplus (\mathbf{1}_2 \otimes \mathrm{DFT}_2)\Big)\cdot(3\,5\,6\,4). \end{aligned}$$

Die Darstellung Π_R ist bei diesem Symmetrietyp bereits ausreduziert. Daher kann einfach $A_R = \mathbf{1}_6$ gewählt werden.

Spezialisierung Die reduzierte Darstellung ρ_L ist nicht identisch mit Π_R, denn die *Algogen*-Prozedur hatte die Freiheit, eine beliebige reduzierte Darstellung zu konstruieren. Durch Wahl der Matrix

$$B = A_L^{-1}\cdot M = (4\,6\,5)\cdot\mathrm{diag}(1,\Leftrightarrow\!1,\zeta^2,\zeta^2,\Leftrightarrow\!\zeta,\Leftrightarrow\!\zeta).$$

wird dies korrigiert. Das folgende Diagramm zeigt die auftretenden Darstellungen und ihre Relationen.

$$\begin{array}{ccc} \Pi_L & \overset{A_L}{\Longleftrightarrow} & \rho_L \\ {\scriptstyle M}\big\downarrow & /\!/\!/ & \big\downarrow{\scriptstyle B} \\ \Pi_R & \underset{\mathbf{1}_6}{\Longleftrightarrow} & \Pi_R \end{array}$$

Der schnelle Algorithmus ist $M = A_L \cdot B$. Ein Signalflußdiagramm zu dieser Zerlegung zeigt Abbildung 1.3. ∎

1.7 Übersicht der Kapitel dieser Arbeit

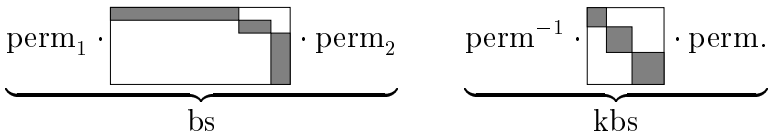
Um den Zugang zu den einzelnen Aspekten dieser Arbeit zu erleichtern, folgt eine Kurzbeschreibung der einzelnen Kapitel. Die Arbeit setzt sich aus zwei Teilen zusammen: Methoden zur Symmetriesuche und Anwendungsbeispiele.

Kapitel 1 (dieses Kapitel) stellt eine Einführung in die vorliegende Arbeit dar. Die Aufgabenstellung wird abgegrenzt und es wird die Methode der symmetriebasierten Algorithmengenerierung erläutert und eingeordnet in den historischen Zusammenhang. Danach wird ein allgemeiner Begriffsrahmen für die Symmetrie linearer Transformationen geschaffen. Mit den so definierten formalen Strukturen

wird die symmetriebasierte Algorithmengenerierung algebraisch beschrieben und an zwei Beispielen ausführlich demonstriert.

Kapitel 2 behandelt das Problem, wie eine dünn besetzte Matrix als Term in den Operationen Matrixmultiplikation, direkte Summe und Kroneckerprodukt geschrieben werden kann. Diese Strukturerkennung von dünn besetzten Matrizen ist ein Hilfsverfahren, um schnelle Algorithmen linearer Transformationen als kompakte Terme zu repräsentieren. Die Problemkomplexität der Strukturerkennung ist mindestens so hoch wie die des bekannten Problems GRAPH-ISOMORPHISMUS. Trotz dieser Tatsache kann das Problem für die praktisch relevanten Größen immer sehr schnell gelöst werden.

In Kapitel 2 wird der Begriff der Blockstruktur einer rechteckigen Matrix definiert. Die Blockstruktur (bs) ist nicht zu verwechseln mit der Spaltenblockstruktur (cbs) aus Kapitel 6 oder mit der konjugierten Blockstruktur (kbs) aus Kapitel 5. Die Blockstruktur ist ein quantitatives Maß für die feinste Darstellung einer rechteckigen Matrix A als direkte Summe von kleineren rechteckigen Matrizen. Dabei dürfen die Zeilen und Spalten zudem unabhängig voneinander permutiert werden. Die Spaltenblockstruktur mißt nur die Zerlegung der Spaltenindexmenge. Dem gegenüber ist die konjugierte Blockstruktur nur definiert für quadratische und invertierbare Matrizen A. Die konjugierte Blockstruktur ist ein Maß für die Zerlegung von A in eine direkte Summe *quadratischer* Teilmatrizen. Dabei dürfen Zeilen und Spalten nur *simultan* permutiert werden. Durch diese Einschränkung wird die Blockstruktur verträglich mit der Matrixmultiplikation.



Kapitel 3 behandelt die Perm-Perm-Symmetrie. Wie Leon (1991) gezeigt hat, kann die Perm-Perm-Symmetrie effizient durch partitionsbasierte Backtrack-Suche bestimmt werden. Die Bestimmung der Perm-Perm-Symmetrie ist mindestens so schwierig, wie die Lösung von GRAPH-ISOMORPHISMUS. Es existieren jedoch stark einschränkende, notwendige Bedingungen, die die exponentielle Suche in der Praxis lösbar machen. In Kapitel 3 werden die Methoden kurz erläutert, und es werden einige Gebiete der Informatik und Mathematik aufgezeigt, in denen die Perm-Perm-Symmetrie eine wichtige Rolle spielt.

Kapitel 4 behandelt die Perm-Mat-Symmetrie. Bei diesem Symmetrietyp besitzt die Matrix mehr Zeilen als Spalten. Es wird eine Permutationsdarstellung auf den Zeilen in eine beliebige Darstellung auf den Spalten verwandelt. Die Bestimmung der Perm-Mat-Symmetrie ist ein Hilfsalgorithmus für die Berechnung der Perm-Irred-Symmetrie.

Kapitel 5 behandelt die Perm-Irred-Symmetrie und als nützliche Verallgemeinerung die Perm-Block-Symmetrie. Bei diesem Symmetrietyp wird eine Permutationsdarstellung durch Konjugation mit der gegebenen Transformation in

Blöcke zerlegt. Die Schwierigkeit besteht darin, eine Übersicht der möglichen Zerlegungen zu erhalten. Das theoretische Haupthilfsmittel ist die konjugierte Blockstruktur (kbs) einer Matrix. Die Perm-Block-Symmetrie wird formal definiert als die Menge der maximalem Permutationsgruppen zu allen möglichen Blockzerlegungen. Die Perm-Block-Symmetrie besitzt eine Verbandstruktur und kann mit dem Verband der Permutationsgruppen und dem Verband der Partitionen in Verbindung gebracht werden.

Zur Bestimmung der Perm-Block-Symmetrie wird zunächst ein naheliegender Algorithmus exponentieller Laufzeit angegeben. Unter Verwendung der Perm-Mat-Symmetrie wird ein zweiter Suchalgorithmus angegeben, der in der Praxis nützlicher ist, obwohl er ebenfalls exponentielle Laufzeit besitzt: Wird eine Schranke für die Größe der auftretenden Blöcke angenommen, so ist die Laufzeit des zweiten Algorithmus polynomial in der Größe der Matrix. Die Problemkomplexität der Perm-Block-Symmetriesuche ist unbekannt.

Kapitel 6 behandelt das Ausdünnen einer rechteckigen Matrix. Dabei wird eine gegebene Matrix durch invertierbare Zeilenoperationen so transformiert, daß maximal viele Nulleinträge entstehen. Dieses Verfahren hat im Zusammenhang mit der symmetriebasierten Algorithmengenerierung die Aufgabe, ein Anwendungsproblem für die Symmetriesuche vorzubereiten.

Es wird zunächst gezeigt, daß das Ausdünnungsproblem Matroid-Struktur besitzt und daher durch einen Greedy-Algorithmus gelöst werden kann. Der Austausch-Schritt des Greedy-Algorithmus ist aufwendig, kann jedoch durch ein Suchverfahren bewerkstelligt werden. Das vorgeschlagene Suchverfahrens besitzt im allgemeinen exponentielle Laufzeit. Besitzt die auszudünnende Matrix nach Basiswechsel eine Blockstruktur, so kann das Problem mit einem Teile-und-Herrsche Ansatz zerkleinert werden. Die Verwendung der Blockstruktur geht im wesentlichen auf T. Minkwitz (1994) zurück. Schließlich wird gezeigt, daß die Problemkomplexität des Ausdünnens mit der des bekannten Problems MINIMALGEWICHT EINES LINEAREN BLOCKCODES übereinstimmt.

Kapitel 7 ist den klassischen Methoden der schnellen Fourier-Transformation auf der regulären Darstellung einer zyklischen Gruppe gewidmet. Im Lichte der automatischen Symmetriebestimmung können die Zerlegungsmethoden von Good-Thomas, Cooley-Tukey und Rader in neuer Form präsentiert werden. Es werden alle auftretenden Permutationen und Twiddle-Faktoren explizit angegeben. Schließlich wird eine vom Autor in der Programmiersprache C erstellte konkrete Implementierung der FFT vorgestellt.

Kapitel 8 zeigt, wie die für diese Arbeit entwickelten Methoden zur Symmetriebestimmung in der statistischen Physik genutzt werden können. Dazu werden klassische Spin-Gitter-Systeme mit dem Transfermatrix-Formalismus untersucht. Die Kenntnis der Symmetrie der Transfermatrix hilft bei der numerischen Untersuchung des Systems. Mit den in dieser Arbeit vorgestellten Verfahren wird die Symmetrie kleiner 2d-Ising-Modelle und kleiner 2d-Potts-Modelle bestimmt.

Diese gefundene Symmetrie wird mit der physikalisch bekannten Symmetrie verglichen. Es ergibt sich, daß die betrachteten Modelle keine weiteren Symmetrien besitzen außer den bekannten Invarianzen unter Raumsymmetrie, Farbsymmetrie und der Ferro/Antiferro-Symmetrie.

Anhang A enthält die durchgängig verwendeten Notationen für Körper, Verbände, abstrakte Gruppen, Permutationsgruppen, Matrizen und Darstellungen von Gruppen. Das Literaturverzeichnis, Stichwortverzeichnis und eine Kurzzusammenfassung in deutscher und englischer Sprache schließen die Arbeit ab.

Teil I

Werkzeuge zur Strukturbestimmung

2

Struktur dünn besetzter Matrizen

In diesem Kapitel geht es darum, eine dünn besetzte Matrix als Term in gut bekannten Matrizenoperationen, wie der Matrixmultiplikation, der direkten Summe ($\oplus$), oder dem Kroneckerprodukt ($\otimes$), zu schreiben. (Zur Definition der Operationen siehe Anhang A auf Seite 116.) Die betrachtete Matrix könnte beispielsweise als Faktor in der Zerlegung einer Transformation auftreten. In dieser Situation ist es wichtig, den Faktor strukturell zu verstehen. Ein einführendes Beispiel möge klarstellen, daß es um eine Art von Strukturkompression geht.

2.1 Beispiel Gegeben sei die dünn besetzte (13×16)-Matrix ($\cdot = 0$)

$$A = \begin{bmatrix}
5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 6 & \cdot & \cdot & \cdot & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 8 \\
\cdot & \cdot & 11 & \cdot & \cdot & \cdot & \cdot & \cdot & 9 & \cdot & 10 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 7 & 8 & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 5 & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 9 & \cdot & \cdot & \cdot & 10 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 11 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & \cdot & \cdot & \cdot \\
3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 5 & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 8 & \cdot & \cdot & \cdot & \cdot & \cdot & 6 & \cdot & 7 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 3 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 9 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 10 & 11 & \cdot & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix}.$$

Man bestimme eine kompakte Term-Darstellung von A in $\cdot$, $\oplus$ und $\otimes$.
Eine Lösung ist

$$A = S \cdot \left(\mathbf{1}_1 \oplus \left(\mathbf{1}_3 \otimes \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \right) \oplus \left(\mathbf{1}_3 \otimes \begin{bmatrix} 6 & 7 & 8 \\ 9 & 10 & 11 \end{bmatrix} \right) \right) \cdot T,$$

mit den beiden Permutationen

$$\begin{aligned} S &= (1,13,6,8,4,7)(2,11,3,9,12) \quad \text{und} \\ T &= (1,7,5,3,13,4,14,9,11,12,6,2,8,15,10). \end{aligned}$$

Dabei bezeichnet $\mathbf{1}_n$ eine $(n \times n)$-Einheitsmatrix, und eine Permutation S steht für ihre Permutationsmatrix $[\delta_{i^S,j} \mid i, j]$, wie es in Anhang A auf Seite 116 erklärt ist. ∎

Für dieses Kapitel bezeichne K immer die Grundmenge der Matrixeinträge. Als einzige Operation auf K wird der Vergleich $(x = y)$ benötigt; er sei entscheidbar. Das Kroneckerprodukt mit einer Einheitsmatrix und die direkte Summe sind strukturelle Konstruktionen auf Matrizen. Beide erfordern keinerlei Arithmetik in K. Die zentrale Aufgabe in diesem Kapitel ist das folgende

2.2 Problem *Gegeben ist eine Matrix $A \in \mathrm{K}^{n \times m}$. Finde Permutationen $S \in \mathsf{S}_n$ und $T \in \mathsf{S}_m$, so daß*

$$A = S \cdot \left((\mathbf{1}_{r_1} \otimes A^{(1)}) \oplus \cdots \oplus (\mathbf{1}_{r_s} \otimes A^{(s)}) \right) \cdot T$$

mit geeigneten Matrizen $A^{(k)} \in \mathrm{K}^{n_k \times m_k}$ und natürlichen Zahlen $r_k \geq 1$ für $k \in \{1..s\}$ und $s \geq 1$. Die Zerlegung sei feinst möglich; das heißt $r_k = \max$.

Bevor das Problem näher betrachtet wird, einige Bemerkungen zu verwandten Arbeiten: Ein systematischer Zugang zu Problem 2.2 ist dem Autor nicht bekannt. Wie in diesem Kapitel gezeigt wird, führt das Problem aber direkt auf das bekannte Problem GRAPH-ISOMORPHISMUS. Die Komplexität dieses Problems wird in dem Buch [35] von Garey und Johnson (1979) behandelt. Algorithmisch wird das Problem GRAPH-ISOMORPHISMUS unter anderem von J. Leon (1991) behandelt, [55].

Der Rest des Kapitels ist wie folgt aufgebaut: In Abschnitt 2.1 wird der Begriff der Blockstruktur formal definiert, und es werden die relevanten Eigenschaften formuliert und bewiesen. Ein quantitatives Maß für die Blockstruktur bildet die Basis der Matrixzerlegung in diesem Kapitel. In Abschnitt 2.2 steht die Algorithmik wieder im Vordergrund; es wird ein Verfahren vorgestellt, das die Matrix auf Blockdiagonalform transformiert. Der Aufwand des Verfahrens wird untersucht. Damit ist der zügig berechenbare Teil beendet. Anschließend wird in Abschnitt 2.3 untersucht, wie permutierte Kopien als identisch entlarvt werden können. Dies führt zum Graphen-Isomorphismus-Problem, denn die Blöcke können Adjazenzmatrizen von Graphen sein. Das Problem GRAPH-ISOMORPHISMUS ist ein Standardproblem der Computeralgebra; es werden zwei Implementierungen zu dessen Lösung zitiert. Darauf aufbauend wird ein Algorithmus zur Lösung des gesamten Problems 2.2 angegeben. Der Abschnitt 2.4 beendet das Kapitel mit einer Zusammenfassung.

2.1 Blockstruktur

Die erste wesentliche Schritt zur Lösung von Problem 2.2 besteht darin, die in der Matrix enthaltenen Blöcke zu erkennen und zu trennen. Dazu ist ein quantitives

Maß für die vorliegende Blockstruktur notwendig. In diesem Abschnitt geht es daher um die formale Definition der Blockstruktur und ihre Eigenschaften.

2.3 Definition *Die **Blockstruktur** (engl. block structure) der Matrix $A \in \mathrm{K}^{n\times m}$ ist die Partition*

$$\mathrm{bs}(A) = (\{1..n\} \times \{1..m\})/\sim^+,$$

wobei $\sim^+$ der transitive Abschluß der symmetrischen Relation $\sim$ ist, welche folgendermaßen definiert ist

$$(i_1, j_1) \sim (i_2, j_2) \Leftrightarrow (i_1 = i_2 \text{ oder } j_1 = j_2) \text{ und } A_{i_1,j_1} \neq 0 \text{ und } A_{i_2,j_2} \neq 0.$$

Man beachte, daß die Relation $\sim^+$ nicht notwendigerweise reflexiv ist; daher ist sie im allgemeinen keine Äquivalenzrelation. Die Faktorstruktur $\mathrm{bs}(A)$ ist trotzdem wohldefiniert; sie enthält die nicht leeren Blöcke der Matrix. Diese überdecken allerdings im allgemeinen nicht $\{1..n\}\times\{1..m\}$, wie das triviale Beispiel $\mathrm{bs}(\mathbf{0}) = \emptyset$ zeigt. Für die Matrix A aus Beispiel 2.1 ist die Blockstruktur gegeben durch

$$\begin{aligned}\mathrm{bs}(A) \;=\; & \{\{1,8\}\times\{1,12\},\{2,6\}\times\{4,8,16\},\{3,10\}\times\{3,9,11\},\\ & \{4,12\}\times\{2,14,15\},\{5,7\}\times\{7,13\},\{9,11\}\times\{5,6\},\{13\}\times\{10\}\}.\end{aligned}$$

Zum Vergleich von bs, cbs (aus Kapitel 6) und kbs (aus Kapitel 5) sei auf Seite 19 der Einleitung verwiesen. Die ersten beiden Lemmata klären, wie sich die Blockstruktur unter Permutationen der Zeilen oder der Spalten transformiert und wie sie sich aus direkten Summen zusammensetzt.

2.4 Lemma Sei $A \in \mathrm{K}^{n\times m}$ sowie $S \in \mathsf{S}_n$ sowie $T \in \mathsf{S}_m$. Dann gilt

$$\mathrm{bs}(SAT) = \Big\{\{(i^{S^{-1}}, j^T) \mid i \in I, j \in J\} \;\Big|\; I \times J \in \mathrm{bs}(A)\Big\}.$$

Beweis Aus der Definition von $\mathrm{bs}(A)$ folgt, daß jeder Block in $\mathrm{bs}(A)$ von der Form $I \times J$ mit $\emptyset \neq I \subseteq \{1..n\}$ und $\emptyset \neq J \subseteq \{1..m\}$ ist. In der gewählten Konvention für Permutationsmatrizen gilt zudem

$$(SAT)_{i,j} = A_{i^S, j^{T^{-1}}}.$$

Sei $\sim$ die Relation für A gemäß Definition 2.3 und $\sim'$ sei die entsprechende Relation für SAT. Dann gilt

$$\begin{aligned}(i_1, j_1) \sim' (i_2, j_2) \;&\Leftrightarrow\; (SAT)_{i_1,j_1} \neq 0, (SAT)_{i_2,j_2} \neq 0, (i_1 = i_2 \vee j_1 = j_2)\\ &\Leftrightarrow\; A_{i_1^S, j_1^{T^{-1}}} \neq 0, A_{i_2^S, j_2^{T^{-1}}} \neq 0, (i_1^S = i_2^S \vee j_1^{T^{-1}} = j_2^{T^{-1}})\\ &\Leftrightarrow\; (i_1^S, j_1^{T^{-1}}) \sim (i_2^S, j_2^{T^{-1}}).\end{aligned}$$

Die Permutationen S und T bezeichnen daher lediglich die Punkte um. ∎

2.5 Lemma Sei $A \in \mathrm{K}^{n\times m}$ und $B \in \mathrm{K}^{r\times s}$. Dann gilt

$$\mathrm{bs}(A \oplus B) = \mathrm{bs}(A) \,\dot\cup\, \Big\{ \{(n+i, m+j) \mid i \in I, j \in J\} \;\Big|\; I \times J \in \mathrm{bs}(B) \Big\}.$$

Beweis Alle ($\neq 0$)-Einträge von $A \oplus B$, falls es welche gibt, liegen in den beiden Blöcken $\{1..n\} \times \{1..m\}$ und $\{n+1..n+r\} \times \{m+1..m+s\}$. Diese Blöcke werden beim Übergang zum transitiven Abschluß $\sim^+$ nicht wieder vereinigt. Daher liefert $\mathrm{bs}(A)$ die Struktur auf dem ersten der beiden großen Blöcke und $(n, m) + \mathrm{bs}(B)$ liefert die Struktur auf dem anderen. ∎

Aus den beiden vorigen Aussagen ergibt sich sofort eine Methode, um eine dünn besetzte Matrix zu sortieren und auf Blockdiagonalform zu transformieren. Dies ist bereits die Hälfte von Problem 2.2.

2.6 Algorithmus (Blöcke sortieren) Gegeben ist eine Matrix $A \in \mathrm{K}^{n\times m}$. Es werden Permutation $S \in \mathsf{S}_n$ und $T \in \mathsf{S}_m$ konstruiert, so daß

$$A = S \cdot (A^{(1)} \oplus \cdots \oplus A^{(r)}) \cdot T$$

eine feinste Blockzerlegung von A ist. Die $A^{(k)}$ sind kleinere, rechteckige Matrizen. Algorithmus: Bestimme $\mathrm{bs}(A)$. Das ist die Struktur der feinsten Zerlegung. Ordne die Blöcke in $\mathrm{bs}(A)$ fest an. Ordne die Punkte in den Blöcke in $\mathrm{bs}(A)$ fest an; etwa

$$\mathrm{bs}(A) = \Big\{ \{i_{11}, i_{12}, \dots\} \times \{j_{11}, j_{12}, \dots\}, \{i_{21}, i_{22}, \dots\} \times \{j_{21}, j_{22}, \dots\}, \dots \Big\}.$$

Wähle S^{-1} so, daß die Folge der Zeilenindizes $i_{11}, i_{12}, \dots$ auf $1, 2, \dots$ abgebildet wird. Analog wähle T so, daß die Folge der Spaltenindizes $j_{11}, j_{12}, \dots$ auf $1, 2, \dots$ abgebildet wird. Wenn nicht alle Zeilen- oder nicht alle Spaltenindizes in $\mathrm{bs}(A)$ vorkommen, dann werden die fehlenden Zahlen an die Folge angefügt. ∎

Wird Algorithmus 2.6 auf die Matrix aus Beispiel 2.1 angewendet, so ergibt sich die folgende Darstellung von A. Die Reihenfolge der Blöcke und der Punkte in $\mathrm{bs}(A)$ ist dabei so gewählt, wie es nach Definition 2.3 angegeben wurde.

$$A = S \cdot \begin{bmatrix}
5 & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
3 & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 6 & 7 & 8 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 9 & 10 & 11 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 11 & 9 & 10 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 8 & 6 & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 6 & 7 & 8 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 9 & 10 & 11 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 5 & 4 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & 2 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 5 & 4 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & 2 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix} \cdot T$$

mit den beiden Permutationen

$$\begin{aligned} S &= (2,8,12,11,9,5,3)(4,6,10,7) \quad \text{und} \\ T &= (2,9,7,12)(3,6,15,11,8,4)(5,14,10,16). \end{aligned}$$

2.2 Finden der Blockstruktur

In Algorithmus 2.6 wird die Blockstruktur $\mathrm{bs}(A)$ explizit verwendet. In diesem Abschnitt wird ein Algorithmus zu ihrer Berechnung angegeben.

2.7 Algorithmus Zu $A \in \mathrm{K}^{n\times m}$ wird $\mathrm{bs}(A)$ berechnet. Der Algorithmus pflegt eine Blockstruktur bs, die mit jedem Eintrag A_{ij} aktualisiert wird.

```
bs := ∅;
for i ∈ {1..n}, j ∈ {1..m} do
  if A_ij ≠ 0 then
    I_1 := {i};
    J_1 := {j};
    bs_1 := ∅;
    for I × J ∈ bs do
      if i ∈ I or j ∈ J then
        I_1 := I_1 ∪ I;
        J_1 := J_1 ∪ J
      else
        bs_1 := bs_1 ∪ {I × J}
      fi
    od;
    bs := bs_1 ∪ {I_1 × J_1}
  fi
od.
```

(Das kartesische Produkt $\times$ ist hier als symbolische Konstruktion zu verstehen; es wird nicht ausgewertet.) Am Ende ist $bs = \mathrm{bs}(A)$. ∎

2.8 Satz *Algorithmus 2.7 ist korrekt. Er benötigt die minimale Anzahl von Zugriffen auf Komponenten von A; das sind $n\cdot m$. Die Anzahl der elementaren Operationen mit ganzen Zahlen ist $O((1+\eta\min(n,m)^2)\cdot nm)$, wobei $0 \le \eta \le 1$ der Anteil von $(\neq 0)$-Einträge in A ist.*

Beweis Die Korrektheit des Algorithmus hängt an der folgenden Schleifeninvariante der äußeren Schleife: $bs = \mathrm{bs}(A^{(i,j)})$ mit der Matrix $A^{(i,j)}$, die definiert ist durch

$$A^{(i,j)}_{kl} = \begin{cases} A_{kl} & \text{falls } k < i \text{ oder } k = i \text{ und } l \le j, \\ 0 & \text{sonst.} \end{cases}$$

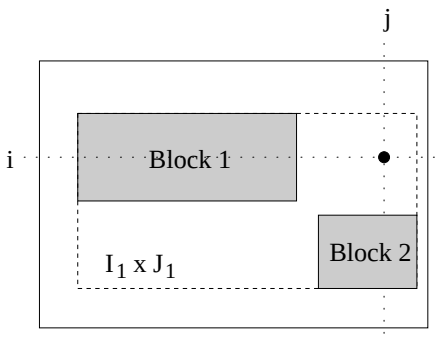


Abbildung 2.1: $A_{ij} \neq 0$ verbindet Block 1 und Block 2

Die Matrix $A^{(i,j)}$ entsteht also aus A, indem alle Komponenten, die von der `for`-Schleife nach (i,j) aufgezählt werden, durch 0 ersetzt werden. Insbesondere ist $A^{(n,m)} = A$, woraus die Korrektheit des Algorithmus folgt.

Die Gültigkeit der Schleifeninvariante wird durch Induktion über (i,j) gezeigt: Am Anfang ist $bs = \mathrm{bs}(\mathbf{0}) = \emptyset$, was korrekt ist. Sei nun (i',j') das unmittelbar vor (i,j) aufgezählte Paar und $bs' = \mathrm{bs}(A^{(i',j')})$ nach Induktionsvoraussetzung. Ist $A_{ij} = 0$, dann setzt das Programm $bs := bs'$, was korrekt ist, da $A^{(i,j)} = A^{(i',j')}$.

Im anderen Fall ist $A_{ij} \neq 0$. Dann berechnet die innere Schleife ein neues bs aus bs'. Dazu wird bs' partitioniert in $bs' = bs'_1 \dot\cup (bs' \setminus bs'_1)$ mit

$$bs'_1 = \{I \times J \in bs' \mid i \notin I \text{ und } j \notin J\}.$$

Damit ist

$$bs = bs'_1 \cup \Big\{ \Big(\{i\} \cup \bigcup_{I\times J \in bs'\setminus bs'_1} I \Big) \times \Big(\{j\} \cup \bigcup_{I\times J \in bs'\setminus bs'_1} J \Big) \Big\}.$$

Es bleibt $bs = \mathrm{bs}(A^{(i,j)})$ zu zeigen: Die I-Komponenten der Blöcke sind paarweise disjunkt, die J-Komponenten ebenso. Daher verbindet (i,j) im transitiven Abschluß genau die Blöcke $I \times J \in bs'$, bei denen $i \in I$ oder $j \in J$ ist. Siehe dazu Abbildung 2.1. Das sind genau die Blöcke in $bs' \setminus bs'_1$. Der neu entstandene Block $I_1 \times J_1$ enthält aber auch (i,j); dies muß noch explizit hinzugefügt werden. Alle anderen Blöcke von $A^{(i',j')}$ werden einfach übernommen, denn sie werden durch A_{ij} nicht verbunden. Damit ist gezeigt, daß die Schleifeninvariante tatsächlich schleifen-invariant ist.

Daß der Algorithmus die minimale Anzahl von Zugriffen auf Komponenten von A benötigt, ist trivial. Die Schleifeninvariante zeigt zudem $|bs| \leq \min(n,m)$, denn mehr Blöcke besitzt keine $(n \times m)$-Matrix. Daher wird die innere Schleife höchstens $\min(n,m)$ mal durchlaufen. Die Operation $J_1 \cup J$ benötigt $O(\max(|J_1|,|J|))$ elementare Operationen (Merging sortierter Listen). Damit ergibt sich die genannte Laufzeitabschätzung. ∎

2.3 Erkennen ähnlicher Blöcke

In den letzten Abschnitten wurde gezeigt, wie die Blöcke der Matrix A getrennt werden können. Das am Anfang des Kapitel formulierte Problem 2.2 ist aber noch umfangreicher: Blöcke, die einander unter Zeilen- und Spaltenvertauschungen ähnlich sind, sollen *identisch* gemacht werden, damit sie mit der Formel

$$\underbrace{A \oplus \cdots \oplus A}_{r \text{ viele}} \stackrel{\rightarrow}{=} \mathbf{1}_r \otimes A$$

zum Kroneckerprodukt zusammengefaßt werden können. Es bleibt also im wesentlichen das folgende

2.9 Problem *Gegeben seien zwei Matrizen $A, B \in \mathrm{K}^{n\times m}$. Bestimme Permutationen $S \in \mathsf{S}_n$, $T \in \mathsf{S}_m$ mit $SAT = B$, oder zeige, daß es kein solches Paar (S,T) gibt.*

Die schlechte Nachricht zuerst:

2.10 Satz *Das zu Problem 2.9 gehörende Entscheidungsproblem („Gibt es (S,T)?“) ist nicht einfacher als das Problem* Graph-Isomorphismus.

Das Problem Graph-Isomorphismus läuft in dem Buch [35] von M. R. Garey und D. S. Johnson unter der Bezeichung [Open1] in Rubrik A13. Es lautet: Gegeben zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$. Sind G_1 und G_2 isomorph? (Gibt es eine Bijektion $f : V_1 \to V_2$ mit $\{u,v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$?) Das Problem ist schwierig; es wird vermutet, daß es in NP $\setminus$ (P$\cup$NP-vollst.) liegt. Nun zum Beweis von Satz 2.10.

Beweis Sei $G = (V,E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_m\}$ und $e_j \in V \times V$ für alle j. Die Inzidenzmatrix $I(G)$ von G ist die 0/1-Matrix der Größe $n \times m$, die definiert ist durch

$$I(G)_{ij} = 1 \Leftrightarrow v_i \in e_j.$$

Nach dieser Vorbemerkung jetzt zur Reduktion von Graph-Isomorphismus auf Problem 2.9. Seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ zwei Graphen mit $n = |V_1| = |V_2|$ und $m = |E_1| = |E_2|$. Dann sind G_1 und G_2 genau dann isomorph, wenn es Permutationen $S \in \mathsf{S}_n$, $T \in \mathsf{S}_m$ gibt mit

$$S \cdot I(G_1) \cdot T = I(G_2).$$

Mit einem polynomialen Algorithmus für Problem 2.9 kann also auch das Problem Graph-Isomorphismus in polynomialer Zeit gelöst werden. ∎

Das Problem 2.9 der ähnlichen Matrizen ist ein bekanntes Problem der Computeralgebra. Es tritt zum Beispiel beim Vergleich von Charaktertafeln endlicher

Gruppen auf. Werden von zwei isomorphen endlichen Gruppen die Charaktertafel berechnet, so sind diese nur bis auf Zeilen- und Spaltenvertauschungen gleich. Umgekehrt ist die Ähnlichkeit von zwei Charaktertafeln ein starkes Indiz für die Isomorphie der zugehörigen Gruppen (jedoch im allgemeinen kein Beweis).

Eine algorithmische Lösung von Problem 2.9 wird von J. Leon (1991) in [55], 10. (g), angegeben. (Der Artikel ist in einer äußerst reichhaltigen Notation verfaßt, was den Zugang etwas erschwert.) Das Verfahren basiert auf der partitionsbasierten Backtracksuche in Permutationsgruppen. Dabei wird im Prinzip die Gruppe aller Kandidaten (S, T) entlang des Baums aller Nebenklassen bezüglich einer Stabilisatorkette durchlaufen. Die Suche orientiert sich zusätzlich entlang eines Turms von immer feineren Partitionen der Punktmenge $\{1..n+m\}$. Diese Technik wurde ursprünglich von B. McKay genau für das Graph-Isomorphismus-Problem entwickelt. Von J. Leon steht eine hoch optimierte Implementierung der Methoden in der Programmiersprache C zur Verfügung.

Eine einfache Implementierung zur Lösung von Problem 2.9 ist in das GAP-System als Funktion `TransformingPermutations` eingebaut. Diese Funktion basiert auf einer einfachen Backtracksuche ohne Verwendung von Partitionen. Diese Methode ist für die meisten Matrizen ausreichend. Nun zur Lösung des Problems vom Anfang dieses Kapitels:

2.11 Algorithmus Zu $A \in K^{n \times m}$ wird Problem 2.2 gelöst. Dazu führe folgende Schritt aus:

1. Bestimme die Blockstruktur von A.
2. Sortiere A und extrahiere die Blöcke $A^{(1)}$ bis $A^{(s)}$ mit Algorithmus 2.6.
3. Vergleiche die Blöcke paarweise und permutiere ihre Zeilen und Spalten so, daß ähnliche Blöcke identisch werden (Problem 2.9). Dies kann mit der GAP-Funktion `TransformingPermutations` oder mit dem Programm von J. Leon geschehen.
4. Ordne die korrigierten Blöcke so an, daß identische Kopien benachbart sind, und fasse diese zu Kronckerprodukten der Form $\mathbf{1}_{r_k} \otimes A^{(k)}$ zusammen. ∎

2.4 Zusammenfassung

In diesem Kapitel wurde das Problem betrachtet, wie die Zeilen und Spalten einer dünn besetzten Matrix so permutiert werden können, daß die vorhandene Blockstruktur explizit hervortritt und daher kompakt repräsentiert werden kann (Problem 2.2).

Dieses Problem wurde in vier Schritten gelöst (Algorithmus 2.11): Zuerst wird die Blockstruktur (Definition 2.3) quantitiv bestimmt (Algorithmus 2.7). Danach werden die Zeilen und Spalten so permutiert, daß die Blöcke getrennt

werden (Algorithmus 2.6). Im dritten Schritt werden die verschiedenen Blöcke miteinander unter Zeilen- und Spaltenpermutationen verglichen (Problem 2.9) und gegebenenfalls so korrigiert, daß aus Ähnlichem Gleiches wird. Schließlich wird im vierten Schritt die Reihenfolge der Blöcke so gewählt, daß gleiche Blöcke $A^{(k)}$ zu Kroneckerprodukten der Form $\mathbf{1}_r \otimes A^{(k)}$ zusammengefaßt werden können.

Das auftretende Teilproblem 2.9 ist nicht einfacher als das bekannte Problem GRAPH-ISOMORPHISMUS. Zu diesem ist bis heute kein polynomialer Algorithmus bekannt. Trotzdem ist die Lösung des Problem in der Praxis keine Schwierigkeit, da die Matrizen meist hinreichend klein sind und leistungsfähige Suchverfahren zur Verfügung stehen. Insbesondere existiert eine effiziente Implementierung von J. Leon (1991), die die von ihm entwickelte partitionsbasierte Backtracksuche in Permutationsgruppen verwendet.

Alle hier vorgestellten Algorithmen sind vom Autor zusammen mit Markus Püschel in GAP implementiert worden. Das Programm hat einen Umfang von etwa 0.5 kloc (engl. kilo lines of code). Für die Lösung von Teilproblem 2.9 wird die GAP-Bibliotheksfunktion `TransformingPermutations` verwendet.

3

Perm-Perm-Symmetrie

Das ist die schwere Zeit der Not,
Das ist die Not der schweren Zeit,
Das ist die schwere Not der Zeit,
Das ist die Zeit der schweren Not.
Adelbert von Chamisso, 1813.[1]

DIESES KAPITEL beschäftigt sich mit dem wichtigen Symmetrietyp der Perm-Perm-Symmetrie. Es geht dabei um das folgende

3.1 Problem (Perm-Perm-Symmetrie) *Gegeben sei die Matrix $M \in \mathrm{K}^{n\times m}$. Bestimme die Gruppe*

$$\mathsf{PermPerm}(M) = \mathsf{Sym}_{\mathsf{S}_n\times\mathsf{S}_n}(M) = \{(L,R) \in \mathsf{S}_n \times \mathsf{S}_m \mid L \cdot M = M \cdot R\}.$$

Mit anderen Worten: Es sollen alle Vertauschungen L der Zeilen gefunden werden, die sich durch eine Vertauschung R der Spalten rückgängig machen lassen. Eine Bemerkung zur Grundmenge K. Da nur Permutationen der Matrixeinträge betrachtet werden, ist die algebraische und topologische Struktur von K irrelevant. Die einzige benötigte Operation auf K ist der Vergleich ($x = y$). Der Vergleich sei daher entscheidbar.

Das Problem 3.1 ist in der Literatur bekannt. Es ist verwandt mit Problem 2.9 aus Kapitel 2. In dem Artikel [55] gibt J. Leon (1991) in Abschnitt 9.(g) eine Lösung von Problem 3.1 an. Das dort vorgeschlagene Verfahren ist eine spezielle Anwendung der von Leon entwickelten partitionsbasierten Backtracksuche in Permutationsgruppen.

Ebenfalls von J. Leon existiert eine hoch effiziente Implementierung dieses Verfahrens in der Programmiersprache C. Außerdem existiert vom Autor und M. Püschel eine Implementierung in der Programmiersprache GAP. Letzteres Programm ist wesentlich langsamer, jedoch zuverlässiger und benötigt keine externe GAP/C-Schnittstelle. Es ist als experimentelles Programm angelegt, um

[1] aus A. Thalmayr, „Das Wasserzeichen der Poesie“, [87], LII. Permutation

verschiedene Lösungsmethoden zu untersuchen. Vom Standpunkt des Praktikers aus gesehen ist das Problem mit dem Programm von J. Leon gelöst. Dieses Programm ist mühelos in der Lage, die Perm-Perm-Symmetrie von (100×100)-Matrizen zu berechnen. Obwohl das Problem PERM-PERM-SYMMETRIE in der Praxis also gut lösbar ist, hat das Problem eine hohe Komplexität:

3.2 Satz *Das Problem* PERM-PERM-SYMMETRIE *ist nicht einfacher als das Problem* GRAPH-ISOMORPHISMUS.

Beweis Für die Definition von GRAPH-ISOMORPHISMUS und den Begriff der Inzidenzmatrix $I(G)$ eines Graphen G sei auf das vorige Kapitel verwiesen (Seite 31). Gegeben seien zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$. Dann betrachte man die Matrix

$$M = \begin{bmatrix} I(G_1) & \mathbf{0} \\ \mathbf{0} & I(G_2) \end{bmatrix}$$

und bestimme $\mathsf{PermPerm}(M)$. Zwischen G_1 und G_2 gibt es genau dann einen Graph-Isomorphismus, wenn $\mathsf{PermPerm}(M)$ ein Element enthält, das $I(G_1)$ und $I(G_2)$ vertauscht. (Genauer gesagt, die zugehörigen Indexmengen als Mengen auf einander abbildet.) Dies kann in polynomialer Zeit in der Größe von M getestet werden. Ein polynomialer Algorithmus für PERM-PERM-SYMMETRIE impliziert also einen polynomialen Algorithmus für GRAPH-ISOMORPHISMUS. ∎

3.1 Skizzen der Lösungsmethoden

In diesem Abschnitt werden die bekannten Lösungsmethoden für Problem 3.1 kurz erklärt, wobei keine Vollständigkeit angestrebt wird.

Umbenennen der Einträge Da die Elemente von $\mathsf{PermPerm}(M)$ nur durch Vertauschen der Einträge auf M operieren, kommt es auf die Identität der Einträge von M nicht an. Deshalb werden in einem ersten Schritt alle auftretenden Werte $\{M_{ij} | i, j\}$ ersetzt durch Zahlen $\{1..s\}$. Daraus ergibt sich, daß die Größe der Eingabe für Problem 3.1 genau $nm \log_2(s)$ bit beträgt. (Bei der Perm-Block-Symmetrie ist die Eingabe a priori unbeschränkt.)

Gleiche Zeilen und Spalten Gleiche Zeilen von M können unabhängig von den Spaltenpermutationen beliebig miteinander vertauscht werden. Analoges gilt für gleiche Spalten. Nach Satz 1.4 auf Seite 9 bilden diese Operationen Normalteiler der Symmetriegruppe, die ein subdirektes Produkt mit vereinigter Faktorgruppe ist. Die Gruppe $\mathsf{PermPerm}(M)$ kann daher zusammengesetzt werden aus den Vertauschungen gleicher Zeilen, den Vertauschungen gleicher Spalten und den echt simultanen Vertauschungen. Konkret geschieht dies so: Zuerst wird die

Indexmenge der Zeilen von M bezüglich Gleichheit der Zeilen partitioniert. Die Young-Gruppe der Partition enthält alle Vertauschungen gleicher Zeilen. (Der Begriff der Young-Gruppe wird in Anhang A auf Seite 116 definiert.) Danach wird eine Menge von Repräsentanten der Zeilenblöcke ausgewählt, und die Matrix wird verkleinert. Analoges geschieht mit den Spalten. Die Young-Gruppen der Partitionen für Zeilen und Spalten bilden die Normalteiler des subdirekten Produkts. Im zweiten Schritt wird die Perm-Perm-Symmetrie der verkleinerten Matrix bestimmt. Ein Beispiel möge die Konstruktion verdeutlichen.

3.3 Beispiel (nach Beispiel 1.5) Die folgende Matrix M wird durch Zusammenfassen gleicher Zeilen und Spalten verkleinert.

$$M = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

Die Partition der Zeilen ist $\{\{1,3\},\{2\},\{4\}\}$, die der Spalten $\{\{1,4\},\{2\},\{3\}\}$. Die zugehörigen Young-Gruppen sind $\langle(1\,3)\rangle$ und $\langle(1\,4)\rangle$. Durch Wahl von Repräsentanten $\{1,2,4\}$ und $\{1,2,3\}$ gelangt man zur Matrix

$$M^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Offensichtlich ist $\mathsf{PermPerm}(M^{(1)}) = \langle((2\,3),(2\,3))\rangle$. Durch Rückumbenennung der Punkte und Hinzufügen der Young-Gruppen ergibt sich die Symmetriegruppe wie sie in Beispiel 1.5 angegeben wurde. ∎

Verwendung von Zeilen- und Spaltenbilanzen Ein Element $(L,R) \in \mathsf{PermPerm}(M)$ genügt der Gleichung $LMR^{-1} = M$, wobei L die Zeilen von M permutiert und R^{-1} die Spalten. Wird nun Zeile i durch L auf Zeile j abgebildet, so ist dies nur möglich, wenn in den Zeilen i und j die gleichen Werte jeweils gleich oft vorkommen. Die selbe Einschränkung gilt analog für die Operation von R^{-1} auf den Spalten. Diese Beobachtung führt auf die wichtigste Problemreduktion der Perm-Perm-Symmetrie:

Zwei Vektoren x und y sollen *bilanzgleich* heißen, wenn sie die gleichen Werte gleich oft enthalten (das heißt $\mathrm{sort}(x) = \mathrm{sort}(y)$). Die Bilanz ist die Mehrfachmenge der Komponenten. Sie ist die allgemeinste Invariante eines Vektors bezüglich beliebiger Vertauschung der Komponenten. Formal kann dies ausgedrückt werden durch eine universelle Abbildungseigenschaft: Jede Invariante eines Vektors bezüglich Vertauschung der Komponenten ist lediglich eine Funktion der Mehrfachmenge aus den Komponenten.

Die Menge der Zeilen von M wird bezüglich Bilanzgleichheit partitioniert. Die in den Blöcken auftretenden Werte von M dürfen ohne Rücksicht auf andere Blöcke umbenannt werden. Dann wird die Menge der Spalten bezüglich Bilanzgleichheit partitioniert, und die Blöcke werden ebenfalls disjunkt umbenannt. Durch diese Umbenennung der Spalten können die Zeilenbilanzen wieder verändert werden. Daher wird wieder die Menge der Zeilen bezüglich Bilanzgleichheit partitioniert, und so weiter. Der Prozeß stagniert nach höchstens $1 + \min(n, m)$ Iterationen, denn in jedem Schritt wird ein Zeilen- oder Spaltenblock aufgespalten. Am Ende ist M eine Matrix von bilanzhomogenen Matrizen, die paarweise disjunkte Wertemengen besitzen. Ein Beispiel möge die Methode veranschaulichen.

3.4 Beispiel Die Matrix M wird fortlaufend mit Hilfe der Zeilen- und Spaltenbilanzen partitioniert, und die Werte werden disjunkt umbenannt. Da eine Symmetrie nur Zeilen und Spalten mit gleicher Bilanz aufeinander abbildet, gilt $\mathsf{PermPerm}(M^{(k+1)}) = \mathsf{PermPerm}(M^{(k)})$.

$$M = \left[\begin{array}{cccc} 2 & 1 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 \end{array}\right] \hookrightarrow \{\{1,2,3\},\{4\}\} \times \{\{1,2,3\},\{4\}\}$$

$$\hookrightarrow M^{(1)} = \left[\begin{array}{ccc|c} 2 & 1 & 1 & 4 \\ 1 & 1 & 2 & 4 \\ 1 & 2 & 2 & 3 \\ \hline 6 & 6 & 5 & 7 \end{array}\right] \hookrightarrow \{\{1,2\},\{3\},\{4\}\} \times \{\{1,2\},\{3\},\{4\}\}$$

$$\hookrightarrow M^{(2)} = \left[\begin{array}{cc|c|c} 2 & 1 & 3 & 5 \\ 1 & 1 & 4 & 5 \\ \hline 6 & 7 & 8 & 9 \\ \hline 10 & 10 & 11 & 12 \end{array}\right] \hookrightarrow \{\{1\},\{2\},\{3\},\{4\}\}^2$$

$$\hookrightarrow M^{(3)} = \left[\begin{array}{c|c|c|c} 1 & 2 & 3 & 4 \\ \hline 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \hline 13 & 14 & 15 & 16 \end{array}\right].$$

Da $M^{(3)}$ vollständig zerlegt ist, gilt $\mathsf{PermPerm}(M) = \{(\mathrm{id}, \mathrm{id})\}$. ∎

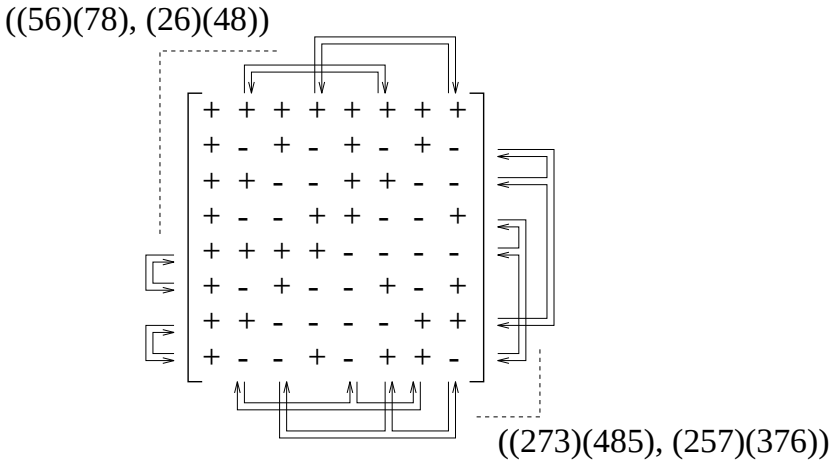
Die Zerlegung einer Matrix in bilanzhomogene Untermatrizen ist das diskrete Analogon zum Konzept der „taktischen Zerlegung“ (engl. tactical decomposition) in der Design-Theorie. Dabei wird eine reell-wertige Matrix in Untermatrizen mit konstanten Zeilen- und Spaltensummen zerlegt (Beth, Jungnickel und Lenz (1985), [13], Kap. III, §4, Def. 4.13).

Backtracksuche Die bisher besprochenen Methoden verkleinern das Problem substantiell, lösen es aber im allgemeinen nicht. Dies kann mit einem Backtrack-Suchverfahren bewältigt werden. Die Backtracksuche ist ein Standardverfahren

für Permutationsgruppen, sie ist in dem Buch von Butler, [16], ausführlich beschrieben. Eine wesentliche Verfeinerung ist von Leon (1991), [55], vorgeschlagen worden und verwendet einen Turm von Partitionen. Das Verfahren sucht zunächst nur Vertauschungen großer Blöcke. Danach werden die Blöcke zerkleinert und es werden die Vertauschungen der kleineren Blöcke durchsucht usw. Durch die Verwendung geeigneter Partitionen wird der Suchbaum oft sehr stark verkleinert. Diese Methode liegt Leons Programm zur Bestimmung von $\mathsf{PermPerm}(M)$ zugrunde. Die Idee der Backtrack-Suche ist wie folgt:

Angenommen $(L, R) \in \mathsf{PermPerm}(M)$. Dann bildet L die erste Zeile auf Zeile 1^L ab. Folglich permutiert R^{-1} die Zeile $M_{1,*}$ so, daß sie die Zeile $M_{1^L,*}$ ergibt. Für jedes $1^L \in \{1..n\}$ wird das Suchproblem also reduziert auf eine Untergruppe. Durch Rekursion ergibt sich ein Suchbaum von Möglichkeiten. Die Kunst besteht wie immer darin, Suchäste möglichst früh abzuschneiden. Bei der hier betrachteten Anwendung ist dies in der Tat sehr effizient möglich.

3.5 Beispiel (Automorphismen der Fano-Ebene) Es sei $M \in \{\Leftrightarrow 1, 1\}^{8\times 8}$ die Walsh-Hadamard-Transformation auf acht Punkten. Die untere-rechte (7×7)-Matrix von M ist die Inzidenzmatrix der Fano-Ebene. Dann ist $\mathsf{PermPerm}(M) \cong \mathsf{PSL}_3(\mathbb{F}_2)$. Die folgende Abbildung zeigt die Matrix M $(+ = 1, \Leftrightarrow = \Leftrightarrow 1)$ und zwei Erzeuger der Gruppe $\mathsf{PermPerm}(M)$.



Außer der ersten Zeile und der ersten Spalte ist die Matrix bilanzhomogen und damit nur mit Backtracksuche zu lösen. ∎

3.2 Wo die Perm-Perm-Symmetrie auftritt

In diesem Abschnitt sind einige Gebiete aufgeführt, in denen die Perm-Perm-Symmetrie eine wesentliche Rolle spielt. Die Querverbindungen sollen hier jedoch nicht tiefer verfolgt werden.

Autormorphismen von Designs Das Beispiel 3.5 behandelt (im wesentlichen) die Inzidenzmatrix der projektiven Ebene über $\mathbb{F}_2$ (die „Fano-Ebene"ť). Dies ist ein spezieller Fall einer Inzidenzstruktur (V, B, I) mit der Punktmenge V, einer dazu disjunkten Menge von Blöcken B und einer Inzidenzrelation $I \subseteq V \times B$. Im Beispiel 3.5 sind die Punkte die eindimensionalen Unterräume von $\mathbb{F}_2^3$, die Blöcke sind die zweidimensionalen Unterräume und die Inzidenzrelation ist $\{(v, b) \mid v \cap b \neq 0\}$ (nach Beth, Jungnickel und Lenz (1985) [13], Ch. I, §2, Prop. 2.3). Die Perm-Perm-Symmetrie einer Inzidenzmatrix ist genau die Automorphismengruppe der zugehörigen Inzidenzstruktur. Ein wichtiger Spezialfall sind Graphen und deren Automorphismen.

Automorphismen von Charaktertafeln Sei G eine endliche Gruppe mit Konjugationsklassen $g_1^G, \ldots, g_h^G$ und irreduziblen Charakteren $\chi_1, \ldots, \chi_h$ über dem Grundkörper $\mathbb{C}$. Die Charaktertafel von G ist dann die $(h \times h)$-Matrix $M = [\chi_i(g_j)|i, j]$. Nach einem Lemma von Brauer operiert die Automorphismengruppe von G auf der Charaktertafel durch Permutation (Huppert, [43], Bd. I, Kap. V, §13, Satz 13.5), wobei die inneren Autormorphismen trivial operieren. Umgekehrt permutiert eine Symmetrieoperation $(L, R) \in \mathsf{PermPerm}(M)$ die Charaktere mit L und die Konjugationsklassen mit R^{-1}, so daß sich wieder M ergibt. Es liegt nahe zu fragen, wie sich die Perm-Perm-Symmetrie der Charaktertafel zur Automorphismengruppe von G verhält. Einige Antworten sind bekannt:

Es gibt Charaktertafeln mit Permutationen, die durch keine Automorphismen der Gruppe erzeugt sind. Beispiel: M_{11} besitzt keine äußeren Automorphismen, aber $((\chi_3\ \chi_4), (8\mathrm{A}\ 8\mathrm{B}))$ ist eine Symmetrie der Charaktertafel. (Die Bezeichnungen beziehen sich auf den ATLAS, [21], Seite 18.)

Nach einem Ergebnis von Burnside (1913), [15], gibt es Gruppen mit Automorphismen, die keine inneren sind, aber trotzdem alle Konjugationsklassen stabilisieren[2] Daher enthält die Perm-Perm-Symmetrie der Charaktertafel nicht notwendigerweise die gesamte äußere Automorphismengruppe, sondern nur einen Normalteiler davon. Ein Beispiel für diese Situation ist die Gruppe `SolvableGroup(32, 44)` aus dem GAP-Katalog der Gruppen bis zur Ordnung 100 (die Notation von Erzeugendensystemen ist in Anhang A auf Seite 114 erläutert):

$$G = \mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_8) = \langle a; b; t, s \mid b^a = bs^2, t^a = ts^{-1}, s^a = s^{-1} \rangle.$$

Diese Gruppe besitzt einen äußeren Automorphismus α, der alle Konjugationsklassen stabilisiert; er ist definiert durch $[a; b; t, s]^\alpha = [a; bs^2; t, s]$.

Gruppenzirkulanten Sei $G = \{g_1, \ldots, g_n\}$ eine reguläre Permutationsgruppe auf $\{1..n\}$. Die Permutationen werden mit ihren Permutationsmatrizen identifiziert. (Die Konvention der verwendeten Darstellung ist in Anhang A erklärt.)

[2] Diesen Hinweis verdanke ich Joachim Neubüser.

Dann nennen wir eine Matrix

$$\mathrm{circ}_G(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i \cdot g_i, \quad \text{mit } a_i \in \mathbb{C} \text{ für alle } i \in \{1..n\},$$

eine G-Zirkulante. Die Matrix M aus Beispiel 1.8 auf Seite 14 ist eine D_6-Zirkulante. Da G regulär ist, überdecken die Permutationsmatrizen aus G disjunkt alle $(n \times n)$-Komponenten mit Eins-Einträgen. Außerdem ist die Multiplikation von G-Zirkulanten die Faltung über der Gruppe G, also die Multiplikation in der Gruppenalgebra $\mathbb{C}[G]$. Sind die Koeffizienten a_i paarweise verschieden, so ist die Perm-Perm-Symmetrie von $\mathrm{circ}_G(a_i|i)$ konjugiert zu G. Sind die Koeffizienten nicht paarweise verschieden, so enthält die Perm-Perm-Symmetrie die Gruppe G. Daher kann durch Bestimmung der Perm-Perm-Symmetrie die Gruppe rekonstruiert werden, von der die Matrix eine Zirkulante ist. Diese Struktur kann für verschiedene Anwendungen nutzbringend verwendet werden, wie in dem Buch von Clausen und Baum, [19], Kapitel 10, näher ausgeführt wird.

3.3 Zusammenfassung

In diesem Kapitel wurde das Problem der Perm-Perm-Symmetrie einer Matrix betrachtet (Problem 3.1). Die Komplexität des Problems ist nicht geringer als GRAPH-ISOMORPHISMUS (Satz 3.2). In der Praxis ist das jedoch kein Hindernis, da starke Problemreduktionen und leistungsfähige Suchverfahren bekannt sind (Abschnitt 3.1). Schließlich wurden einige Anwendungsgebiete angegeben, in denen die Perm-Perm-Symmetrie auftritt (Abschnitt 3.2).

Die Methoden wurden vom Autor zusammen mit Markus Püschel in der Programmiersprache GAP implementiert. Das Programm umfaßt etwa 1.3 kloc. Für größere Matrizen steht eine hoch optimierte Implementierung von J. Leon (1991) in der Programmiersprache C zu Verfügung.

4

Perm-Mat-Symmetrie

Mit Blick auf die Anwendung bei der Perm-Irred-Symmetrie wird in diesem Kapitel der Perm-Mat-Symmetrietyp definiert und betrachtet. Die Bezeichnung „Perm-Mat“ soll dabei andeuten, daß links Permutationen stehen und rechts beliebige invertierbare Matrizen. Als Spezialisierung der Definition 1.1 geht es in diesem Kapitel um die folgende Form von Symmetrie:

4.1 Definition *Die* ***Perm-Mat-Symmetrie*** *von* $M \in \mathrm{K}^{n\times m}$ *ist gegeben durch*

$$\mathsf{PermMat}(M) = \Big\{(L,R) \in \mathsf{S}_n \times \mathsf{GL}_m(\mathrm{K}) \;\Big|\; L \cdot M = M \cdot R\Big\}.$$

Betrachtet werden also alle Permutationen L der Zeilen von M, so daß die Spalten durch lineare invertierbare Operationen R wieder auf die Gestalt von M gebracht werden können.

In dieser Allgemeinheit ist $\mathsf{PermMat}(M)$ schwer im Rechner darzustellen. Das Problem ist, daß die Menge $\mathsf{GL}_m(\mathrm{K})$ sehr unhandlich werden kann. Falls M invertierbar ist, ist zudem jede Permutation L in der Symmetriegruppe, denn $L \cdot M = M \cdot (M^{-1}LM)$. Um das Problem algorithmisch zugänglich zu machen, muß also die Menge der betrachteten Matrizen M eingeschränkt werden.

Dazu seien Π_L und Π_R die Projektionen $(L,R) \mapsto L$ und $(L,R) \mapsto R$. Nach Satz 1.4 (auf Seite 9) ist $\mathsf{PermMat}(M)$ ein subdirektes Produkt mit vereinigter Faktorgruppe. Der rechte Normalteiler des subdirekten Produkts ist

$$\ker \Pi_L = \{(\mathrm{id}, R) \mid M = MR\}.$$

Wird dieser Normalteiler trivial, dann entartet das subdirekte Produkt. In der entarteten Situation ist Π_L ein Isomorphismus, denn sein Kern verschwindet. Das heißt, die Paare (L,R) in der Symmetriegruppe werden eindeutig bestimmt durch die Permutation L, das heißt die $(m \times m)$-Matrix R kann aus L und M berechnet werden. Es bleibt zu erwähnen, daß der Normalteiler $\ker \Pi_L$ genau dann verschwindet, wenn M den Rang m hat. Damit lautet das in diesem Kapitel zu lösende algorithmische Problem:

4.2 Problem (Perm-Mat-Symmetrie) *Gegeben eine Matrix $M \in \mathrm{K}^{n\times m}$ vom Rang $m \leq n$. Bestimme die Permutationsgruppe $\Pi_L(\mathsf{PermMat}(M))$ sowie den Homomorphismus $L \mapsto R$ mit $LM = MR$.*

4.3 Beispiel Sei $M = [1, \Leftrightarrow\!1, 1, \Leftrightarrow\!1]^T \in \mathbb{Q}^{4\times 1}$. Dann ist

$$\Pi_L(\mathsf{PermMat}(M)) = \langle (1,3), (2,4), (1,2)(3,4)\rangle.$$

Der zugehörige Homomorphismus $L \mapsto R$ ist definiert durch

$$(1,3) \mapsto 1,\ (2,4) \mapsto 1 \ \text{ und } \ (1,2)(3,4) \mapsto \Leftrightarrow\!1.$$

4.1 Permutationen gleicher Zeilen

Das Problem PERM-MAT-SYMMETRIE wird in zwei Schritten gelöst. Als erstes wird

$$\Pi_L(\ker \Pi_R) = \{L \mid LM = M\}$$

bestimmt. Als Kern eines Homomorphismus ist dies ein Normalteiler von $\Pi_L(\mathsf{PermMat}(M))$. Als zweites wird die Faktorgruppe bezüglich des Normalteilers bestimmt. Doch zunächst zum ersten Problem. Die Menge $\Pi_L(\ker \Pi_R)$ enthält offensichtlich genau die Permutationen, die gleiche Zeilen von M miteinander vertauschen. Dies ist die Young-Gruppe der Partition $\{1..n\}/\sim$ mit der Äquivalenzrelation

$$i_1 \sim i_2 \Leftrightarrow M_{i_1,*} = M_{i_2,*} \quad \text{für } i_1, i_2 \in \{1..n\}.$$

(Die Young-Gruppe wird in Anhang A auf Seite 116 erklärt.)

4.4 Algorithmus (Gleiche Zeilen) Gegeben $M \in \mathrm{K}^{n\times m}$ mit $m \leq n$. Der Algorithmus konstruiert $\Pi_L(\ker \Pi_R)$ für $\mathsf{PermMat}(M)$: Konstruiere die Partition $p = \{1..n\}/\sim$, mit der wie oben definierten Äquivalenzrelation $\sim$. Dann konstruiere die Young-Gruppe $\mathsf{S}(p)$.

4.2 Suche nach der Faktorgruppe

Nun zum zweiten Teilproblem bei der Berechnung von $\mathsf{PermMat}(M)$. Die Zeilen von M sind in den einzelnen Blöcken $b_1, \ldots, b_r$ der Partition p identisch. Man wähle nun eine Menge von Punkten $P = \{p_1, \ldots, p_r\}$ mit $p_i \in b_i$ und bilde die $(r \times m)$-Untermatrix $M_{P,*}$ aus den zugehörigen Zeilen von M. Für $M_{P,*}$ ist nach Konstruktion auch $\ker \Pi_L$ ist trivial; das heißt, sowohl Π_L als auch Π_R sind Isomorphismen. Durch die Verkleinerung der Matrix M auf $M_{P,*}$ kann also geschickt zur gewünschten Faktorgruppe von $\mathsf{PermMat}(M)$ übergegangen werden.

Der Algorithmus zur Bestimmung der Faktorgruppe basiert auf der folgenden Beobachtung: *Die Permutation L ist eindeutig bestimmt durch ihr Bild auf einer Basis der Länge m und jede linear unabhängige Menge von Zeilen aus M bildet eine solche.* Da der Rang von M gleich m ist gibt es ein m-Tupel $I = [i_1, \ldots, i_m]$ von ganzen Zahlen so daß die $(m \times m)$-Matrix $M_{I,*}$ invertierbar ist. Mit $I^L = [i_1^L, \ldots, i_m^L]$ sei das Bildtupel von I unter der Permutation L bezeichnet. Der folgende Algorithmus rekonstruiert die Permutation L eindeutig aus ihrem Basisbild I^L.

4.5 Algorithmus (Perm aus Basisbild) Gegeben ist eine Matrix $M \in \mathrm{K}^{n \times m}$, $m \leq n$, vom Rang m. Die Matrix enthalte keine gleichen Zeilen. Weiterhin ist ein m-Tupel I gegeben, so daß $M_{I,*}$ invertierbar ist. Der Algorithmus bestimmt zu jedem m-Tupel J die Permutation $L \in \mathsf{S}_n$ mit $I^L = J$, oder zeigt, daß es eine solche nicht gibt. Berechne dazu

$$\tilde{M} = M \cdot M_{I,*}^{-1} \cdot M_{J,*}$$

und suche eine Permutation L, die die Zeilen von M auf die Zeilen von $\tilde{M}$ abbildet, also $LM = \tilde{M}$. Solch ein L existiert genau dann, wenn die Mehrfachmenge der Zeilen von $\tilde{M}$ gleich der von M ist.

Beweis Seien M und I wie in Algorithmus 4.5. Betrachte $L \in \mathsf{S}_n$ und $R \in \mathsf{GL}_m(\mathrm{K})$ mit $LM = MR$. Dann folgt für die Untermatrizen

$$(LM)_{I,*} = M_{I^L,*} = M_{I,*}R = (MR)_{I,*} \quad \Rightarrow \quad R = M_{I,*}^{-1} \cdot M_{I^L,*}.$$

Die Matrix R wird also durch $J = I^L$ eindeutig bestimmt, denn I ist fest. Umgekehrt bestimmt jedes m-Tupel J eine Matrix $R = M_{I,*}^{-1} \cdot M_{J,*}$. Die zugehörige Permutation L, falls es eine solche gibt, erfüllt die Gleichung

$$L \cdot M = M \cdot M_{I,*}^{-1} M_{J,*}.$$

Da M nach Voraussetzung keine gleichen Zeilen enthält, ist L durch diese Gleichung eindeutig bestimmt, falls es existiert. ∎

In einer tatsächlichen Implementierung von Algorithmus 4.5 kann die Matrix $M \cdot M_{I,*}^{-1}$ vorberechnet werden, denn sie hängt nicht von der eigentlichen Eingabe J ab. Außerdem sollte der Test, ob es ein L mit $LM = \tilde{M}$ gibt, inkrementell erfolgen: Die Komponenten $\tilde{M}_{i,j}$ werden eine nach der anderen ausgerechnet, und es wird sofort getestet, ob sie überhaupt in der Menge $\{M_{i,j} \mid i, j\}$ liegen. Ist eine Zeile $\tilde{M}_{i,*}$ komplett, so wird getestet, ob sie in der Menge $\{M_{i,*} \mid i\}$ liegt. Dieses Vorgehen bewirkt einen schnellen Abbruch, falls kein L existiert. Nun zur kombinatorischen Suche.

4.6 Algorithmus (Suche) Gegeben $M \in \mathrm{K}^{n\times m}$ vom Rang $m \le n$ ohne gleiche Zeilen. Der Algorithmus findet die Permutationsgruppe

$$\Pi_L(\mathsf{PermMat}(M)) = \{L \mid \exists R \in \mathsf{GL}_m(\mathrm{K}) : LM = MR\}$$

sowie den Isomorphismus $L \mapsto R$ mit $LM = MR$. Der Algorithmus pflegt eine Gruppe G, die mit jeder gefundenen Permutation L vergrößert wird und schließlich die gesamte Gruppe $\Pi_L(\mathsf{PermMat}(M))$ enthält. Es werden nur solche Basisbilder J weiter betrachtet, die nicht schon mit der bekannten Gruppe G aus I erreichbar sind. Der Algorithmus:

Wähle I wie in Algorithmus 4.5;
$G := \{\mathrm{id}\}$;
`for` $J \in \{1..n\}^m$ ohne Wiederholungen `do`
  `if` $J \notin$ Orbit von I unter G `then`
    $L :=$ ⟨ Algorithmus 4.5 auf M, I, J ⟩;
    `if` L existiert `then`
      $G := \langle G \cup \{L\}\rangle$
    `fi`
  `fi`
`od`.

Am Ende ist $G = \Pi_L(\mathsf{PermMat}(M))$ mit Isomorphismus $L \mapsto M_{I,*}^{-1} M_{I^L,*}$. (Implementierungshinweis: Der Orbit-Test kann mit der Aufzählung der J verschränkt werden. Dies entfernt ganze Teilbäume bei der Aufzählung.) ∎

4.7 Algorithmus (Perm-Mat-Symmetrie) Gegeben eine Matrix $M \in \mathrm{K}^{n\times m}$ vom Rang $m \le n$. Der Algorithmus bestimmt $\Pi_L(\mathsf{PermMat}(M))$ sowie den Homomorphismus $L \mapsto R$ und löst damit Problem 4.2:

1. Reduziere das Problem 4.2 mit Hilfe von Algorithmus 4.4 auf den Fall, daß M keine gleichen Zeilen mehr enthält. Dabei entsteht eine Young-Gruppe $\mathsf{S}(p)$ sowie eine Liste $P \subseteq \{1..n\}$, so daß $M_{P,*}$ keine gleichen Zeilen enthält.

2. Bestimme mit Algorithmus 4.6 die Gruppe $\Pi_L(\mathsf{PermMat}(M_{P,*}))$ sowie den Isomorphismus $\varphi = L \mapsto R$ dieser Gruppe in die $\mathsf{GL}_m(\mathrm{K})$.

3. Vereinige die beiden Teile. Die gesuchte Gruppe besteht aus dem Normalteiler $\mathsf{S}(p)$ und der zugehörigen Faktorgruppe $\Pi_L(\mathsf{PermMat}(M_{P,*}))$, mit geeignet umbenannten Punkten. Der gesuchte Homomorphismus entsteht aus φ durch triviale Fortsetzung auf der Young-Gruppe und geeignete Umbenennung der Punkte. ∎

Der Aufwand von Algorithmus 4.7 ist im schlechtesten Fall sehr hoch; es müssen dann in Algorithmus 4.6 alle $\binom{n}{m}m!$ vielen J durchlaufen werden. Der im Mittel

zu erwartende Aufwand ist viel geringer, da die Eigenschaft der Permutation L ein passendes R zu besitzen sehr restriktiv ist. Leider ist der zu erwartende Aufwand nur schwer formal abzuschätzen, da er stark von der Lösung abhängt und viele Ausgaben in Frage kommen. Immerhin gilt

4.8 Satz *Für festes m benötigt Algorithmus 4.7 nicht mehr als $O(n^{m+1})$ Operationen im Grundkörper.*

Beweis Es werden $\binom{n}{m}m!$ viele mögliche Basisbilder J betrachtet. Für jedes J wird Algorithmus 4.5 ausgeführt. Das erfordert im wesentlichen eine Matrixmultiplikation einer $(n \times m)$-Matrix mit einer $(m \times m)$-Matrix, also $O(m^2 n)$. Da $\binom{n}{m}$ ein Polynom m-ten Grades in n ist, ergibt sich der behauptete Aufwand. ∎

4.3 Zusammenfassung

In diesem Kapitel wurde der Symmetrietyp der Perm-Mat-Symmetrie untersucht (Definition 4.1). Um diesen Symmetrietyp algorithmisch zugänglich zu machen, wird eine milde Einschränkung betrachtet. Diese besteht in der Forderung nach maximalem Spaltenrang (Problem 4.2).

Es wurde ein Algorithmus zur Bestimmung der Perm-Mat-Symmetrie in zwei Schritten angegeben (Algorithmus 4.7): Zuerst werden die Permutationen gleicher Zeilen bestimmt (Algorithmus 4.4). Dann wird die verbliebene Faktorgruppe mit einer kombinatorischen Suche bestimmt (Algorithmus 4.6). Die kombinatorische Suche basiert auf der Beobachtung, daß die Permutation bereits durch ihr Bild auf einer kurzen Basis eindeutig bestimmt ist (Algorithmus 4.5).

Der Aufwand zur Bestimmung von $\mathsf{PermMat}(M)$ für eine Matrix $M \in \mathrm{K}^{n \times m}$ vom Rang $m \leq n$ ist im schlechtesten Fall exponentiell. Der zu erwartende Aufwand ist viel geringer als der Aufwand im schlechtesten Fall; er läßt sich aber nur schwer abschätzen. Alle vorgeschlagenen Algorithmen wurden vom Autor in der Programmiersprache GAP implementiert. Das Programm hat einen Umfang von etwa 1.0 kloc.

5

Perm-Irred-Symmetrie

Door meten tot weten
(dt. Durch Messen zum Wissen)
Heike Kamerlingh-Onnes, 1853–1926.

Dieses Kapitel behandelt den Perm-Irred-Symmetrietyp. Sei dazu M eine gegebene invertierbare Matrix und G eine Gruppe von Permutationsmatrizen L. Es werden alle Produkte der Form $M^{-1}LM$ betrachtet. Einige dieser Produkte zerfallen in Blöcke, schematisch $M^{-1}LM =$  . Es wird nun versucht, die Gruppe G so zu wählen, daß M alle $L \in G$ simultan in möglichst kleine Blöcke zerlegt — im Extremfall sind alle Produkte $M^{-1}LM$ Diagonalmatrizen. Die *Perm-Block-Symmetrie* von M erfaßt systematisch alle solchen Zerlegungen. Die *Perm-Irred-Symmetrie* isoliert jene Gruppen, welche durch keine andere Matrix weiter zerlegt werden können. Diese informelle Erklärung wird später formalisiert werden, nachdem die benötigten quantitativen Begriffe definiert wurden.

5.1 Beispiel (Translationsinvarianz der DFT) Sei $M = \mathrm{DFT}_n$ die diskrete Fourier-Transformation, wie in Abschnitt 7.1 definiert. Bekanntlich bewirkt bei der DFT eine zyklische Verschiebung des Eingangsignals eine Phasenverschiebung des Ausgangssignals. Formal kann dies geschrieben werden als $L \cdot M = M \cdot R$ mit

$$\begin{aligned} L &= (1 \ldots n) \quad \text{und} \\ R &= \mathrm{diag}(1, \omega, \omega^2, \ldots, \omega^{n-1}), \end{aligned}$$

wobei ω die zur Definition von M verwendete primitive n-te Einheitswurzel ist. (Zur Notation von Permutation[smatriz]en konsultiere man Anhang A.) In der Sprechweise der Perm-Irred-Symmetrie bedeutet diese Eigenschaft: Die Permutationsgruppe $G = \langle L \rangle \cong \mathsf{Z}_n$ ist eine Perm-Irred-Symmetrie der Matrix M mit Blöcken der Größe eins. ∎

Die Perm-Irred-Symmetrie entspricht dem „Symmetrie Typ I“ aus der Dissertation [64] von T. Minkwitz (1993). In seiner Arbeit zeigt Minkwitz unter an-

derem, wie zu einer linearen Abbildung mit Perm-Irred-Symmetrie ein schneller Algorithmus konstruiert werden kann. Zudem hat Minkwitz die Algorithmenkonstruktion für eine Perm-Perm-Symmetrie auf den Perm-Irred-Fall zurückgeführt. Dem Perm-Irred-Fall kommt daher eine zentrale Rolle für die symmetriebasierte Algorithmensynthese zu.

Der Rest dieses Kapitels ist wie folgt aufgebaut: In Abschnitt 5.1 wird der relevante Begriff von Blockstruktur definiert, und es werden dessen fundamentalen Eigenschaften formuliert und bewiesen. In Abschnitt 5.2 wird untersucht, welche Blockstrukturen eine feste Matrix erzeugen kann. Danach folgt in Abschnitt 5.3 die Definition der Perm-Block-Symmetrie und der Perm-Irred-Symmetrie. Abschnitt 5.4 behandelt die algorithmische Seite der Perm-Block-Symmetrie und stellt zwei alternative Ansätze zu deren Bestimmung vor. Die Zusammenfassung in Abschnitt 5.5 beendet das Kapitel.

5.1 Konjugierte Blockdiagonalstruktur

Zentral für die Perm-Irred-Symmetrie ist der Begriff der Blockstruktur. Der in diesem Kapitel wesentliche Begriff von Blockstruktur zeichnet sich dadurch aus, daß er verträglich ist mit der Matrixmultiplikation. Das heißt, bei der Multiplikation von Matrizen mit kompatibler Blockstruktur entsteht wieder eine Matrix mit dieser Blockstruktur (oder einer feineren).

Eine quadratische Matrix $A \in \mathrm{K}^{n\times n}$ heiße *blockdiagonal*, wenn sie von der Form $A^{(1)} \oplus \cdots \oplus A^{(r)}$ mit quadratischen Teilmatrizen $A^{(k)}$ und $r \geq 1$ ist. Die Matrix A heiße *konjugiert blockdiagonal*, wenn es eine Permutation $S \in \mathsf{S}_n$ gibt, so daß $S^{-1}AS$ blockdiagonal ist. Mit dem Begriff der konjugierten Blockdiagonalität wird die Theorie der Perm-Irred-Symmetrie unabhängig von einer Umbenennung der Punkte $\{1..n\}$ in der S_n. Die folgende Definition führt ein quantitatives Maß für konjugierte Blockdiagonalität ein.

5.2 Definition *Die **konjugierte Blockdiagonalstruktur** der Matrix $A \in \mathrm{K}^{n\times n}$ ist die Partition*

$$\mathrm{kbs}(A) = \{1..n\}/\sim^*,$$

wobei $\sim^*$ der reflexiv-symmetrisch-transitive Abschluß der Relation $\sim$ ist. Die Relation $\sim$ ist definiert durch $i \sim j \Leftrightarrow A_{ij} \neq 0$.

Zur Erinnerung sei an dieser Stelle erwähnt, daß die Menge $\mathrm{Part}(\{1..n\})$ aller Partitionen der Menge $\{1..n\}$ einen endlichen Verband bildet, vergleiche Anhang A. Zum Vergleich von kbs, cbs (aus Kapitel 6) und bs (aus Kapitel 2) sei auf Seite 19 der Einleitung verwiesen.

5.3 Beispiel Eine Matrix A mit $\mathrm{kbs}(A) \sqsubseteq \{\{1,3\},\{2,5\},\{4\}\}$ hat die folgende Gestalt ($\cdot$ bezeichnet 0, $*$ verdeckt einen beliebigen Eintrag):

$$A = \begin{bmatrix} * & \cdot & * & \cdot & \cdot \\ \cdot & * & \cdot & \cdot & * \\ * & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & * & \cdot \\ \cdot & * & \cdot & \cdot & * \end{bmatrix}.$$

Für Permutationsmatrizen A sind die Blöcke in $\mathrm{kbs}(A)$ genau die Zyklen der zu A gehörenden Permutation. ∎

Das nächste Lemma zeigt, daß die kbs wirklich den informellen Begriff der konjugierten Blockdiagonalität mißt. Dazu wird untersucht, wie sich die kbs unter direkter Summe und Konjugation mit einer Permutationsmatrix transformiert.

5.4 Lemma Für alle $A \in \mathrm{K}^{n\times n}$, $B \in \mathrm{K}^{m\times m}$ und $S \in \mathsf{S}_n$ gilt

1. $\mathrm{kbs}(A \oplus B) = \mathrm{kbs}(A) \cup \{\{n+i \mid i \in b\} \mid b \in \mathrm{kbs}(B)\}$.
2. $\mathrm{kbs}(S^{-1}AS) = \{\{i^S \mid i \in b\} \mid b \in \mathrm{kbs}(A)\}$.

Beweis

1. Aus der Lage der Nullen in $A \oplus B$ erkennt man sofort, daß $\mathrm{kbs}(A) \sqsubseteq \{\{1..m\},\{m+1..m+n\}\}$. Die feinere Struktur auf $\{1..m\}$ wird durch $\mathrm{kbs}(A)$ bestimmt, die auf $\{m+1..m+n\}$ durch $\mathrm{kbs}(B)$.

2. Es ist $(S^{-1}AS)_{ij} = A_{i^{S^{-1}},j^{S^{-1}}}$. Daher wirkt S in der Definition von kbs einfach durch Umbenennung der Punkte $\{1..n\}$. ∎

Hat eine Matrix eine nicht-triviale konjugierte Blockstruktur, so enthält sie an bestimmten Stellen Nulleinträge. Zudem können an manchen Stellen weitere Nullen auftreten. Die Matrix besitzt also ein bestimmtes Muster aus Null- oder Nicht-Null-Einträgen, wie es in Beispiel 5.3 gezeigt ist. Die konjugierte Blockstruktur hat nun die Eigenschaft, dieses Muster bei Matrixmultiplikation im wesentlichen zu erhalten. Dies ist die Aussage des folgenden wichtigen Lemmas.

5.5 Lemma Es gilt

1. $\mathrm{kbs}(A^{-1}) = \mathrm{kbs}(A)$ für alle $A \in \mathsf{GL}_n(\mathrm{K})$.
2. $\mathrm{kbs}(AB) \sqsubseteq \mathrm{kbs}(A) \sqcup \mathrm{kbs}(B)$ für alle $A, B \in \mathrm{K}^{n\times n}$.

Beweis

1. Es gilt $(B \oplus C)^{-1} = B^{-1} \oplus C^{-1}$. Die Blockdiagonalstruktur wird also beim Invertieren nicht gröber. Da die Inversion involutiv ist, wird die Blockstruktur dann auch nicht feiner. Aus Lemma 5.4.1 folgt damit die Behauptung für den

Spezialfall von blockdiagonalen Matrizen. Mit Lemma 5.4.2 folgt der allgemeine Fall durch Umbenennen der Punkte.

2. Nach einer geeigneten Umbenennung der Punkte können die beiden Matrizen A und B als blockdiagonal mit Struktur $\mathrm{kbs}(A) \sqcup \mathrm{kbs}(B)$ aufgefaßt werden (obwohl beide feiner zerlegt sind). Als Produkt kompatibler Blockdiagonalmatrizen ist AB auch blockdiagonal und seine Struktur ist kompatibel zu $\mathrm{kbs}(A)$ und $\mathrm{kbs}(B)$. Daher ist $\mathrm{kbs}(AB)$ durch $\mathrm{kbs}\, A \sqcup \mathrm{kbs}\, B$ beschränkt. ∎

5.2 Erzeugbare Blockstrukturen

Im letzten Abschnitt wurde die konjugierte Blockstruktur definiert. In diesem Abschnitt geht es um ihre Rolle bei der Perm-Irred-Symmetrie. Gesucht sind Mengen von Permutationen L, die durch M simultan und kompatibel in Blöcke zerlegt werden, also

$$M^{-1} \cdot L \cdot M = \text{konjugiert blockdiagonal.}$$

Es liegt daher nahe, die Abbildung $L \mapsto \mathrm{kbs}(M^{-1}LM)$ zu studieren. Die folgende Definition geht noch einen Schritt weiter.

5.6 Definition *Sei $M \in \mathsf{GL}_n(\mathrm{K})$ eine feste Matrix. Die folgende Abbildung ordnet jeder Untergruppe $G \leq \mathsf{S}_n$ ihre Blockstruktur unter M zu.*

$$\mathrm{kbs}_M : \left\{ \begin{array}{ccc} \mathrm{Subgrp}(\mathsf{S}_n) & \to & \mathrm{Part}(\{1..n\}) \\ G & \mapsto & \bigsqcup_{L \in G} \mathrm{kbs}(M^{-1}LM). \end{array} \right.$$

Umgekehrt ordnet die folgende Abbildung jeder Partition p von $\{1..n\}$ die größte Gruppe mit dieser Blockstruktur zu.

$$\mathrm{G}_M : \left\{ \begin{array}{ccc} \mathrm{Part}(\{1..n\}) & \to & \mathrm{Subgrp}(\mathsf{S}_n) \\ p & \mapsto & \left\{ L \in \mathsf{S}_n \;\middle|\; \mathrm{kbs}(M^{-1}LM) \sqsubseteq p \right\}. \end{array} \right.$$

Das nächste Lemma zeigt, daß $\mathrm{G}_M(p)$ wohldefiniert ist und wirklich eine „Gruppe“ ist.

5.7 Lemma $\mathrm{G}_M(p)$ ist eine Gruppe für alle $M \in \mathsf{GL}_n(\mathrm{K})$ und $p \in \mathrm{Part}(\{1..n\})$.

Beweis Die Menge $\mathrm{G}_M(p)$ ist nicht leer, da sie die Identität enthält. Aus Lemma 5.5.1. folgt die Abgeschlossenheit von $\mathrm{G}_M(p)$ unter Inversion, und aus Lemma 5.5.2. folgt die Abgeschlossenheit unter Produktbildung (verwende $p, q \sqsubseteq r \Rightarrow (p \sqcup q) \sqsubseteq r$). Mit dem Untergruppenkriterium folgt die Behauptung. ∎

Mit der Definition 5.6 werden die Abbildungen kbs_M und G_M ins Rampenlicht gestellt. Die Abbildungen verbinden die beiden endlichen Verbände $(\mathrm{Subgrp}(\mathsf{S}_n), \leq)$ und $(\mathrm{Part}(\{1..n\}), \sqsubseteq)$. Es könnte daher vermutet werden, daß die Bildmengen der Abbildungen isomorphe Unterverbände von $\mathrm{Subgrp}(\mathsf{S}_n)$ und $\mathrm{Part}(\{1..n\})$ sind. Diese Vermutung hat sich als falsch herausgestellt! Der folgende Satz faßt zusammen, was gilt, und das Beispiel 5.9 setzt weitergehenden Vermutungen ein Ende.

5.8 Satz *Sei* $M \in \mathsf{GL}_n(\mathrm{K})$. *Dann sind* kbs_M *und* G_M *ordnungserhaltende Abbildungen, die jeweils eine der beiden Verbandsoperationen erhalten und die andere abschätzen. Formal: Für alle Partitionen* p *und* q *von* $\{1..n\}$ *und alle Untergruppen* G *und* H *von* S_n *gilt*

$$\begin{aligned} G \leq H \Rightarrow \mathrm{kbs}_M(G) \sqsubseteq \mathrm{kbs}_M(H), \qquad & p \sqsubseteq q \Rightarrow \mathrm{G}_M(p) \leq \mathrm{G}_M(q), \\ \mathrm{kbs}_M(G \cap H) \sqsubseteq \mathrm{kbs}_M(G) \sqcap \mathrm{kbs}_M(H), \qquad & \mathrm{G}_M(p \sqcap q) = \mathrm{G}_M(p) \cap \mathrm{G}_M(q), \\ \mathrm{kbs}_M(\langle G \cup H \rangle) = \mathrm{kbs}_M(G) \sqcup \mathrm{kbs}_M(H), \qquad & \mathrm{G}_M(p \sqcup q) \geq \langle \mathrm{G}_M(p) \cup \mathrm{G}_M(q) \rangle. \end{aligned}$$

Darüber hinaus sind kbs_M *und* G_M *in folgendem Sinne zu einander invers: Für alle Partitionen* p *und Gruppen* G *gilt*

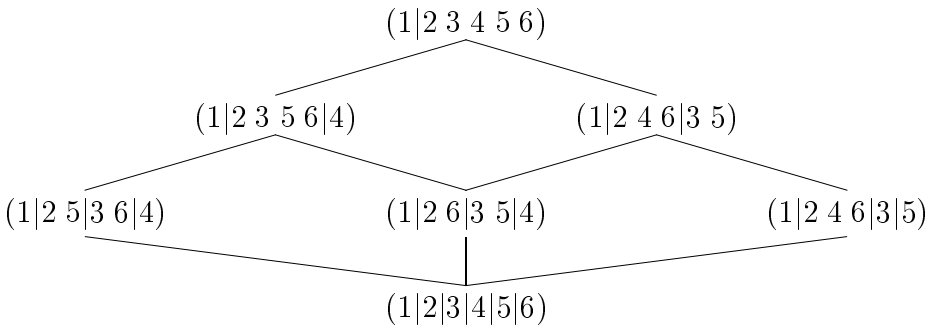
$$\begin{aligned} \mathrm{kbs}_M(\mathrm{G}_M(p)) \sqsubseteq p, \qquad & \mathrm{kbs}_M(\mathrm{G}_M(\ \mathrm{kbs}_M(G)\ )) = \mathrm{kbs}_M(G), \\ \mathrm{G}_M(\mathrm{kbs}_M(G)) \geq G, \qquad & \mathrm{G}_M(\mathrm{kbs}_M(\ \mathrm{G}_M(p)\ )) = \mathrm{G}_M(p). \end{aligned}$$

Beweis Alle genannten Aussagen können direkt nachgerechnet werden. Erforderlich ist lediglich Lemma 5.5 sowie wenige elementare Eigenschaften der Verbandsoperationen. ∎

5.9 Beispiel Sei $M = \mathrm{DFT}_6$ die diskrete (6×6)-Fourier-Transformationsmatrix in $\mathbb{C}$. Sie ist in Abschnitt 7.1 definiert. Die folgende Tabelle enthält die auftretenden Blockstrukturen $p = \mathrm{kbs}(M^{-1}LM)$ für alle $L \in \mathsf{S}_6$. Zur Notation von Gruppen und deren Erzeugendensystemen sei auf Anhang A, Seite 114, verwiesen.

p	$\mathrm{G}_M(p)$	$\lvert \mathrm{G}_M(p) \rvert$	Erzeuger von $\mathrm{G}_M(p)$
$(1\vert 2\ 3\ 4\ 5\ 6)$	S_6	120	$(1\,2), (1\,2\,3\,4\,5\,6)$
$(1\vert 2\ 3\ 5\ 6\vert 4)$	$\mathsf{S}_3 \wr \mathsf{Z}_2$	72	$(1\,2)(3\,4)(5\,6), (1\,3\,5)(2\,6\,4); (4\,6)$
$(1\vert 2\ 4\ 6\vert 3\ 5)$	$\mathsf{Z}_2 \times \mathsf{S}_4$	48	$(1\,4)(2\,5)(3\,6); (1\,3)(4\,6), (1\,2\,4\,5)(3\,6)$
$(1\vert 2\ 5\vert 3\ 6\vert 4)$	$\mathsf{Z}_3 \times \mathsf{S}_3$	36	$(1\,3\,5)(2\,4\,6); (1\,2)(3\,4)(5\,6), (1\,3\,5)(2\,6\,4)$
$(1\vert 2\ 4\ 6\vert 3\vert 5)$	$\mathsf{Z}_2 \times \mathsf{A}_4$	24	$(1\,4)(2\,5)(3\,6); (1\,2\,3)(4\,5\,6), (1\,2\,6)(3\,4\,5)$
$(1\vert 2\ 6\vert 3\ 5\vert 4)$	D_{12}	12	$(1\,5)(2\,4), (1\,2\,3\,4\,5\,6)$
$(1\vert 2\vert 3\vert 4\vert 5\vert 6)$	Z_6	6	$(1\,2\,3\,4\,5\,6)$

Man beachte insbesondere, daß die gröbste auftretende Blockstruktur nicht $\{\{1..6\}\}$ ist, sondern $\{\{1\}, \{2..6\}\}$. Die Tabelle ist durch die Verfeinerungsrelation und die Untergruppenrelation partiell geordnet. Das Hasse-Diagramm dieser partiellen Ordnung:



Darüber hinaus ist auch die partiell geordnete Menge der Partitionen *kein* Unterverband von $\mathrm{Part}(\{1..6\})$. Dazu berechne man den Schnitt

$$(1|2\ 3\ 5\ 6|4) \sqcap (1|2\ 4\ 6|3|5) = (1|2\ 6|3|4|5).$$

Diese Partition ist jedoch nicht realisiert, denn jede Permutation mit dieser Blockstruktur unter M ist bereits in Z_6. Daher ist der Schnitt der Partitionen nur eine obere Schranke für die tatsächliche Verfeinerung $(1|2|3|4|5|6)$.

Die partiell geordnete Menge der Gruppen ist auch *kein* Unterverband von $\mathrm{Subgrp}(\mathsf{S}_6)$. Dazu betrachte man die Vereingung

$$\langle(\mathsf{Z}_3 \times \mathsf{S}_3) \cup \mathsf{D}_{12}\rangle = \mathsf{S}_3 \times \mathsf{S}_3.$$

Diese Gruppe tritt jedoch nicht auf, denn die Permutation $(4\,6)$ besitzt ebenfalls die gleiche Blockstruktur unter M. Daher ist die Vereinigungsgruppe nur eine untere Schranke der tatsächlichen Vereinigung $\mathsf{S}_3 \wr \mathsf{Z}_2$. ∎

5.3 Perm-Block-/Perm-Irred-Symmetrie

Mit den Abbildungen kbs_M und G_M wird das Potential der Matrix M zum Zerlegen in Blöcke untersucht. Es liegt daher nahe, die Gesamtheit der möglichen Zerlegungen auch als Symmetrietyp aufzufassen. Dies geschieht in der folgenden Definition.

5.10 Definition *Die* ***Perm-Block-Symmetrie*** *von* $M \in \mathsf{GL}_n(\mathrm{K})$ *ist die Bildmenge von* G_M*, also*

$$\mathsf{PermBlock}(M) = \Big\{ \mathrm{G}_M(p) \;\Big|\; p \in \mathrm{Part}(\{1..n\}) \Big\}.$$

Zu jeder Gruppe $G \in \mathsf{PermBlock}(M)$ ist $\mathrm{kbs}_M(G)$ die konjugierte Blockstruktur. Außerdem treten alle mit kbs_M erreichbaren Blockstrukturen auch als $\mathrm{kbs}_M(G)$ mit $G \in \mathsf{PermBlock}(M)$ auf. Die Menge $\mathsf{PermBlock}(M)$ erfaßt daher die wesentliche Information aus kbs_M und G_M. Der folgende Satz faßt die Verbandsstruktur der Perm-Block-Symmetrie zusammen.

5.11 Satz *Sei* $M \in \mathsf{GL}_n(\mathrm{K})$. *Dann gilt*

1. $\mathsf{PermBlock}(M)$ *ist bezüglich der Untergruppenrelation* $\leq$ *ein Verband.*

2. $\mathrm{kbs}_M(\mathsf{PermBlock}(M))$, *die Menge der zugehörigen Blockstrukturen, ist bezüglich der Verfeinerungsrelation* $\sqsubseteq$ *ein zu* $(\mathsf{PermBlock}(M), \leq)$ *isomorpher Verband. Die Abbildung* kbs_M *ist ein Verbandsisomorphismus mit der Inversen* G_M.

3. $(\mathsf{PermBlock}(M), \leq)$ *ist im allgemeinen kein Unterverband des Verbands aller Untergruppen der symmetrischen Gruppe* S_n.

Beweis

1./2. Nach Definition ist ein Verband eine partiell geordnete Menge, in der zu je zwei Elementen eine kleinste obere und eine größte untere Schranke existiert. Die Mengen

$$\mathcal{G} = \mathsf{PermBlock}(M) \quad \text{und} \quad \mathcal{P} = \mathrm{kbs}(\mathsf{PermBlock}(M))$$

sind durch $\leq$ bzw. $\sqsubseteq$ partiell geordnet. Nach Satz 5.8 sind die Einschränkungen

$$\mathrm{kbs}_M : \mathcal{G} \to \mathcal{P} \quad \text{und} \quad \mathrm{G}_M : \mathcal{P} \to \mathcal{G}$$

zueinander inverse, ordnungserhaltende Bijektionen. Offensichtlich ist $\mathsf{S}_n \in \mathcal{G}$ maximal und $\{\{1\}, \ldots, \{n\}\} \in \mathcal{P}$ minimal. Da die Mengen $\mathcal{G}$ und $\mathcal{P}$ endlich sind, gibt es offensichtlich zu je zwei Elementen eine kleinste obere und eine größte untere Schranke.

3. Das Beispiel 5.9 der DFT_6 zeigt direkt ein Gegenbeispiel. ∎

Die Perm-Block-Symmetrie erfaßt jede mögliche Zerlegung einer Untergruppe der S_n in Blöcke. Von besonderem Interesse sind jedoch die Zerlegungen, bei denen die Blöcke kleinstmöglich sind. Diese Teilmenge soll einen eigenen Namen erhalten.

5.12 Definition *Die* ***Perm-Irred-Symmetrie*** *von* $M \in \mathsf{GL}_n(\mathrm{K})$ *ist folgende Menge von Gruppen*

$$\mathsf{PermIrred}(M) = \Big\{ G \in \mathsf{PermBlock}(M) \;\Big|\; \textit{Blöcke von } M^{-1}GM \textit{ irreduzibel} \Big\}.$$

In Beispiel 5.9 etwa sind alle Gruppen in $\mathsf{PermBlock}(M)$ sogar in $\mathsf{PermIrred}(M)$. Im Sinne der Darstellungstheorie endlicher Gruppen sind die Elemente $G \in \mathsf{PermIrred}(M)$ genau diejenigen Darstellungen, die durch Konjugation mit M voll ausreduziert werden.

5.4 Bestimmung der Symmetrie

In den vorangegangenen Abschnitten wurde die Strukturtheorie der Perm-Block- und Perm-Irred-Symmetrie entwickelt. Das zugehörige algorithmische Problem ist

5.13 Problem *Gegeben* $M \in \mathsf{GL}_n(\mathrm{K})$. *Bestimme* $\mathsf{PermBlock}(M)$.

Eine einfache aber zuverlässige Lösung ist

5.14 Algorithmus (Robust) Der Algorithmus durchläuft alle Permutationen $L \in \mathsf{S}_n$ und sammelt alle auftretenden Blockstrukturen $\mathrm{kbs}(M^{-1}LM)$ in einer Menge von Gruppen T auf. Am Ende ist $T = \mathsf{PermBlock}(G)$.

$T := \emptyset$;
`for` $L \in \mathsf{S}_n$ `do`
  $p := \mathrm{kbs}(M^{-1} \cdot L \cdot M)$;
  `for` $G \in T$ `do`
    `if` $\mathrm{kbs}_M(G) \sqsupseteq p$ `then`
      erweitere G um L;
    `fi`;
  `od`;
  `if` $p \notin \{\mathrm{kbs}_M(G) \mid G \in T\}$ `then`
    füge die neue Gruppe $\langle L, G \mid G \in T, \mathrm{kbs}_M(G) \sqsubseteq p\rangle$ zu T hinzu;
  `fi`;
`od`. ∎

Der Rest dieses Abschnitts erklärt einen alternativen Lösungsansatz. Die alternative Methode basiert auf der Perm-Mat-Symmetrie, die in Kapitel 4 behandelt wurde: Der vorige Algorithmus verwendet ausschließlich $L \mapsto \mathrm{kbs}(M^{-1}LM)$ um etwas über die Matrix M in Erfahrung zu bringen. Das folgende Lemma ermöglicht nun auch die Auswertung von $\mathrm{G}_M(p)$.

5.15 Lemma Sei $M \in \mathsf{GL}_n(\mathrm{K})$ und p eine Partition der Menge $\{1..n\}$. Dann gilt

$$\mathrm{G}_M(p) = \bigcap_{b \in p} \Pi_L(\mathsf{PermMat}(M_{*,b})).$$

Beweis Nach Definition von PermMat ist $(L, R) \in \mathsf{PermMat}(M_{*,b})$ genau dann, wenn $L \cdot M_{*,b} = M_{*,b} \cdot R$. Dies ist gleichbedeutend damit, daß der von den Spalten aufgespannte Vektorraum $\langle M_{*,j} \mid j \in b\rangle$ invariant ist unter L. Daher ist

$$\Pi_L(\mathsf{PermMat}(M_{*,b})) = \mathrm{G}_M(\{b, \{1..n\} \setminus b\}) \quad \text{für alle } b \subseteq \{1..n\}.$$

Die Behauptung folgt mit $\mathrm{G}_M(p \sqcap q) = \mathrm{G}_M(p) \cap \mathrm{G}_M(q)$ aus Lemma 5.8. ∎

Da M invertierbar ist, ist der Spaltenrang von $M_{*,b}$ für jeden Block $b \subseteq \{1..n\}$ maximal, also $|b|$. Daher kann die Permutationsgruppe $\Pi_L(\mathsf{PermMat}(M_{*,b}))$ mit dem Algorithmus 4.7 von Seite 46 konstruiert werden. Der Schnitt von Permutationsgruppen ist ein Standardverfahren und kann in dem Buch von Butler (1991), [16], nachgelesen werden. Damit kann $p \mapsto \mathrm{G}_M(p)$ berechnet werden. Diese Möglichkeit wird verwendet, um die Perm-Block-Symmetrie zu bestimmen.

5.16 Algorithmus (via Blockstabilisatoren) Gegeben $M \in \mathsf{GL}_n(\mathrm{K})$. Der Algorithmus berechnet $\mathsf{PermBlock}(M)$, indem alle möglichen Blöcke abgespalten werden. Am Ende ist $T = \mathsf{PermBlock}(G)$.

$T := \{\mathsf{S}_n\}$;
`for` $k \in \{1..\lfloor n/2 \rfloor\}$ `do`
  `for` $b \in \binom{\{1..n\}}{k}$ `do`
    $T = T \cup \{\mathsf{PermMat}(M_{*,b})\}$;
  `od`;
`od`. ∎

Auch dieser Algorithmus zur Berechnung von $\mathsf{PermMat}(M)$ hat eine exponentielle Laufzeit, denn es werden 2^{n-1} viele Teilmengen b betrachtet. Trotzdem ist der Algorithmus wesentlich nützlicher als Algorithmus 5.14, denn die für die Algorithmengenerierung nützlichsten Gruppen werden sehr früh gefunden. Dies zeigt der folgende Satz.

5.17 Satz *Wird die äußere Schleife von Algorithmus 5.16 bei einer festen Schranke k_0 abgebrochen, so werden in polynomialer Zeit genau die Elemente von $\mathsf{PermBlock}(M)$ gefunden, die einen Block b der Größe $|b| \leq k_0$ enthalten. Es werden dazu $O(n^{2k_0+1})$ viele Operationen im Grundkörper benötigt.*

Beweis Da der Algorithmus die Elemente von $\mathsf{PermBlock}(M)$ bezüglich ihres kleinsten Blocks aufzählt, werden genau die behaupteten Elemente gefunden. Nun zum Aufwand.

Für festes k_0 ist $\binom{n}{k_0}$ ein Polynom k_0-ten Grades in n. Nach Satz 4.8 benötigt die Berechnung von $\mathsf{PermMat}(M_{*,b})$ höchstens $O(n^{|b|+1})$ viele Operationen im Grundkörper und dies wird großzügig durch $O(n^{k_0+1})$ abgeschätzt. Als Funktion von n ergibt sich für festes k_0 damit der behauptete polynomiale Aufwand. ∎

Bei einer konkreten Implementierung des Suchverfahrens lassen sich viele redundante Berechnungen sparen. Dabei wird die Struktur des Untergruppenverbands und die Struktur des Verbands der Partitionen verwendet. Die beiden wichtigsten Ideen sind

- Es kann gefordert werden, daß die Menge T immer unter Schnittmengenbildung abgeschlossen ist, denn $\mathrm{G}_M(p \sqcap q) = \mathrm{G}_M(p) \cap \mathrm{G}_M(q)$. Beim Hinzufügen einer neuen Gruppe werden dann rekursiv alle Schnittgruppen ebenfalls hinzugefügt, wobei jeweils die kbs neu berechnet werden muß, denn es gilt nur $\mathrm{kbs}_M(G \cap H) \sqsubseteq \mathrm{kbs}_M(G) \sqcap \mathrm{kbs}_M(H)$.

- Zu jeder Gruppe G kann eine untere Abschätzung L_G für die Menge $\{p \mid \mathrm{G}_M(p) = G\}$ gespeichert und gepflegt werden. Diese Menge ist eine Vereinigung von Intervallen der Form $\{p \mid \mathrm{kbs}_M(G) \sqsubseteq p \sqsubseteq q\}$ und kann daher durch die Menge der maximalen Elemente q im Rechner dargestellt werden. Ein abzuspaltender Block b muß in Algorithmus 5.16 nur dann betrachtet werden, wenn er in keinem L_G für ein $G \in T$ liegt. Andernfalls führt b zu keiner neuen Blockstruktur.

Asymptotisch ändern diese beiden Optimierungen nichts an der Laufzeit, denn es müssen im schlechtesten Fall exponentiell viele k-Teilmengen von $\{1..n\}$ betrachtet werden. Es werden jedoch viele langsame Berechnungen, wie PermMat, durch schnellere Berechnungen, wie dem Schnitt zweier Permutationsgruppen oder dem Schnitt zweier Partitionen, ersetzt.

5.5 Zusammenfassung

In diesem Kapitel wurde die Perm-Block- und die Perm-Irred-Symmetrie betrachtet. Bei diesen Symmetrieformen zerlegt die gegebene Matrix M eine Gruppe von Permutationen simultan in Blöcke (Beispiel 5.1).

In Abschnitt 5.1 wird zunächst die konjugierte Blockstruktur, kbs, als quantitatives Maß für Blockstruktur eingeführt (Definition 5.2). Die kbs hat die wichtige Eigenschaft, mit der Matrixmultiplikation und der Matrixinversion verträglich zu sein (Lemma 5.5). Die kbs erfaßt zudem eine Umbenennung der Punkte (Lemma 5.4). Daher ist die Perm-Block-Symmetrie robust gegen „Ausgangspermutationen“ von M. Das bedeutet, $\mathsf{PermBlock}(MS)$ mit einer Permutation S geht aus $\mathsf{PermBlock}(M)$ durch Umbenennen der Punkte hervor. („Eingangspermutationen“ $M \mapsto SM$ werden dadurch erfaßt, daß mit $\mathsf{PermBlock}(M)$ alle Permutationen der S_n betrachtet werden.)

In Abschnitt 5.2 wird untersucht, welche Blockstrukturen von einer festen invertierbaren Matrix M erzeugt werden können. Die dualen Abbildungen kbs_M und G_M ordnen einer Permutationsgruppe eine Blockstruktur und einer Partition eine Permutationsgruppe zu (Definition 5.6, Lemma 5.7). In Satz 5.8 sind die wesentlichen Eigenschaften von kbs_M und G_M zusammengetragen und Beispiel 5.9 zeigt, daß die angegebenen Abschätzungen scharf sind.

In Abschnitt 5.3 wird schließlich die Perm-Block-Symmetrie (Definition 5.10) formal definiert. Die Perm-Block-Symmetrie ist eine partiell geordnete Menge von Permutationsgruppen. Wie die Gruppentabelle aus Beispiel 5.9 illustriert, ist

die Perm-Block-Symmetrie eine recht kompakte Datenstruktur. Sie umfaßt alle Informationen, um die beiden Abbildungen kbs_M und G_M direkt auszuwerten. Das heißt, die Perm-Block-Symmetrie erfaßt *alle* Möglichkeiten der Zerlegung einer Permutationgruppe durch Konjugation mit der Matrix M. Es hängt von der Anwendung ab, welche der Gruppen zur Algorithmengenerierung verwendet werden sollte. Besonders interessante Kandidaten sind die Elemente der Perm-Irred-Symmetrie, denn sie sind feinst möglich zerlegt (Definition 5.12).

Die wesentliche algorithmische Aufgabe dieses Kapitels ist die Bestimmung der Perm-Block-Symmetrie (Problem 5.13). Sie wird in Abschnitt 5.4 behandelt. Es wird zunächst ein direkter und robuster Algorithmus angegeben (Algorithmus 5.14). Da die ganze S_n durchsucht wird, ist der Algorithmus nur etwa bis Grad $n = 8$ erträglich schnell. Einen völlig anderen Lösungsansatz verwendet der Algorithmus 5.16: Er durchsucht nicht S_n und berechnet dann $L \mapsto \mathrm{kbs}(M^{-1}LM)$, sondern er durchsucht die Potenzmenge $2^{\{1..n\}}$ und berechnet dabei $b \mapsto \mathrm{G}_M(\{b, \{1..n\} \setminus b\})$. Letzteres geschieht mit der Perm-Mat-Symmetrie (Lemma 5.15). Der Hauptvorteil des zweiten Algorithmus ist, daß die Zerlegungen in kleine Blöcke sehr schnell gefunden werden, nämlich polynomial in n und der Blockgrößenschranke. Alle vorgestellten Verfahren wurden vom Autor in der Programmiersprache GAP implementiert. Das Programm besteht aus rund 1.8 kloc.

6

Ausdünnen rechteckiger Matrizen

GEGEBEN SEI eine rechteckige Matrix $\boxed{\quad A \quad}$. Konstruiere eine invertierbare Matrix $\boxed{T}$ so, daß $\boxed{T} \cdot \boxed{\quad A \quad}$ maximal viele Nulleinträge besitzt. Die Matrix A soll also durch lineare Zeilenoperationen „ausgedünnt“ werden. Der Konstruktionsprozeß für T soll daher *Ausdünnen* von A heißen. Von diesem handelt das Kapitel. Ein kleines Beispiel möge klarstellen, daß Ausdünnung nicht das gleiche ist wie Gaußelimination:

$$\begin{bmatrix} 1 & 0 \\ \Leftrightarrow 2 & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 2 & 2 \end{bmatrix}}_{\text{bereits in Treppenform}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ \Leftrightarrow 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

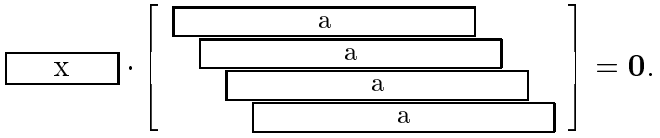
Im Kontext der Symmetriesuche von Matrizen ist Ausdünnen als ein wichtiger Hilfsalgorithmus zu verstehen. Nach meinem Kenntnisstand wurde das Problem des Ausdünnens in dieser Form noch nicht behandelt, Aspekte davon sind in der Literatur jedoch bekannt. Insbesondere greift die Theorie der Matroide. Es war genau die lineare Abhängigkeit, die H. Whitney (1935) in der grundlegenden Arbeit [91] als Beispiel für Matroide betrachtet hat. Lineare Abhängigkeit ist zentral beim Ausdünnen.

In dem speziellen Fall des Grundkörpers $\mathbb{F}_2$ wird das Problem in der Codierungstheorie behandelt. Es firmiert dort unter dem Namen „Bestimmung das Minimalgewichts“: Die Matrix A ist die Generatormatrix eines linearen Codes. Gesucht ist das Minimalgewicht des von A erzeugten Codes, dem von den Zeilen aufgespannten Vektorraum. In dieser Sprechweise ist das Ausdünnen von A der Übergang zu einer Generatormatrix TA des gleichen Code, aber von minimalem Gewicht. Das Minimalgewicht des Codes kann aus TA trivial abgelesen werden. Zudem ist das summierte Gewicht der Zeilen von TA eine Invariante des Codes unter Basiswechsel. Die in diesem Kapitel vorgestellten allgemeinen Algorithmen lösen zwar auch den Codierungstheorie-Fall, sind dort aber nicht sehr erfolgreich. Die in der Codierungstheorie erfolgreichen Verfahren, etwa auf Basis von „Informationsmengen“ wie bei A. Kerber und K.-H. Zimmermann (1996) beschrieben, nutzen die spezielle Situation in der Codierungstheorie geschickt aus.

Der Begriff der „Informationsmenge“ wurde von Prange (1962) zur Dekodierung von Blockcodes eingeführt, [75]. In dieser Arbeit geht es jedoch um reichhaltigere Grundkörper, wie etwa $\mathbb{Q}$. Daher können die kombinatorischen Verfahren der Codierungstheorie nicht verwendet werden (zumal hier mehr als das Minimalgewicht berechnet werden soll). Die Verbindung zur Codierungstheorie ist für die vorliegende Arbeit jedoch dennoch von Interesse. Sie hilft, die Komplexität des Problems zu klären. Dies wird in Abschnitt 6.5 behandelt. Zudem liegt Algorithmus 6.2 eine ähnliche Idee zugrunde, wie den Informationsmengen.

Der dritte aus der Literatur bekannte Aspekt betrifft die in Abschnitt 6.3 beschriebene Methode der Blockstruktur. Dieser von T. Minkwitz (1994) erdachte Ansatz konnte vom Autor auf den Begriff der „Pseudoinversen“ einer Matrix zurückgeführt werden. Da zur Pseudoinversen eine wohlverstandene und etablierte Theorie existiert, konnte die Blockstruktur-Methode systematisch untersucht werden. Insbesondere wurde geklärt, wo sie anwendbar ist und wo ihre Grenzen liegen. Die in Abschnitt 6.3 verwendete Theorie der Pseudoinversen entstammt im wesentlichen dem schönen Buch von A. Ben-Israel und T. N. E. Greville (1974), [7]. Die notwendige Verallgemeinerung auf beliebige Grundkörper bedeutet nur eine geringe Schwierigkeit. Sie wird in Abschnitt 6.4 dargestellt. Zudem konnte die algebraische Struktur der Menge aller Blockzerlegungen einer festen Matrix geklärt werden; sie bilden einen Verband. Dies wird in Abschnitt 6.3 bewiesen.

Auf einen vierten Aspekt des Ausdünnens einer rechteckigen Matrix hat Th. Beth vor kurzem hingewiesen: Ein wesentlicher Algorithmus der Kryptanalyse ist der Algorithmus von Berlekamp-Massey. Er ermöglicht es, zu einer periodischen Bitfolge a ein lineares rückgekoppeltes Schieberegister minimaler Länge mit Ausgabe a zu konstruieren. Der Algorithmus löst auf geschickte Weise ein homogenes lineares Gleichungssystem der Form



$= \mathbf{0}.$

In der Matrix $A \in \mathbb{F}_2^{n \times m}$ stehen also n verschobene Kopien des Zeilenvektors a untereinander (zyklisch verschoben oder mit Null aufgefüllt, je nach Modell). Die Anzahl n der Zeilen ist die Länge des minimalen linearen Schieberegisters, welches a reproduziert. Die Länge n ist bei einem guten Pseudozufallsgenerator logarithmisch klein gegen die Länge der Periode von a. Ist jedoch ein einziges Bit von a nicht korrekt gemessen worden, so wird das Gleichungssystem erst lösbar, wenn n ungefähr die Periode von a erreicht. Dies ist in der Praxis ein großes Problem, denn die Folge a ist meist nur zum Teil bekannt. Wird nun statt eines *Lösungsvektors* des Gleichungssystems nur ein *ausdünnender Vektor* gesucht, so wird das Problem robust gegen kleine Störungen. Für die Kryptanalyse ist die Methode des Ausdünnens wegen der hohen Komplexität zwar unpraktikabel, sie löst aber das richtige Problem: Durch Ausdünnen der Matrix A wird ein lineares

Schieberegister vorgegebener Länge n bestimmt, welches die Folge a unter allen Schieberegistern der Länge n bestmöglich approximiert.

Der restliche Teil dieses Kapitels ist folgendermaßen aufgebaut: In Abschnitt 6.1 wird das Problem formalisiert und dazu werden einige praktische Notationen eingeführt. Der Abschnitt 6.1 stellt danach drei einfache aber wichtige Eigenschaften des Problems vor. Es folgt in Abschnitt 6.2 das erste nicht-triviale Ergebnis: Es wird ein Algorithmus zum Ausdünnen angegeben. Die Korrektheit, Terminierung und der Aufwand dieses Algorithmus werden untersucht.

Die Abschnitte 6.3 bis 6.4 enthalten die Blockzerlegungsmethode, das zweite wesentliche Ergebnis. Nach der Einführung in Abschnitt 6.3 wird die Struktur der Blockzerlegungen aufgeklärt. Danach werden in Abschnitt 6.4 Methoden diskutiert, wie die feinste Blockzerlegung algorithmisch gefunden werden kann. Abschnitt 6.4 schließlich zeigt auf, wie sich die Blockzerlegungsmethode einfügt in andere Ausdünnungsalgorithmen.

In Abschnitt 6.5 wird die Komplexität des Ausdünnens behandelt. Es stellt sich heraus, daß das Problem genauso schwer ist wie ein Standardproblem aus der Codierungstheorie. Von diesem wird vermutet, daß es NP-vollständig ist.

In Abschnitt 6.6 wird ein nicht-triviales Beispiel vorgestellt: Es sollen bessere Erzeuger eines vorgebenen Erweiterungskörpers der rationalen Zahlen konstruiert werden. Ausdünnung liefert dabei eine Alternative zur klassischen Minimierung der Diskriminante. Abschnitt 6.7 schließt das Kapitel mit eine Zusammenfassung der wesentlichen Ergebnisse zum Ausdünnungsproblem.

6.1 Formalisierung und einfache Folgerungen

In diesem Kapitel wird reger Gebrauch gemacht von den in Anhang A definierten Notationen. Insbesondere wird die kompakte Notation $A_{*,J}$ für Untermatrizen einer Matrix A verwendet. Diese wird auf Seite 116 definiert. Darüber hinaus ist es in diesem Kapitel praktisch, lineare Abbildungen durch Rechtsmultiplikation (das heißt $x \mapsto xA$) mit Matrizen zu definieren. Daraus folgt der Kern der Matrix A als $\ker(A) = \{x \in \mathrm{K}^n \mid xA = 0\}$ und das Bild als $\mathrm{K}^n \cdot A$.

Außer den in Anhang A definierten Bezeichnungen werden in diesem Kapitel noch weitere Notationen benötigt. Sei x ein $(1 \times m)$-Vektor, so bezeichnet $\mathrm{supp}(x) = \{k \mid x_k \neq 0\}$ den Träger (engl. support) von x. Das Hamming-Gewicht $|\,x\,| = |\mathrm{supp}(x)|$ von x ist die Größe des Trägers, also die Anzahl der $\neq 0$-Einträge von x. Für einen Vektor $c \in \mathbb{R}^m_{>0}$ von Gewichtskoeffizienten sei außerdem das c-Gewicht $|\,x\,|_c = \sum_{k \in \mathrm{supp}(x)} c_k$ vereinbart. Das c-Gewicht einer Matrix ist die Summe der c-Gewichte ihrer Zeilen. Mit diesen Bezeichnungen kann das in diesem Kapitel zu behandelnde Problem formal präsentiert werden.

6.1 Problem (Ausdünnen einer Matrix) *Gegeben sei die Matrix $A \in \mathrm{K}^{n \times m}$ über dem effektiven Körper* K *und ein Vektor $c \in \mathbb{N}^m_{>0}$ von Gewichtskoeffizienten.*

Berechne eine invertierbare Matrix $T \in \mathsf{GL}_n(\mathrm{K})$, so daß das c-Gewicht von TA minimal ist.

Zur Erinnerung: Ein Körper heißt effektiv, wenn die arithmetischen Operationen und der Vergleich $(x = 0)$ berechenbar sind. Der Vektor c erlaubt es, die einzelnen Spalten von TA unterschiedlich zu gewichten. Diese leichte Verallgemeinerung des Ausdünnungsproblems ist notwendig, um die Spaltenmenge vereinfachen zu können. Im Zweifelsfall denke man sich diese Komplikation einfach weg und stelle sich immer nur das Hamming-Gewicht vor.

Als Prototyp des Problems stelle man sich den Grundkörper $\mathrm{K} = \mathbb{Q}$ vor. Andere Grundkörper sind aber auch von Interesse. Insbesondere die „Körper" der reellen und komplexen Gleitkommazahlen sind nützlich; sie liefern schnelle Approximationen zu unhandlichen algebraischen Strukturen, wie etwa Erweiterungskörpern von $\mathbb{Q}$. Dies gilt umso mehr, als eine Matrix T schnell exakt berechnet werden kann, sobald die Positionen der Nulleinträge in TA bekannt sind. (Dies wird weiter unten gezeigt.) Es kann also durchaus günstig sein, numerisch auszudünnen und anschließend T exakt algebraisch zu rekonstruieren. Doch zunächst werden drei einfache Eigenschaften des Problems besprochen.

Ausdünnen ist ein Raffke-Problem Die erste wesentliche Eigenschaft von Problem 6.1 ist, daß es durch einen Raffke-Algorithmus (engl. greedy algorithm) gelöst werden kann. Diese Erkenntnis ist entscheidend für die später vorgestellte Suchmethode aus Abschnitt 6.2. In gewissem Sinne erlaubt es die Raffke-Struktur nämlich, nach den Zeilen von T unabhängig voneinander zu suchen.

Grundlegend für den Beweis der Raffke-Eigenschaft ist der Begriff des *Matroids*. Dieser geht auf die Arbeit [91] von H. Whitney (1935) zurück. Eine moderne Einführung in die Theorie der endlichen Matroide, sowie ihrer näheren Verwandten findet sich in dem Buch [51] von B. Korte, L. Lovász und R. Schrader (1991). In der hier gegebenen Situation sind aufgrund des unendlichen Grundkörpers einige einfache Ergänzungen zur Theorie notwendig. Die Begriffe und Beweise aus [51], Sects. II.1 und II.2, lassen sich aber sinngemäß anwenden. Zur Erinnerung (nach [51], hier für abzählbare Trägermenge):

6.2 Definition *Das Paar (E, M) mit $\emptyset \in M \subseteq 2^E$ heißt **Matroid**, wenn die folgenden beiden Eigenschaften erfüllt sind:*

- $\forall X \in M : Y \subseteq X \Rightarrow Y \in M$.
- $\forall X, Y \in M$ *mit* $|X| = |Y| + 1 : \exists x \in X \setminus Y : Y \cup \{x\} \in M$.

Nun zur Matroid-Eigenschaft des Ausdünnens.

6.3 Satz *Es seien A und c wie in Problem 6.1 sowie*

$$\begin{array}{rcl} M & = & \{X \subseteq \mathrm{K}^{1\times n} \mid X \text{ ist linear unabhängig}\} \\ w & : & \begin{cases} \mathrm{K}^{1\times n} \to \mathbb{R}_{>0} \\ x \mapsto |\, xA \,|_c. \end{cases} \end{array}$$

Dann ist $(\mathrm{K}^{1\times n}, M)$ ein Matroid und w induziert eine Bewertung des Matroids durch die Fortsetzung

$$w(X) \;=\; \sum_{x\in X} w(x) \quad \textit{für alle } X \in M.$$

Beweis M ist das bekannte Standardmatroid aus den Basen aller Unterräume eines festen, endlich dimensionalen Vektorraums (hier $\mathrm{K}^{1\times n}$). Der Definition des c-Gewichts kann angesehen werden, daß die Bildmenge $w(\mathrm{K}^{1\times n})$ endlich ist. Daher besitzt jede Teilmenge $X \subseteq \mathrm{K}^{1\times n}$ ein w-minimales Element. ∎

Nachdem (M, w) als bewertetes Matroid erkannt ist, folgt aus der Theorie der Matroide sofort, daß der folgende Raffke-Algorithmus ein maximales Element $T \in M$ von minimalem Gewicht $|\, TA \,|_c$ berechnet, d.h. ausdünnt. (Um aus der Teilmenge $T \subseteq \mathrm{K}^{1\times n}$ eine ausdünnende Matrix zu erhalten, schreibe man die Elemente untereinander.)

6.4 Algorithmus (Abstrakt)

$T := \emptyset$;
`while` $|T| < n$ `do`
  $X := \{x \in \mathrm{K}^{1\times n} \setminus T \mid T \cup \{x\} \in M\}$;
  wähle $x \in X$ mit $w(x) = \min\{w(x') \mid x' \in X\}$;
  $T := T \cup \{x\}$
`od`. ∎

Man beachte, daß die Menge X niemals tatsächlich konstruiert wird; es wird immer nur ein Element minimalen Gewichts aus X gewählt. Die Operation ist zwar schwierig, kann aber mit kombinatorischer Suche gelöst werden. Dies wird in Abschnitt 6.2 gezeigt.

T kann rekonstruiert werden Gesetzt den Fall es sind nur die *Positionen* der Nulleinträge einer Lösung bekannt. Das heißt, es sind A, c und Indexmengen $J_1, \ldots, J_n$ gegeben, so daß es ein $T \in \mathsf{GL}_n(\mathrm{K})$ gibt mit $(TA)_{i,j} = 0$ für alle $j \in J_i$ für alle i.

In dieser Situation kann mit linearer Algebra leicht eine passende Matrix T konstruiert werden. Es gilt für alle i

$$(TA)_{i,J_i} = \mathbf{0} \;\Leftrightarrow\; T_{i,*} \in \ker(A_{*,J_i}).$$

(Die Indizierung bezeichnet Teilmatrizen; zur Notation siehe Seite 116.) Die Zeilenvektoren $T_{i,*}$ können also einfach der Reihe nach linear unabhängig aus den Kernen $\ker(A_{*,J_i})$ gewählt werden. Die Matrix T ist dann invertierbar und erzeugt die richtigen Nulleinträge in TA.

Die Tatsache, daß T aus dem Träger der Lösung TA rekonstruiert werden kann, zeigt die *kombinatorische* Natur des Ausdünnungsproblems. Zudem ermöglicht sie es, ausreichend genaue Näherungslösungen nachträglich algebraisch exakt zu rekonstruieren.

Linear abhängigen Spaltenpaare Angenommen in der Matrix A gibt es zwei linear abhängige Spalten A_{*,j_1} und A_{*,j_2}. Dann ist $xA_{*,j_1} = 0$ genau dann, wenn $xA_{*,j_2} = 0$. Zum Ausdünnen ist also eine der beiden Spalten redundant. Das $(n \times m)$-Problem kann auf ein $(n \times (m \Leftrightarrow 1))$-Problem reduziert werden, indem eine der beiden Spalten weggelassen wird und die Gewichtskoeffizienten c entsprechend zusammengefaßt werden.

Wird dieses Verfahren iteriert, so gelangt man schließlich zu einer Matrix A in der keine zwei Spalten linear abhängig sind. Jede der neuen Spalten steht für eine ganze Schar von Spalten der ursprünglichen Matrix und im Vektor c der Gewichtskoeffizienten ist protokolliert, wie groß die Scharen waren. Tatsächlich ist diese Protokollfunktion der Grund, warum die Gewichtskoeffizienten überhaupt in die Theorie eingeführt wurden.

6.2 Ein Suchalgorithmus zur Lösung

In diesem Abschnitt wird ein Algorithmus zum Ausdünnen vorgestellt. Er findet eine Matrix T mit maximal vielen Nullen in TA.

Der Algorithmus betrachtet im wesentlichen alle Möglichkeiten, $n \Leftrightarrow 1$ Nullspalten aus m vielen gegebenen Spalten auszuwählen. Essentiell dafür ist die in Abschnitt 6.1 eingeführte Korrespondenz zwischen Spaltenindexmengen, auf denen xA verschwindet und Vektorräumen, die bestimmte Spalten in A annullieren: Sei $x \in \mathrm{K}^{1\times n}$ ein Vektor, so ist $\{1..m\} \setminus \mathrm{supp}(xA)$ die Menge der Indizes, an denen xA verschwindet. Umgekehrt sei $J \subseteq \{1..m\}$ eine Indexmenge, dann ist $\ker(A_{*,J})$ der Vektorraum aller x, für die xA auf J verschwindet.

Nun zum Algorithmus selbst. Sei $A \in \mathrm{K}^{n\times m}$ und $c \in \mathbb{N}_{>0}^m$. Nach Abschnitt 6.1 darf ohne Einschränkung gefordert werden, daß A keine linear abhängigen Spaltenpaare enthält. Für $m \leq n$ ist das Problem lediglich lineare Algebra; Gauß-Elimination liefert die dünnste Matrix. Sei also außerdem $m > n$ und $\mathrm{rank}(A) = n$. Der Algorithmus zur Konstruktion von T lautet

6.5 Algorithmus (Ausdünnen) Der Algorithmus pflegt eine Agenda S, die $(n \Leftrightarrow 1)$-Teilmengen der Menge $\{1..m\}$ enthält; die Agenda wird abgebaut. Außerdem

wird eine Menge X von Zeilenvektoren gehalten. Die Elemente sind Kandidaten für die Zeilen des Resultats. Die Menge X wird aufgebaut.

$X := \emptyset$;
$S := \binom{\{1..m\}}{n-1}$;
`while` $|S| > 0$ `do`
  wähle $J \in S$;
  `while` $\dim\ker(A_{*,J}) > 1$ `and` $|J| < m$ `do`
    wähle $j \in \{1..m\} \setminus J$;
    $J := J \cup \{j\}$
  `od`;
  berechne eine Basis B von $\ker(A_{*,J})$;
  $X := X \cup B$;
  wähle $x \in B$;
  $S := S \setminus \binom{\{1..m\}\setminus\mathrm{supp}(xA)}{n-1}$ $(*)$
`od`;

Nun ist eine Menge X von Zeilenvektoren bestimmt. Fasse diese Menge als Liste auf und sortiere sie bezüglich des c-Gewichts, d.h. bezüglich ihres Wertes unter $(x \mapsto |\,xA\,|_c)$. Die Elemente kleinen Gewichts sind vorne. Dann wähle aus der Liste eine Basis aus. Dazu gehe von vorne nach hinten durch und nimm den aktuellen Vektor hinzu, falls er linear unabhängig ist. Die so erhaltenen Basisvektoren bilden die Zeilen der ausdünnenden Matrix T. ∎

6.6 Satz *Der Algorithmus 6.5 löst das Problem 6.1 in* $O(\binom{m}{n-1}n^3)$ *Operationen des Grundkörpers.*

Beweis (Durch Analyse von Algorithmus 6.5) Zentral für die Korrektheit und die Terminierung des Algorithmus ist Schritt $(*)$; dort wird die Agenda verkleinert. Dazu muß etwas weiter ausgeholt werden. Seien I und J Spaltenindexmengen und $I \subseteq J$. Dann gilt für die annulierenden Vektorräume

$$\{0\} = \ker(A) \leq \ker(A_{*,J}) \leq \ker(A_{*,I}),$$

denn je größer die auferlegte Einschränkung durch die Spaltenmenge ist, um so kleiner wird der Raum. Da $\ker(A_{*,J})$ mindestens eindimensional ist, solange $|J|$ kleiner als n ist, kann jedes gewählte J solange vergrößert werden, bis der Kern $\ker(A_{*,J})$ *exakt* eindimensional ist. Die Basis B aus dem Algorithmus enthält also immer genau ein Element, dieses wird als x gewählt.

Weiterhin enthält die Menge $\{1..m\}\setminus\mathrm{supp}(xA)$ mindestens das vorher gewählte J. Daher wird in Schritt $(*)$ mindestens J aus S entfernt. Es werden darüber hinaus aber auch noch alle weiteren $(n \Leftrightarrow 1)$-Teilmengen von $\{1..m\}$ entfernt, die zu der gleichen Basis B führen würden. Jedenfalls terminiert der Algorithmus nach höchstens $\binom{m}{n-1}$ Durchläufen der äußeren Schleife.

Jeder der Durchläufe ist mit der Bestimmung von $\dim\ker(A_{*,J})$ verbunden. Diese Aufgabe der linearen Algebra kann in $O(n^3)$ Körperoperationen erledigt werden. Damit ist auch die behauptete Aufwandsangabe bewiesen. ∎

Um das soeben dargestellte Aufwandsergebnis richtig würdigen zu können, mache man sich folgendes klar: Der Binomialkoeffizient $\binom{n}{k}$ ist polynomial, falls nur eine Variable variiert wird. Als Funktion von zwei Variablen ist der Aufwand jedoch exponentiell, denn $\binom{2n}{n}$ geht gegen $2^{2n}/\sqrt{\pi n}$ für große n.

6.3 Die Blockzerlegungsmethode

Dieser und die nachfolgenden Abschnitte handeln von einer speziellen Methode zu Beschleunigung des Ausdünnens. Die Methode basiert auf Blockzerlegungen. Sie wurde ursprünglich von T. Minkwitz (1994) verwendet, um günstigere Basen in abelschen Erweiterungskörpern von $\mathbb{Q}$ zu finden. Unabhängig davon entstand aus meiner Arbeit an der Stewart-Plattform (1993) die Formulierung des Ausdünnungsproblems als abstrakte Aufgabe. Die hier entwickelte Theorie der Blockzerlegungen entstand daraufhin aus der Zusammenarbeit mit T. Minkwitz.

Zur Blockstruktur. Gegeben sei eine dicht besetzte rechteckige Matrix $\boxed{A}$. Man konstruiere eine invertierbare Matrix $\boxed{T}$ und eine Permutationsmatrix P, so daß das Produkt $\boxed{T}\cdot\boxed{A}\cdot P =$  (schematisch) die feinste mögliche Blockzerlegung ist.

Mit der Blockzerlegung kann A wesentlich einfacher ausgedünnt werden, denn die einzelnen Blöcke dürfen unabhängig voneinander ausgedünnt werden. Man beachte jedoch, daß die feinst mögliche Blockzerlegung auch nur aus ganz A bestehen kann. In diesem Fall ist die Blockzerlegungsmethode nicht anwendbar. Es drängen sich folgende Fragen auf:

1. Was ist Blockstruktur? Gibt es eine feinste Blockzerlegung? Ist diese eindeutig bestimmt?

2. Kann die feinste Blockzerlegung konstruktiv berechnet werden? Wie aufwendig ist dies? (Abschnitt 6.4)

3. Wird das Ausdünnungsproblem durch Blockzerlegung wirklich verkleinert? Um wieviel? (Abschnitt 6.4)

Zunächst zur ersten Frage. Eine Matrix A heiße *block-diagonal*, falls sie von der Form $A^{(1)} \oplus \cdots \oplus A^{(b)}$ ist mit $b > 1$. Sie heiße *permutiert block-diagonal*, falls es Permutationsmatrizen P und Q gibt, so daß PAQ block-diagonal ist. Als weitere Abschwächung schließlich heiße A *potentiell block-diagonal*, falls es eine invertierbare Matrix T gibt, für die TA permutiert block-diagonal ist.

Zum Ausdünnen ist es ausreichend, mit den Partitionen der Spaltenindexmenge zu arbeiten. Grundlegend für den Umgang mit Blockstruktur ist dazu die folgende

6.7 Definition *Die* ***Spaltenblockstruktur*** *(engl. column block structure) der Matrix* $A \in \mathrm{K}^{n\times m}$ *ist die Partition*

$$\mathrm{cbs}(A) = \{1, \ldots, m\}/\sim^* .$$

Dabei ist $\sim^*$ *der reflexiv-transitive Abschluß der symmetrischen Relation* $\sim$*, die definiert ist durch* $j_1 \sim j_2 \Leftrightarrow \exists i : A_{i,j_1} \neq 0$ *und* $A_{i,j_2} \neq 0$.

Für einen permutiert block-diagonale Matrix A enthält $\mathrm{cbs}(A)$ genau die Spaltenindizes der Spalten aus den Blöcken. Die $\mathrm{cbs}(A)$ ist also eine Menge von paarweise disjunkten Teilmengen von $\{1..m\}$. Zum Vergleich von cbs, bs (aus Kapitel 2) und kbs (aus Kapitel 5) sei auf Seite 19 der Einleitung verwiesen. Zum Begriff des Verbands und zum Verband der Partitionen einer festen Menge vergleiche Anhang A auf Seite 114. Der nun folgende Hilfssatz zeigt, daß sich die Blockstruktur einer Matrix immer vergröbern läßt.

6.8 Lemma Sei $A \in \mathrm{K}^{n\times m}$ eine Matrix ohne Nullspalten und $p \sqsupseteq \mathrm{cbs}(A)$. Dann gibt es eine invertierbare Matrix T mit $\mathrm{cbs}(TA) = p$.

Beweis Da der Verband aller Partitionen von $\{1..m\}$ endlich ist, genügt es, die Behauptung für die minimalen Vergröberungen von $\mathrm{cbs}(A)$ zu zeigen. Sei also $p \sqsupset \mathrm{cbs}(A)$ minimal. Die minimalen Vergröberungen von $\mathrm{cbs}(A)$ bedeuten, daß genau zwei Blöcke verbunden werden. O.B.d.A. (Umbezeichnen der Indizes) sei daher $A = A^{(1)} \oplus A^{(2)} \oplus A^{(\mathrm{rest})}$, wobei $A^{(1)}$, $A^{(2)}$ die beiden zu verbindenden Blöcke sind und $A^{(\mathrm{rest})}$ der Rest. Alle Nullzeilen von A seien in $A^{(\mathrm{rest})}$. In dieser Situation betrachte man die Matrix

$$T = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} .$$

Offensichtlich ist T invertierbar. Das Produkt mit A hat die Form

$$TA = \begin{bmatrix} A^{(1)} & A^{(2)} & \mathbf{0} \\ \mathbf{0} & A^{(2)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A^{(\mathrm{rest})} \end{bmatrix} .$$

Da $A^{(1)}$ und $A^{(2)}$ weder Nullzeilen noch Nullspalten enthalten, gibt es eine Zeile i und Spalten j_1, j_2 mit $(TA)_{i,j_1} \neq 0$ und $(TA)_{i,j_2} \neq 0$. Aus der reflexiv-transitiven Fortsetzung folgt dann $\mathrm{cbs}(A) = p$. ∎

Mit dieser Vorbereitung kann das Hauptergebnis in diesem Abschnitt formuliert und bewiesen werden.

6.9 Satz (Struktur der cbs**)** *Für jede Matrix* $A \in \mathrm{K}^{n\times m}$ *ist die Menge*

$$\mathcal{L}(A) = \{\mathrm{cbs}(TA) \mid T \in \mathsf{GL}_n(\mathrm{K})\}$$

ein Unterverband des Verbands aller Partitionen von $\{1..m\}$.

Beweis Angenommen A enthält Nullspalten. Dann bildet jede Nullspalte einen eigenen Block in $\mathrm{cbs}(TA)$ für jede invertierbare Matrix T. Das Problem kann dann reduziert werden auf die restlichen Spalten und die sind nicht Null. Sei also ab jetzt A eine Matrix ohne Nullspalten. Angenommen A enthält Nullzeilen. Dann dürfen diese einfach weggelassen werden, denn $\mathrm{cbs}(\cdot)$ ist unter ihnen invariant. Sei A also ab jetzt auch ohne Nullzeilen.

Offensichtlich ist $\mathcal{L}(A) \neq \emptyset$. Nach dem Unterverbandskriterium ist zu zeigen, daß $\mathcal{L}(A)$ abgeschlossen ist unter $\sqcap$ und $\sqcup$, den Verbandsoperationen für Partitionen. Dazu seien $\mathrm{cbs}(T_1A)$ und $\mathrm{cbs}(T_2A)$ Elemente aus $\mathcal{L}(A)$.

Zunächst zu „$\sqcup$“. Die Partition $\mathrm{cbs}(T_1A) \sqcup \mathrm{cbs}(T_2A)$ ist eine Vergröberung von $\mathrm{cbs}(T_1A)$. Nach Lemma 6.8 gibt es daher ein invertierbares T mit

$$\mathrm{cbs}(TT_1A) = \mathrm{cbs}(T_1A) \sqcup \mathrm{cbs}(T_2A).$$

Also ist $\mathcal{L}(A)$ abgeschlossen unter $\sqcup$.

Nun zu „$\sqcap$“. Mit $\{e_i\}_i$ sei die Standardbasis von $\mathrm{K}^{1\times n}$ bezeichnet. Für jedes $T \in \mathsf{GL}_n(\mathrm{K})$ werde nun die Abbildung $f_T : 1..n \to \mathrm{cbs}(TA)$ betrachtet, die jedem i einen Block $f_T(i)$ zuordnet, in dem $\{j \mid (TA)_{i,j} \neq 0\}$ als Teilmenge enthalten ist. Der Block $f_T(i)$ ist genau der Spaltenblock, auf dem Zeile i von TA (das ist e_iTA) nicht verschwindet. Aus der Definition von $\mathrm{cbs}(\cdot)$ folgt, daß f_T wohldefiniert ist, sofern A keine Nullzeilen enthält — und nur dieser Fall wird hier betrachtet.

Den Abbildungen f_T entsprechen Darstellungen des ganzen Vektorraum $\mathrm{K}^{1\times n}$ als direkte Summen. Diese Darstellungen werden gemeinsam verfeinert: Man betrachte $J_1 \in \mathrm{cbs}(T_1A)$ und $J_2 \in \mathrm{cbs}(T_2A)$ sowie die zugehörigen Unterräume von $\mathrm{K}^{1\times n}$

$$V_1(J_1) = \langle e_iT_1 \mid f_{T_1}(i) = J_1\rangle \quad \text{und} \quad V_2(J_2) = \langle e_iT_2 \mid f_{T_2}(i) = J_2\rangle.$$

Dann gilt für alle $x_1 \in V_1(J_1)$ und $x_2 \in V_2(J_2)$ nach Definition von f_T

$$\forall j_1 \notin J_1 : (x_1A)_{j_1} = 0 \quad \text{und} \quad \forall j_2 \notin J_2 : (x_2A)_{j_2} = 0.$$

Folglich gilt für alle $x \in V_1(J_1) \cap V_2(J_2)$ im Schnitt der beiden Räume

$$\forall J \notin J_1 \cap J_2 : (xA)_j = 0.$$

Um zu einer geeigneten Matrix T zu gelangen, konstruiere man daher zu allen $J_1 \in \mathrm{cbs}(T_1A)$ und allen $J_2 \in \mathrm{cbs}(T_2A)$ den Raum $V_1(J_1) \cap V_2(J_2)$ und wähle aus all diesen Räumen eine Basis von $\mathrm{K}^{1\times n}$ aus. (Aus der linearen Algebra ist bekannt, daß dies immer möglich ist.) Die Basisvektoren bilden die Zeilen von T. Mit diesem T gilt

$$\mathrm{cbs}(TA) \sqsubseteq (\mathrm{cbs}(T_1A) \sqcap \mathrm{cbs}(T_2A))$$

und mit Lemma 6.8 kann $\mathrm{cbs}(TA)$ wieder soweit vergröbert werden, daß sogar Gleichheit gilt. Daher ist $\mathcal{L}(A)$ auch unter $\sqcap$ abgeschlossen. ∎

Die Aussage von Satz 6.9 klärt die Frage, ob es eine eindeutige feinste Blockzerlegung gibt: Ja, es gibt sie. Als Unterverband eines endlichen Verbandes hat $\mathcal{L}(A)$ ein eindeutig bestimmtes kleinstes Element. Diese Tatsache erlaubt die folgende Definition:

6.10 Definition *Die **minimale Spaltenblockstruktur** (engl. minimal column block structure) der Matrix $A \in \mathrm{K}^{n\times m}$ ist die Partition*

$$\mathrm{mcbs}(A) = \min\{\mathrm{cbs}(TA) \mid T \in \mathsf{GL}_n(\mathrm{K})\}.$$

Mit dieser Begriffsbildung und Lemma 6.8 sind die Blockzerlegungen einer festen Matrix A vollständig bekannt: Ist $\mathrm{mcbs}(A)$ bekannt, dann kann mit linearer Algebra leicht ein invertierbares T konstruiert werden mit $\mathrm{cbs}(TA) = \mathrm{mcbs}(A)$. Ist dieses T bekannt, so können mit Lemma 6.8 daraus alle anderen von A aus erreichbaren Blockstrukturen durch Vergröberung konstruiert werden.

6.4 Finden der feinsten Blockzerlegung

Dieser Abschnitt behandelt die Frage, wie die feinste Blockzerlegung einer festen Matrix A algorithmisch gefunden werden kann. Es zeigt sich, daß diese Aufgabe mit dem Konzept der Pseudoinversen lösbar ist. Die Pseudoinverse (auch Moore-Penrose-Inverse genannt) einer rechteckigen Matrix ist eine Art verallgemeinerte Inverse der Matrix. Sie erlaubt es, über- oder unterbestimmte lineare Gleichungssysteme systematisch zu lösen — ähnlich der gewöhnlichen Inversen, die eindeutige Gleichungssysteme löst. Die Pseudoinverse hängt zudem mit der Singulärwertzerlegung zusammen; ist die Singulärwertzerlegung bekannt, so kann die Pseudoinverse direkt angegeben werden.

Der nachfolgend dargestellte kleine Ausschnitt aus der Theorie der Pseudoinversen stammt im wesentlichen aus dem Klassiker von A. Ben-Israel und T. N. E. Greville (1974), [7], sowie aus dem Buch [79] von C. R. Rao und S. K. Mitra (1971). Da die Theorie der Pseudoinversen dort jedoch nur über dem Grundkörper $\mathbb{C}$ entwickelt wird, war es notwendig, die Resultate geringfügig zu verallgemeinern. Die Erweiterung der Theorie hat sich im Nachhinein als glücklich erwiesen; durch sie wurde klar, woran die Blockzerlegung hängt.

In diesem Abschnitt sei der Grundkörper K mit einem festen Automorphismus $\overline{(\cdot)} : \mathrm{K} \to \mathrm{K}$ der Ordnung zwei ausgestattet. (Als Prototyp stelle man sich $\mathrm{K} = \mathbb{C}$ und $\overline{(\cdot)}$ als komplexe Konjugation vor.) Die Involution $\overline{(\cdot)}$ des Grundkörpers induziert eine Konjugation der Matrizen über K; namentlich ist die konjugierte Matrix

$$A^* = \left[\overline{A_{ji}} \mid i, j\right] \quad \text{für } A \in \mathrm{K}^{n\times m}.$$

Die Notation A^* bezeichnet in diesem Abschnitt also eine beliebige aber feste Konjugation auf den Matrizen $\mathrm{K}^{n\times m}$.

Die folgende Definition aus dem Buch von Ben-Israel und Greville definiert 16 Begriffe einer verallgemeinerten Inversen; für jede Teilmenge I einen.

6.11 Definition *Die Matrix $X \in \mathrm{K}^{m\times n}$ heißt I-**Inverse** der Matrix $A \in \mathrm{K}^{n\times m}$ bezüglich der Konjugation $(\cdot)^*$, falls X alle Gleichungen in der Menge $I \subseteq \{(1),\dots,(4)\}$ erfüllt. Die möglichen Gleichungen sind*

$$AXA = A \tag{1}$$
$$XAX = X \tag{2}$$
$$(AX)^* = AX \tag{3}$$
$$(XA)^* = XA. \tag{4}$$

Die Menge aller I-Inversen von A werde mit $A\{I\}$ bezeichnet. Die $\{1,2,3,4\}$-Inverse von A ist eindeutig bestimmt, falls sie existiert (Lemma 6.12, Nr. 6). Sie soll die Pseudoinverse *heißen und mit A^+ bezeichnet werden.*

Die vorige Definition ist etwas ungewöhnlich, deshalb hier eine kurze Erklärung: Es gibt vier Eigenschaften, die eine Matrix X erfüllen kann. Für jede Auswahl I aus den vier Gleichungen wird ein Begriff von verallgemeinerten Inversen definiert. Zum Beispiel sind $A\{1\} = \{X \mid AXA = X\}$ und $A\{1,3\} = \{X \mid AXA = X, (AX)^* = AX\}$. Obwohl manche der 16 Begriffe trivial sind, hat sich die Definition als sehr praktisch herausgestellt. Namentlich kann das folgende Lemma sehr kompakt formuliert werden. Es klärt die Existenz und Eindeutigkeit der relevanten verallgemeinerten Inversen.

6.12 Lemma Man betrachte die Matrix $A \in \mathrm{K}^{n\times m}$.

1. $A\{I\} = \{A^{-1}\}$ für alle $\{1\} \subseteq I \subseteq \{1..4\}$, falls $n = m = \operatorname{rank}(A)$.
2. $A\{1\} = \{A^{(1)} + Z \Leftrightarrow A^{(1)}AZAA^{(1)} \mid Z \in \mathrm{K}^{m\times n}\}$, wobei $A^{(1)}$ eine spezielle $\{1\}$-Inverse von A ist. Jedes A besitzt eine $\{1\}$-Inverse.
3. $A\{1,2\} = \{YAZ \mid Y, Z \in A\{1\}\}$. Jedes A besitzt eine $\{1,2\}$-Inverse.
4. $A\{1,3\} = \{A^{(1,3)} + (\mathbf{1}_m \Leftrightarrow A^{(1,3)}A)Z \mid Z \in \mathrm{K}^{m\times n}\}$, falls es überhaupt eine $\{1,3\}$-Inverse $A^{(1,3)}$ von A gibt. Dies ist genau dann der Fall, wenn $\operatorname{rank}(A^*A) = \operatorname{rank}(A)$. Gibt es eine $\{1,3\}$-Inverse, dann ist $(A^*A)^{(1)}A^* \in A\{1,3\}$ für jede $\{1\}$-Inverse $(A^*A)^{(1)}$ von A^*A.
5. $A\{1,4\} = \{A^{(1,4)} + Z(\mathbf{1}_n \Leftrightarrow AA^{(1,4)}) \mid Z \in \mathrm{K}^{m\times n}\}$, falls es überhaupt eine $\{1,4\}$-Inverse $A^{(1,4)}$ von A gibt. Dies ist der Fall genau dann, wenn $\operatorname{rank}(AA^*) = \operatorname{rank}(A)$. Gibt es eine $\{1,4\}$-Inverse, dann ist $A^*(AA^*)^{(1)} \in A\{1,4\}$ für jede $\{1\}$-Inverse $(AA^*)^{(1)}$ von AA^*.
6. $A\{1,2,3,4\} = \{ZAY \mid Y \in A\{1,3\}, Z \in A\{1,4\}\}$. Die $\{1,2,3,4\}$-Inverse ist eindeutig, falls sie überhaupt existiert. Sie existiert genau dann, wenn $\operatorname{rank}(A^*A) = \operatorname{rank}(AA^*) = \operatorname{rank}(A)$.

7. Im Spezialfall $\mathbb{Q} \leq \mathrm{K} \leq \mathbb{C}$ und $\overline{(\cdot)}$ als komplexe Konjugation gilt $\mathrm{rank}(A^*A) = \mathrm{rank}(AA^*) = \mathrm{rank}(A)$ für jede Matrix A.

Beweis Alle Teilbeweise finden sich in [7] für Grundkörper $\mathbb{C}$. Die Beweise sind jedoch wörtlich gültig über anderen Grundkörpern mit einer Konjugation auf den Matrizen. Im einzelnen: 1. trivial, 2. (§2.1-Co.1, §1.2-Th.1), 3. (§1.5-Le.3), 4. (§2.2, §1.3-Le.1(a), §1.6-Th.3), 5. analog 4., 6. (§1.1-Ex.1, §1.6-Th.4), 7. (§1.6-Le.4). ∎

Zur Konstruktion der verallgemeinerten Inversen geben Ben-Israel, Greville und Rao, Mitra verschiedene numerische und algebraische Methoden an ([7], §7.1–7.3 oder [79], §11.4). Insbesondere eignet sich die Singulärwertzerlegung, um die Pseudoinverse A^+ numerisch zu berechnen. Die Singulärwertzerlegung ist ein numerisches Verfahren, das zum Beispiel in den NUMERICAL RECIPES, [76] Ch. 2.6, von W. H. Press et al. (1992) beschrieben wird.

Aus Lemma 6.12 folgt eine einfache Methode zur Konstruktion der Pseudoinversen, falls sie existiert. Dazu wird der hier relevante Spezialfall betrachtet. Sei $A \in \mathrm{K}^{n\times m}$ mit $m > n$ und $\mathrm{rank}(AA^*) = \mathrm{rank}(A) = n$. Aus Lemma 6.12.1 folgt, daß $(AA^*)^{-1}$ die Pseudoinverse von AA^* ist. Nach Lemma 6.12.5 ist dann $X = A^*(AA^*)^{-1}$ eine $\{1,4\}$-Inverse von A. Wird X in (2) und (3) eingesetzt, so zeigt sich, daß X auch eine $\{2,3\}$-Inverse ist. Nach Lemma 6.12.6 ist X daher die eindeutig bestimmte Pseudoinverse

$$A^+ = A^*(AA^*)^{-1} \quad \text{falls } \det(AA^*) \neq 0.$$

Algorithmisch läßt sich A^+ also durch Inversion einer $(n\times n)$-Matrix und Multiplikation mit einer $(m\times n)$-Matrix berechnen. Der Aufwand dafür ist in $O(n^3+n^2m)$. Man beachte dabei jedoch, daß die Bedingung $\det(AA^*) \neq 0$ eine Einschränkung bedeutet. Falls AA^* jedoch nicht invertierbar ist, dann existiert A^+ nach Lemma 6.12.6 nicht.

Nun zum Hauptergebnis in diesem Abschnitt.

6.13 Satz (Bestimmung der mcbs**)** *Sei $A \in \mathrm{K}^{n\times m}$ mit $m > n$ und die Determinante* $\det(AA^*) \neq 0$. *Dann existiert die Pseudoinverse A^+ und es gilt*

$$\mathrm{mcbs}(A) = \mathrm{cbs}(A^+A).$$

Beweis Es wird gezeigt, daß jede Blockzerlegung von TA in $\mathrm{cbs}(A^+A)$ auftaucht — insbesondere also auch die feinste. Der Beweis basiert auf den folgenden beiden Eigenschaften der Pseudoinversen. Sei A wie im Satz, dann gilt

1. $(TAS)^+ = S^{-1}A^+T^{-1}$ für jedes $T \in \mathsf{GL}_n(\mathrm{K})$ und $S \in \mathsf{S}_m$. Bew.: Durch Nachrechnen wird gezeigt, daß die rechte Seite eine $\{1,2,4\}$-Inverse ist. Wegen der Rangbedingung an AA^* ist $A^+ = A^*(AA^*)^{-1}$, wie oben ausgeführt wurde. Damit wird Bedingung (3) trivial.

2. $A^+ = A^{(1)+} \oplus \cdots \oplus A^{(r)+}$, falls A von der Form $A^{(1)} \oplus \cdots \oplus A^{(r)}$ mit $r \geq 1$ ist. Bew.: Es kann direkt nachgerechnet werden, daß die rechte Seite eine $\{1,2,3,4\}$-Inverse von A ist. Die Gleichheit folgt aus der Eindeutigkeit von A^+ (Lemma 6.12.6.).

Aus der ersten Eigenschaft folgt sofort

$$\mathrm{cbs}((TA)^+(TA)) = \mathrm{cbs}(A^+A) \quad \text{für alle } T \in \mathsf{GL}_n(\mathrm{K}).$$

Das heißt, $\mathrm{cbs}(A^+A)$ hängt nicht T ab. Daher kann einfach angenommen werden, daß A schon durch ein geeignetes T in die minimale Blockstruktur zerlegt ist. Die erste Eigenschaft zeigt außerdem

$$\mathrm{cbs}((AS)^+(AS)) = \mathrm{cbs}(S^{-1}A^+AS) = \mathrm{cbs}(A^+AS) \quad \text{für alle } S \in \mathsf{S}_m,$$

S bezeichnet also die Spalten in $\mathrm{cbs}(A^+A)$ auf die gleiche Weise um wie in A. Daher kann zusätzlich angenommen werden, daß A bereits blockdiagonal ist, etwa $A = A^{(1)} \oplus \cdots \oplus A^{(r)}$. Aus der zweiten Eigenschaft folgt dann

$$\mathrm{cbs}(A^+A) = \mathrm{cbs}(A^{(1)+}A^{(1)} \oplus \cdots \oplus A^{(r)+}A^{(r)}),$$

und dies liefert genau die Behauptung. ∎

Satz 6.13 und die Algorithmen zur Konstruktion der Pseudoinversen zeigen, daß es einen Algorithmus zur Bestimmung der minimalen Spaltenblockstruktur gibt, falls die Rangbedingung $\det(AA^*) \neq 0$ erfüllt ist. Für $A \in \mathrm{K}^{n\times m}$ werden $O(n^3 + n^2m)$ arithmetische Operationen in K benötigt.

Es sei noch angemerkt, daß die Pseudoinverse nicht unbedingt notwendig ist zur Bestimmung der minimalen Spaltenblockstruktur — Gaußelimination ist ausreichend. Dies wird in dem folgenden Lemma gezeigt. Die Eigenschaften der Pseudoinverse liefern jedoch den tieferen Grund für die Struktur der minimalen Blockstruktur. Darüber hinaus ist die Pseudoinverse unter praktischen Gesichtspunkten numerisch günstiger als Gaußelimination, da die Pseudoinverse keine Rechenfehler ansammelt.

6.14 Lemma Sei $A \in \mathrm{K}^{n\times m}$ mit $m > n$ und $\det(AA^*) \neq 0$. Dann gilt

$$\mathrm{mcbs}(A) = \mathrm{cbs}(\mathrm{Echelon}(A)),$$

wobei $\mathrm{Echelon}(A)$ die Zeilennormalform (Echelonform, Hermite-Normalform) von A bezüglich einer festen Wähl von Pivot-Spalten bezeichnet.

Beweis Es ist

$$\begin{aligned} \mathrm{cbs}(\mathrm{Echelon}(A)) &= \mathrm{cbs}(\mathrm{Echelon}(A^+A)) \\ &= \mathrm{cbs}(A^+A) \\ &= \mathrm{mcbs}(A). \end{aligned}$$

Die erste Gleichung basiert darauf, daß die Zeilennormalform bezüglich einer festen Wahl der Pivot-Spalten eine Invariante der aufgespannten Vektorräume ist. Die zweite Gleichung gilt, weil A^+A bereits in der feinst möglichen Blockstruktur vorliegt. Die dritte Gleichung kommt aus Satz 6.13. ∎

Ausdünnen mit Blockzerlegungen Die letzten beiden Abschnitte haben gezeigt, daß $\mathrm{mcbs}(A)$ die relevante Information über alle möglichen Blockzerlegungen der Matrix A enthält und daß diese Information schnell algorithmisch beschafft werden kann. An dieser Stelle ist es angebracht, klar zu zeigen, daß die Blockstruktur das Ausdünnen drastisch vereinfacht.

Dazu werden zwei Ausdünnungsprobleme $(A^{(1)}, c^{(1)})$ und $(A^{(2)}, c^{(2)})$ mit Matrizen $A^{(k)}$ und Vektoren von Gewichtskoeffizienten $c^{(k)}$ betrachtet. Mit der Suchmethode aus Algorithmus 6.5 ist es möglich, für jedes Probleme eine ausdünnende Matrix $T^{(k)}$ zu finden, also $|\, T^{(k)}A^{(k)} \,|_{c^{(k)}} = \min$. Nun werden die beiden Probleme zusammengesetzt zu einem größeren. Es geht dabei um die Matrix $A = A^{(1)} \oplus A^{(2)}$ mit dem Vektor von Gewichtskoeffizienten $c = [c^{(1)}, c^{(2)}]$. Offensichtlich ist dann

$$|\, (T^{(1)} \oplus T^{(2)}) \cdot A \,|_c = \min,$$

denn die beiden diagonalen Teile der Matrix sind minimal. Das bedeutet, daß sich auch die Lösung blockdiagonal zusammensetzen läßt! Ist umgekehrt (A, c) ein Problem mit Blockstruktur, so dürfen die Blöcke unabhängig voneinander ausgedünnt werden und aus diesen Teillösungen kann sofort eine Lösung des Gesamtproblems konstruiert werden. Die Blockzerlegungsmethode ergänzt also die Suchmethode. Dies kann substantiell schneller gehen als die Suchmethode alleine:

6.15 Satz *Falls* $\det(AA^*) \neq 0$, *dann kann Problem 6.1 in*

$$O(n^3 + n^2 m + \sum_{k=1}^{r} \tbinom{m_k}{n_k - 1} n_k^3)$$

arithmetischen Operationen gelöst werden. Dabei ist (n, m) *die Größe der Matrix* A *(Anzahl Zeilen, Anzahl Spalten) und* $(n_1, m_1), \ldots, (n_r, m_r)$ *sind die Größen der Blöcke in der minimalen Blockzerlegung von* A.

6.16 Beispiel Die folgende Matrix A kann bereits mit der Blockzerlegungsmethode allein ausgedünnt werden:

$$A = \begin{bmatrix} \Leftrightarrow 2 & \Leftrightarrow 3 & 2 & \Leftrightarrow 3 \\ 1 & 9 & 6 & \Leftrightarrow 9 \\ \Leftrightarrow 1 & \Leftrightarrow 3 & \Leftrightarrow 2 & 3 \end{bmatrix} \Rightarrow A^+ = \frac{1}{156} \begin{bmatrix} 0 & \Leftrightarrow 78 & \Leftrightarrow 234 \\ \Leftrightarrow 26 & 39 & 91 \\ 12 & \Leftrightarrow 6 & \Leftrightarrow 30 \\ \Leftrightarrow 18 & 9 & 45 \end{bmatrix}.$$

Damit wird A^+A blockdiagonal mit

$$\mathrm{mcbs}(A) = \mathrm{cbs}(A^+A) = \{\{1\},\{2\},\{3,4\}\}.$$

Durch Auswahl der 1., 2. und 3. Zeile von A^+ wird ein T gefunden, das A blockdiagonalisiert zu

$$TA = \frac{1}{13}\begin{bmatrix} 13 & 0 & 0 & 0 \\ 0 & 13 & 0 & 0 \\ 0 & 0 & 4 & \Leftrightarrow 6 \end{bmatrix}.$$

Dies ist offensichtlich ausgedünnt. ∎

6.5 Komplexität des Ausdünnungsproblems

Dieser Abschnitt handelt von der Problemkomplexität des Ausdünnens. Es gibt starke Hinweise dafür, daß das Problem NP-vollständig ist. Dies konnte jedoch nicht bewiesen werden. Immerhin wird die Ausdünnung aber im folgenden zurückgeführt auf ein bekanntes offenes Problem. Es handelt sich dabei um das von E. R. Berlekamp, R. J. McEliece und H. C. A. van Tilborg 1978 formulierte Problem des Minimalgewichts eines linearen Blockcodes. Meines Wissens ist von diesem Problem bis heute nicht entschieden worden, ob es NP-vollständig ist.

Zunächst werden präzise Definitionen der auftretenden Probleme gegeben. Schwierigkeiten mit der Arithmetik im Grundkörper sollen hier nicht von Interesse sein. Daher wird nur der Grundkörper $\mathrm{K} = Q$ oder ein endlicher Körper $\mathrm{K} = \mathbb{F}_q$ betrachtet.

[T?] BERECHNE AUSGEDÜNNTE MATRIX Gegeben $A \in \mathbb{Q}^{n\times m}$, $m > n$ und ein Vektor $c \in \mathbb{Q}^m_{>0}$. Berechne $T \in \mathsf{GL}_n(\mathbb{Q})$ mit $|\,TA\,|_c = \min$.

[$V \le w$] VEKTOR MINIMALEN GEWICHTS Gegeben $\{a_1,\ldots,a_n\} \subseteq \mathbb{Q}^m$, $m > n$ und ein w mit $m \ge w > 0$. Gibt es $x \in \langle a_1,\ldots,a_n\rangle \setminus \{0\}$ mit $|\,x\,| \le w$?

[$V = w$] VEKTOR VORGEGEBENEN GEWICHTS Gegeben $\{a_1,\ldots,a_n\} \subseteq \mathbb{Q}^m$, $m > n$ und ein w mit $m \ge w > 0$. Gibt es $x \in \langle a_1,\ldots,a_n\rangle$ mit $|\,x\,| = w$?

[$P \le w$] PUNKT MINIMALEN GEWICHTS Gegeben $\{a_0,\ldots,a_n\} \subseteq \mathbb{Q}^m$, $m > n$ und ein w mit $m \ge w > 0$. Gibt es $x \in a_0 + \langle a_1,\ldots,a_n\rangle \setminus \{a_0\}$ mit $|\,x\,| \le w$?

[$U = w$] GEWICHT EINES UNTERRAUMS Gegeben eine Matrix $A \in \mathbb{Q}^{n\times m}$ und eine ganze Zahl $w > 0$. Gibt es $x \in \mathbb{Q}^n$ mit $|\,x\,| = w$ und $x \cdot A = 0$?

[$X \le w$] LGS-LÖSUNG VON MINIMALEM GEWICHT Gegeben eine endliche Menge X von Paaren (x,b), wobei $x \in \mathbb{Q}^m$, $b \in \mathbb{Q}$, $m > 0$ und eine ganze Zahl $m \ge K > 0$. Gibt es $y \in \mathbb{Q}^m$, so daß $|\,y\,| \le K$ und für alle $(x,b) \in X$ gilt $y^{\mathsf{T}} \cdot x = b$?

Das folgende Lemma sammelt die relevanten Aussagen.

6.17 Lemma Es gilt

1. Rationale lineare Algebra ist P-time.
2. $[X \leq w]$ ist NP-vollständig.
3. $[U = w]$ ist NP-vollständig.
4. $[P \leq w]$ ist NP-vollständig und mindestens so schwer wie $[V \leq w]$.
5. $[V = w]$ ist NP-vollständig und mindestens so schwer wie $[V \leq w]$.
6. $[T?]$ ist mindestens so schwer wie $[V \leq w]$.

Beweis

1. Es soll gezeigt werden, daß die *Bitkomplexität* vom Gleichungslösen über $\mathbb{Q}$ polynomial ist. Dazu sei $xA = b$ ein Gleichungssystem mit $A \in \mathsf{GL}_n(\mathbb{Q})$ und $b \in \mathbb{Q}^n$, $n > 0$. Außerdem sei $2^B \geq \max |A_{ij}|, |b_j|$ eine Schranke für die auftretenden Zahlen. Nach der cramerschen Regel kann die Lösung ausgedrückt werden als $x = [y_1/d, \ldots, y_n/d]$, wobei y_k und d jeweils $(n \times n)$-Determinanten in den Komponenten von A und b sind. In der expliziten Determinantenformel werden $n!$ Produkte mit jeweils n Faktoren aufsummiert. Damit ergibt sich die Schranke $|y_k|, |b| \leq n! 2^{Bn}$. Mit der Stirling-Approximation der Fakultät zeigt sich schließlich, daß die Bitlänge des Resultats in $O(Bn^2 \log n)$ liegt. Es kann auch polynomial berechnet werden, nämlich mit Hilfe des chinesischen Restesatzes durch Rekonstruktion aus hinreichend vielen modularen Gleichungssystemen. (Gaußelimination ist nicht bit-polynomial beschränkt, da die Zwischenergebnisse zu stark wachsen.)

2. Bekannt. Das $[X \leq w]$ ist das Problem [MP5] aus dem Buch [35] von M. R. Garey und D. S. Johnson (1979).

3. Der Beweis für Grundkörper $\mathbb{F}_2$ von Berlekamp, McEliece und van Tilborg (1978) aus [8] ist über $\mathbb{Q}$ wortwörtlich gültig.

4. Reduziere $[X \leq w]$ auf $[P \leq w]$: Sei (X, K) eine Instanz von $[X \leq w]$. Dann kann die Lösungsaffinität des Gleichungssystems in polynomialer Zeit in der Form $a_0 + \langle a_1, \ldots, a_n \rangle$ dargestellt werden (Teil 1. dieses Lemmas). Das Restproblem ist genau $[P \leq w]$; eine Lösung davon liefert auch eine Lösung von $[X \leq w]$. Daher ist $[P \leq w]$ mindestens so schwierig wie $[X \leq w]$. Außerdem ist $[P \leq w]$ in NP, weil eine mögliche Lösung in polynomialer Zeit geprüft werden kann.

Reduziere $[V \leq x]$ auf $[P \leq x]$: Man wähle $a_0 = 0$.

5. Reduziere $[U = w]$ auf $[V = w]$: Sei (A, w) eine Instanz von $[U = w]$. Dann kann wieder in polynomialer Zeit eine Basis $\{a_1, \ldots, a_n\}$ für den durch $xA = 0$ definierten Unterraum berechnet werden. Es bleibt das Problem $[V = w]$.

Reduziere $[V \leq w]$ auf $[V = w]$: Sei $(a_1, \ldots, a_n, w)$ eine Instanz von $[V \leq w]$. Dann kann durch sukzessives Testen von $w' = 1, 2, \ldots, w$ mit einem Algorithmus für $[V = w']$ entschieden werden, ob es $[V \leq w]$ eine Lösung hat.

6. Sei $(\{a_1, \ldots, a_n\}, w)$ eine Instanz von $[V \leq w]$. Aus den Zeilenvektoren a_i bilde man die Matrix $A \in \mathbb{Q}^{n \times m}$. Als Gewichtsvektor wird $c = [1, \ldots, 1]$ verwendet. Mit Hilfe eines Algorithmus für $[T?]$ berechne man ein ausdünnendes $T \in \mathsf{GL}_n(\mathbb{Q})$. Da das Ausdünnungsproblem die Raffke-Eigenschaft (engl. greedy) besitzt, enthält TA eine Zeile mit minimalem Gewicht. Mit dieser Zeile kann das Entscheidungsproblem $[V \leq w]$ beantwortet werden. ∎

Zur Erklärung. $[V \leq w]$ ist das Entscheidungsproblem, das der Ausdünnung zugrunde liegt. Es ist, abgesehen vom Grundkörper $\mathbb{Q}$ statt $\mathbb{F}_2$, genau das Problem „Bestimme das Minimalgewicht eines linearen Blockcodes“. Dieses Problem wurde von Berlekamp, McEliece und van Tilborg 1978 in [8] formuliert. Es ist bis heute nicht bewiesen worden, daß diese Problem NP-vollständig ist.

Das Problem $[V \leq w]$ ist etwas leichter als die zwei verwandten Probleme $[V = w]$ und $[P \leq w]$. Beide Probleme sind bereits NP-vollständig, wie in Lemma 6.17.4. und Lemma 6.17.5. gezeigt wird. Die weiteren Aussagen von Lemma 6.17 dienen als Ausgangsbasis für diese Aussagen.

6.6 Anwendung: Teilkörper von $\mathbb{Q}(\zeta_n)$

Ein schönes Beispiel für den Nutzen der Ausdünnung stammt von T. Minkwitz (1994). Um die irreduziblen Darstellungen einer endlichen Gruppe zu konstruieren, hat Minkwitz einfachere Erzeuger für abelsche Erweiterungen von $\mathbb{Q}$ gesucht. Dazu hat er zunächst Blockzerlegungen verwendet. Mit der Zeit wurde klar, daß der Kern des Problems das Ausdünnen der Generatormatrix ist. Dies soll im folgenden näher ausgeführt werden.

Bei Berechnungen mit Charakteren von endlichen Gruppen tritt oft das Problem auf, einen Erweiterungskörper von $\mathbb{Q}$ zu konstruieren, der eine Menge von vorgegebenen Elementen enthält. Es ist klar, daß es sich dabei um einen Teilkörper eines Kreisteilungskörpers handelt. Obwohl es einfach ist, irgendeine Darstellung für diesen Körper zu konstruieren, ist es sehr schwer, eine gute zu finden. Ausdünnung liefert dazu einen neuen Ansatz.

Man betrachte den festen Kreisteilungskörper $\mathbb{Q}(\omega)$, wobei ω eine primitive 156. Einheitswurzel ist. Dieser Körper ist vom Grad $m = [Q(\omega) : \mathbb{Q}] = \varphi(156) = 48$ über den rationalen Zahlen. Von Interesse ist nun der Körper $\mathbb{Q}(\alpha)$ mit dem Erzeuger

$$\begin{aligned} \alpha \;=\; & \Leftrightarrow\!\omega^{47} \Leftrightarrow\!\omega^{45} + \omega^{41} + \omega^{40} + \omega^{39} + \omega^{38} \Leftrightarrow\! 2\omega^{35} \Leftrightarrow\!\omega^{33} + 2\omega^{29} + \\ & 2\omega^{27} + \omega^{26} \Leftrightarrow\!\omega^{22} + \omega^{19} + \omega^{16} \Leftrightarrow\!\omega^{13} \Leftrightarrow\!\omega^{12} \Leftrightarrow\!\omega^{11} \Leftrightarrow\!\omega^{10} + \\ & \omega^{7} + \omega^{5} + \omega^{4} \Leftrightarrow\! 2\omega \Leftrightarrow\! 1. \end{aligned}$$

Er erzeugt die Normalbasis $B = \{\alpha^g \mid g \in \mathsf{Gal}(\mathbb{Q}(\alpha)/\mathbb{Q})\}$ der Dimension $n = [\mathbb{Q}(\alpha) : \mathbb{Q}] = 8$. Die Galois-Gruppe der Erweiterung ist $\mathsf{Gal}(\mathbb{Q}(\alpha)/\mathbb{Q}) \cong \mathsf{Z}_2 \times \mathsf{Z}_4$. Das Minimalpolynom von α über $\mathbb{Q}$ ist

$$\begin{aligned} f &= X^8 + 2X^7 + 22X^6 + 2X^5 + 93X^4 + \\ &\quad 124X^3 + 1108X^2 + 3352X + 5788. \end{aligned}$$

Auch die Standardbasis eignet sich zum Ausdünnen; dabei wachsen jedoch im allgemeinen die auftretenden rationalen Zahlen wesentlich stärker als bei der Normalbasis.

Es soll nun ein Erzeuger β für $\mathbb{Q}(\alpha)$ gefunden werden, der ein "schöneres" Minimalpolynom über $\mathbb{Q}$ besitzt. Dazu wird die (8×48)-Koeffizientenmatrix A der Normalbasis B ausgedünnt. (Jedes Element α^g wird als Polynom in ω aufgefaßt und der Koeffizientenvektor in eine Zeile von A geschrieben.) Nach 14 Sekunden Rechenzeit eines MATHEMATICA-Programms zum Ausdünnen auf einem Rechner vom Typ SUN SPARC IPX wurde eine invertierbare Matrix T und eine Permutationsmatrix S bestimmt, so daß

$$TAS = \begin{bmatrix}
*\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots*\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots*******\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots\ldots\ldots*********\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots*.*...*.*.**..*.***. \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots*..**...**..*.**... \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots**.*..*....*...*** \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots*...***..*..*.*.*
\end{bmatrix}.$$

Hier steht $*$ für einen Wert ungleich Null. Jede Zeile von TAS repräsentiert ein Element von $\mathbb{Q}(\alpha)$, namentlich

$$\begin{aligned}
\beta_1 &= 1 \\
\beta_2 &= \omega^{26} \\
\beta_3 &= \omega^{40} + \omega^{38} \Leftrightarrow \omega^{22} + \omega^{16} \Leftrightarrow \omega^{12} \Leftrightarrow \omega^{10} + \omega^4 \\
\beta_4 &= \omega^{46} + \omega^{34} \Leftrightarrow \omega^{28} + \omega^{24} \Leftrightarrow \omega^{20} \Leftrightarrow \omega^{18} \Leftrightarrow \omega^8 \Leftrightarrow \omega^6 + \omega^2 \\
\beta_5 &= \Leftrightarrow \omega^{47} \Leftrightarrow \omega^{41} \Leftrightarrow \omega^{37} \Leftrightarrow \omega^{31} + \omega^{21} \Leftrightarrow \omega^{19} + \omega^{15} + \omega^7 \\
\beta_6 &= \Leftrightarrow \omega^{47} + \omega^{45} \Leftrightarrow \omega^{41} \Leftrightarrow \omega^{33} \Leftrightarrow \omega^{19} \Leftrightarrow \omega^{11} + \omega^7 \Leftrightarrow \omega^5 \\
\beta_7 &= \omega^{39} + 2\omega^{37} \Leftrightarrow 2\omega^{33} + 2\omega^{27} \Leftrightarrow 2\omega^{21} \Leftrightarrow 2\omega^{11} \Leftrightarrow 2\omega^9 + 2\omega^3 \\
\beta_8 &= \Leftrightarrow 2\omega^{45} + 2\omega^{41} + \omega^{39} \Leftrightarrow 2\omega^{35} + 2\omega^{29} + 2\omega^{27} + 2\omega^{19} \Leftrightarrow \omega^{13} + \\
&\quad 2\omega^5 \Leftrightarrow 2\omega.
\end{aligned}$$

Die Minimalpolynome dieser Elemente, skaliert und normiert auf ganzzahlige

Koeffizienten, lauten

$$\begin{aligned} f_1 &= X \Leftrightarrow 1 \\ f_2 &= X^2 \Leftrightarrow X + 1 \\ f_3 &= X^4 \Leftrightarrow X^3 + 4X^2 + 3X + 9 \\ f_4 &= X^2 + X \Leftrightarrow 3 \\ f_5 &= X^8 + 13X^6 + 156X^4 + 169X^2 + 169 \\ f_6 &= X^8 + 13X^6 + 156X^4 + 169X^2 + 169 \\ f_7 &= X^4 \Leftrightarrow 26X^2 + 117 \\ f_8 &= X^8 + 26X^6 + 559X^4 + 3042X^2 + 13689. \end{aligned}$$

Jedes dieser Polynome vom Grad $[\mathbb{Q}(\alpha) : \mathbb{Q}] = 8$ kann als Minimalpolynom der Erweiterung gewählt werden. Das Ergebnis der Ausdünnung lautet also: $\mathbb{Q}(\alpha) = \mathbb{Q}(\beta_5)$ mit dem Minimalpolynom $f_5 = X^8 + 13X^6 + 156X^4 + 169X^2 + 169$, was offensichtlich viel schöner ist das das ursprüngliche f für α.

Das Ergebnis kann verglichen werden mit dem in das KANT-System[1] eingebauten Verfahren. Die Funktion `BetterPolynomial` des Computeralgebrasystems MAGMA[2], das einen relevanten Teil von KANT enthält, liefert nach 36 Sekunden das Minimalpolynom

$$\begin{aligned} g &= X^8 \Leftrightarrow 4X^7 + 23X^6 \Leftrightarrow 94X^5 + \\ &\quad 331X^4 \Leftrightarrow 614X^3 + 998X^2 \Leftrightarrow 1148X + 508. \end{aligned}$$

Das Polynom g hat eine kleinere Diskriminante als f und sogar als f_5. Allerdings ist es dicht besetzt und die auftretenden Zahlen sind wesentlich größer als bei f_5. Das zur Minimierung verwendete Kriterium der Diskriminante ist daher nicht notwendigerweise das Maß der Wahl. Geringes Hamming-Gewicht ist es vermutlich ebenfalls nicht, kann aber als Alternative nützlich sein.

6.7 Zusammenfassung

In diesem Kapitel wurde das Problem des „Ausdünnens einer rechteckigen Matrix“ behandelt. Die Aufgabe wurde zunächst formalisiert (Problem 6.1). Es soll eine rechteckige Matrix durch invertierbare Zeilenoperationen so umgeformt werden, daß sie möglichst dünn besetzt ist, im Sinne eines leicht verallgemeinerten Hamming-Gewichts (c-Gewicht).

Es wurde gezeigt, daß Ausdünnen ein Raffke-Algorithmus (engl. greedy algorithm) ist (Satz 6.3). Darauf aufbauend wurde ein Suchverfahren angegeben

[1] KANT wurde an der Universität Düsseldorf entwickelt, [31].

[2] MAGMA wird von der Arbeitsgruppe um J. Cannon in Sydney entwickelt und kommerziell vertrieben.

(Algorithmus 6.5), mit dem eine Matrix $A \in \mathrm{K}^{n\times m}$, $m > n$, in $O(\binom{m}{n-1}n^3)$ arithmetischen Operationen ausgedünnt werden kann (Satz 6.6). Dieser Aufwand ist im allgemeinen exponentiell, denn $\binom{2n}{n} \to 2^{2n}/\sqrt{\pi n}$ für große n.

Das Ausdünnen kann substantiell beschleunigt werden, wenn die Matrix eine Blockstruktur erzeugen kann. Dazu wurde der Begriff der Spaltenblockstruktur einer rechteckigen Matrix formal definiert (Definition 6.7), und es wurde gezeigt, daß die möglichen Spaltenblockstrukturen einer festen Matrix einen endlichen Verband bilden (Satz 6.9). Daher kann jeder Matrix A eine eindeutig bestimmte minimale Spaltenblockstruktur zugeordnet werden (Definition 6.10). Die minimale Spaltenblockstruktur kann mit Hilfe der Pseudoinversen (Definition 6.11) in $O(n^3+n^2m)$ arithmetischen Operationen gefunden werden, falls die Pseudoinverse überhaupt existiert (Satz 6.13). Es wurde gezeigt, daß dies für $\det(AA^*) \neq 0$ immer der Fall ist (Lemma 6.12.5. und 6.). Durch die Blockstruktur wurde der Aufwand zum Ausdünnen im allgemeinen erheblich reduziert (Satz 6.15). Die Methode der Blockstruktur wurde ursprünglich von T. Minkwitz (1994) vorgeschlagen.

Es konnte gezeigt werden, daß das zum Ausdünnen gehörende Entscheidungsproblem genauso schwer ist wie das bekannte Problem [Minimalgewicht eines linearen Blockcodes] aus der Theorie fehlerkorrigierender Codes (Lemma 6.17.6.). Von diesem Problem wird vermutet, daß es NP-vollständig ist (Berlekamp, McEliece, van Tilborg, 1978). Das Problem ist sehr nahe verwandt mit zwei etwas schwierigeren Problemen, welche beide bereits NP-vollständig sind (Lemma 6.17.4. und 5.).

Alle dargestellten Algorithmen wurden vom Autor zusammen mit Torsten Minkwitz implementiert. Die Programme bestehen aus etwa 3.5 kloc MATHEMATICA und 1.0 kloc MAGMA.

Teil II

Anwendungen

7

Klassische Methoden der FFT

```
     ∇ Z←TF33 A;K;M;W;O;P;Q;R;S;V;N
[1]  W← 2 1 ∘.○(,⍳(2,P)ρ(Pρ V←0),-O-V[-O-2×⍳P])÷P,0ρO←⍳1,0ρS←2,N,0ρR←
     (M+1)ρ2,0ρZ←A[;V←,(Φ⍳M)⍉((K←M←1+2⊕P←0.5×N)ρ2)ρ⍳N←¯1↑ρA]
[2]  →(0<K←K-1)/2,0ρW←W[;,⍉(2,P)ρ⍳P]+0,0ρZ←Sρ(-/[O] W×Z),+/[O] W×⊖Z←S
     ρ((O+K),((-K)Φ0,Mρ1)/⍳M+1)⍉Rρ(,+/[K+O] Z),,-/[K+O] Z←RρZ
     ∇
```

G. K. McAuliffe, 2^e-FFT in APL, aus [67]

In diesem Kapitel sind die klassischen Methoden der schnellen diskreten Fourier-Transformation zusammengetragen. Die Algorithmik der diskreten Fourier-Transformation lebt von der Symmetrie; mit ihr können alle schnellen Verfahren, außer der Methode von Bluestein, gefunden werden. Die klassischen Algorithmen für die effiziente Auswertung der diskreten Fourier-Transformation zeigen die große Vielfalt an nutzbarer Struktur.

Da jede nicht-triviale endliche Gruppe eine zyklische Untergruppe enthält, ist die klassische diskrete Fourier-Transformation zudem ein Grundalgorithmus für viele fortgeschrittene Verfahren in der konstruktiven Darstellungstheorie. Ein Beispiel neueren Datums ist die konstruktive Berechnung von Charaktertafeln von p-Gruppen durch A. Thümmel [88]. Sie macht wesentlichen Gebrauch von der diskreten schnellen Fourier-Transformation.

Im Rahmen dieser Arbeit bildet die diskrete Fouriertransformation ein umfassendes Beispiel für symmetriebasierte Algorithmengenerierung. Dieses Kapitel ist zudem eine elementare und explizite Einführung in die klassischen Algorithmen und zeigt deren Schönheit. Es werden sämtliche Permutationen, Twiddle-Faktoren und Symmetrien explizit in geschlossener Form durch Terme von Matrizen angegeben werden. Die hier gegebene Darstellung ergänzt in diesem Sinne die ausführlicheren Darstellungen von Beth (1984), [10], sowie von Clausen und Baum (1993), [19]. In diesem Kapitel wird die schnelle Fourier-Transformation ausschließlich unter dem klassischen algorithmischen Aspekt behandelt.

Der Rest dieses Kapitels ist wie folgt aufgebaut: Abschnitt 7.1 gibt die Definition der zentralen Begriffe dieses Kapitels an: die DFT und die FFT. In Abschnitt 7.2 wird die Symmetrie der DFT angegeben und bewiesen. Es gibt eine Symmetrie

vom Typ Perm-Perm und eine vom Typ Perm-Irred.

In Abschnitt 7.3, werden die einzelnen FFT-Methoden vorgestellt. Da jede der Methoden lediglich einen einzelnen Zerlegungsschritt eines Teile-und-Herrsche-Algorithmus realisiert, ist es notwendig, die Methoden zusammenzusetzen. Dieses Thema ist in der Praxis ein nicht unerhebliches Problem; es wird in Abschnitt 7.4 theoretisch beprochen. In Abschnitt 7.5 wird dann von einer tatsächlichen Implementierung der schnellen Fourier-Transformation in der Programmiersprache C berichtet. Abschnitt 7.6 beendet das Kapitel über die klassischen Methoden der schnellen Fourier-Transformation. Für die verwendeten Notationen sei an dieser Stelle auf den Anhang A ab Seite 113 verwiesen. Abweichend von Anhang A wird in diesem Kapitel $S(i)$ statt i^S für das Bild eines Punktes i unter einer Permutation S geschrieben. *In diesem Kapitel bietet es sich zudem an, Matrizen grundsätzlich mit* $\{0, \ldots, n \Leftrightarrow 1\}$ *statt mit* $\{1, \ldots, n\}$ *zu indizieren.*

7.1 DFT und FFT

Unter der *diskreten Fourier-Transformation* der Größe $n \geq 1$ wird in diesem Kapitel die folgende Matrix verstanden:

$$\mathrm{DFT}_n = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2n-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2n-2} & \omega^{3n-3} & \cdots & \omega^{(n-1)^2} \end{bmatrix} = \left[\omega^{i \cdot j}\right]_{i,j \in [0..n)}.$$

Die Notation $[0..n)$ bezeichnet die Folge oder Menge $\{0, 1, \ldots, n \Leftrightarrow 1\}$, wie in Anhang A auf Seite 113 bezeichnet. Mit ω wird eine primitive n-te Einheitswurzel des zugrundeliegenden Körpers bezeichnet. Im Fall der DFT über den komplexen Zahlen $\mathbb{C}$ stelle man sich einfach $\omega = \exp(2\pi\,\mathrm{i}\,/n)$ vor. Der Kreisteilungskörper $\mathbb{Q}(\zeta_n) \cong \mathbb{Q}[X]/(\Phi_n(X))$, mit dem n-ten Kreisteilungspolynom $\Phi_n(X)$, enthält nach Konstruktion die primitive n-te Einheitswurzel ζ_n. In endlichen Körpern $\mathbb{F}_{p^e}$ schließlich gibt es genau dann eine primitive n-te Einheitswurzel, wenn n ein Teiler von $p^e \Leftrightarrow 1$ ist. (Die multiplikative Gruppe $(\mathbb{F}_{p^e} \setminus \{0\}, \cdot)$ ist zyklisch von der Ordnung $p^e \Leftrightarrow 1$; sie sei von α erzeugt. Dann ist $\alpha^{(p^e-1)/n}$ eine primitive n-te Einheitswurzel.) Bei jedem Grundkörper ist $\{\omega^k \mid k \in \{1..n\},\ \mathrm{ggT}(k, n) = 1\}$ die Menge aller primitiven n-ten Einheitswurzeln, wenn ω eine ist.

Ein Algorithmus zur Auswertung der Abbildung $\mathbf{x} \mapsto \mathrm{DFT}_n\,\mathbf{x}$ wird als *schnelle Fourier-Transformation* (engl. fast fourier transform, FFT) bezeichnet, auch wenn der Algorithmus nicht schnell ist. In diesem Kapitel wird von Methoden berichtet, mit denen „schnelle FFTs“ konstruiert werden können.[1]

[1] Diese paradoxe Begriffsbildung folgt dem „red herring principle“ (engl. red herring = Ablenkungsmanöver): Ein roter Hering braucht in der Mathematik weder rot zu sein, noch muß es sich überhaupt um einen Hering handeln.

Eine fundamentale und interessante Eigenschaft der DFT ist, daß sie nahezu ihre eigene Inverse ist. In der hier gewählten Konvention gilt

$$\mathrm{DFT}_n^{-1} \;=\; \frac{1}{n} S \cdot \mathrm{DFT}_n = \frac{1}{n}\,\mathrm{DFT}_n \cdot S,$$

wobei die Permutation S von $[0..n)$ definiert ist durch

$$S(k) \;=\; \begin{cases} 0 & \text{für } k = 0 \text{ und} \\ n \Leftrightarrow k & \text{für } 0 < k < n. \end{cases}$$

Hieraus folgt insbesondere $S^2 = \mathrm{id}$ und $\mathrm{DFT}_n^2 = nS$, sowie $\mathrm{DFT}_n^4 = n^2 \mathbf{1}_n$. Die DFT ist also eine Matrix der Ordnung vier, bis auf einen skalaren Faktor, und die inverse DFT ist im wesentlichen die DFT selbst.

Eine weitere besondere Eigenschaft der DFT ist der enge Zusammenhang mit der zyklischen Faltung. Als diskretes Analogon der Faltungseigenschaft der kontinuierlichen Fourier-Transformation kann im diskreten Fall die *zyklische* Faltung auf drei DFTs zurückgeführt werden. Es gilt für den Faltungskern $\mathbf{v} = [v_0, \ldots, v_{n-1}]^T$

$$\mathrm{circ}\,(\mathbf{v}) \;=\; \mathrm{DFT}_n \cdot \mathrm{diag}\,(\mathrm{DFT}_n \cdot \mathbf{v}) \cdot \mathrm{DFT}_n^{-1}.$$

Man beachte, daß die mittlere der drei DFTs die Koeffizienten der Diagonalmatrix bestimmt; bei der Auswertung der zyklischen Faltung $\mathbf{x} \mapsto \mathrm{circ}(\mathbf{v}) \cdot \mathbf{x}$ müssen daher nur die beiden äußeren DFTs berechnet werden. Eine zyklische Faltung der Länge n ist also im wesentlichen äquivalent zu *zwei* DFT_n.

7.2 Symmetrien der DFT

Die DFT_n zerlegt eine reguläre Darstellung der zyklischen Gruppe Z_n in ihre irreduziblen Komponenten. Diese Eigenschaft ist eine *Perm-Irred-Symmetrie*. Der folgende Satz präzisiert dies.

7.1 Satz *Sei $n > 0$ und ω die zur Definition von DFT_n verwendete primitive n-te Einheitswurzel im Körper* K. *Man betrachte die zyklische Untergruppe G von $\mathsf{S}_n \times \mathsf{GL}_n(\mathrm{K})$, die definiert ist als*

$$G = \langle \Big((0, 1, \ldots, n \Leftrightarrow 1),\ \mathrm{diag}(1, \omega, \omega^2, \ldots, \omega^{n-1}) \Big) \rangle \cong \mathsf{Z}_n.$$

Dann gilt

$$L \cdot \mathrm{DFT}_n = \mathrm{DFT}_n \cdot R \quad \textit{für alle } (L, R) \in G.$$

Beweis Es reicht, die Gleichung für den Erzeuger von G nachzuprüfen. Nach Formel (A.1) auf Seite 115 ist

$$L \cdot \mathrm{DFT}_n = [\omega^{L(i)\cdot j} \mid i, j] = [\omega^{(i+1)\cdot j} \mid i, j] = [\omega^{ij+j} \mid i, j] = \mathrm{DFT}_n \cdot R.$$

Damit ist die Behauptung gezeigt. ∎

Die zweite wesentliche Symmetrie der DFT_n ist ihre *Perm-Perm-Symmetrie*. Sie entspricht genau der Automorphismengruppe der Z_n.

7.2 Satz *Sei $n > 0$ und $Z_n^\times = \{k \in \{1..n\} \mid \mathrm{ggT}(n,k) = 1\}$. Für zwei Permutationen L und R von $[0..n)$ gilt*

$$L \cdot \mathrm{DFT}_n = \mathrm{DFT}_n \cdot R \;\Leftrightarrow\; \textit{Es gibt } k \in Z_n^\times \textit{ mit } L = R^{-1} = (i \mapsto (k \cdot i) \bmod n).$$

Mit anderen Worten,

$$\mathsf{PermPerm}(\mathrm{DFT}_n) = \{(L, L^{-1}) \mid L = (i \mapsto ki),\ k \in Z_n^\times\}.$$

Beweis Zunächst wird die Perm-Perm-Symmetrie der DFT durch ein System von Kongruenzen charakterisiert. Dazu seien L und R Permutationen von $[0..n)$. Nach Formel (A.1) ist die Komponentendarstellung gegeben durch

$$L \cdot \mathrm{DFT}_n \cdot R^{-1} \;=\; \left[\omega^{L(i)\cdot R(j)} \;\middle|\; i,j \in [0..n)\right].$$

Komponentenvergleich mit DFT_n zeigt

$$L \cdot \mathrm{DFT}_n = \mathrm{DFT}_n \cdot R \;\Leftrightarrow\; \forall i,j \in [0..n) : L(i)\,R(j) \equiv i\,j \bmod n.$$

Seien nun L und R Permutationen mit $L \cdot \mathrm{DFT}_n = \mathrm{DFT}_n \cdot R$ und zur Abkürzung sei $k = L(1)$. Dann folgt aus den Kongruenzen $kR(1) \equiv 1$, also ist k modulo n invertierbar. Das heißt, $k \in Z_n^\times$. Durch k sind die Permutationen L und R eindeutig bestimmt, denn $L(i) \equiv ki$ und $kR(j) \equiv j$ für alle i und alle j. ∎

7.3 Klassische Zerlegungsmethoden

In diesem Abschnitt werden die klassischen Zerlegungsmethoden der DFT vorgestellt. Bei jeder der Methoden wird eine DFT von N Einzelsignalen auf mehrere DFTs von anderer Größe zurückgeführt. Werden diese Zerlegungsschritte sinnvoll zusammengesetzt, so ergibt sich strukturell ein Teile-und-Herrsche-Algorithmus (engl. divide and conquer). Jede einzelne Zerlegungsmethode wird hier kurz als „FFT-Methode“ bezeichnet. Man mache sich jedoch klar, daß jede der FFT-Methoden nur einen einzigen Schritt einer vollständigen Zerlegung der DFT vollzieht.

Die Methode von Good und Thomas Die erste hier zu behandelnde FFT-Methode ist die effizienteste bekannte Zerlegungsmethode einer diskreten Fourier-Transformation. Die Methode wird unter anderem von R. C. Agarwal und J. W. Cooley (1977) in [2] beschrieben. In der ursprünglichen Arbeit [37] beschreibt I. J. Good (1971), wann und wie eine DFT der Länge N auf eine multidimensionale DFT zurückgeführt werden kann.

Bei einer zweidimensionalen DFT werden zuerst alle Zeilen unabhängig voneinander mit einer 1d-FFT transformiert. Danach werden alle neuen Spalten transformiert. In algebraischer Sprechweise handelt es sich bei einer multidimensionalen DFT um das Tensorprodukt der DFTs in den einzelnen Dimensionen. Sind nun die einzelnen Dimensionen paarweise teilerfremd, so sind die Tensorfaktoren trennbar, wie es in dem nachstehenden Lemma ausgedrückt wird. Dieser Fall wird in der Literatur auch oft Mixed-radix-FFT genannt. Das Lemma formuliert den allgemeinen Fall mit beliebig vielen Faktoren.

7.3 Lemma Sei $N = q_1 \cdots q_r$ für $r > 1$ und die Zahlen $q_1, \ldots, q_r > 1$ seien paarweise teilerfremd. Dann gibt es Permutationen L und R mit

$$\mathrm{DFT}_N \quad = \quad L \cdot (\mathrm{DFT}_{q_1} \otimes \cdots \otimes \mathrm{DFT}_{q_r}) \cdot R.$$

Die Permutationen L und R sind eindeutig bestimmt bis auf die Perm-Perm-Symmetrie der DFT_N. Ein spezielles Paar L, R ist definiert durch

$$\begin{aligned} L(k) &= \sum_{\alpha=1}^{r} (k \bmod q_\alpha) \prod_{\beta>\alpha} q_\beta \quad \text{und} \\ R(k) &= \left(\sum_{\alpha=1}^{r} k_\alpha \prod_{\beta\neq\alpha} q_\beta \right) \bmod N \quad \text{für } k \in [0..N). \end{aligned}$$

Dabei bezeichnet $k_\alpha \in [0..q_\alpha)$ die α-Stelle in der Zahldarstellung von k bezüglich der gemischten Basis $q_1, \ldots, q_r$, also

$$k = \sum_{\alpha=1}^{r} k_\alpha \prod_{\beta>\alpha} q_\beta \quad \text{bzw.} \quad k_\alpha = \left(k \operatorname{div} \prod_{\beta>\alpha} q_\beta \right) \bmod q_\alpha.$$

Beweis Der Beweis macht wesentlichen Gebrauch vom Chinesischen Restesatz (CR), hier $x \equiv_N y \Leftrightarrow \forall \alpha : x \equiv_{q_\alpha} y$. Seien L und R Permutationen von $[0..N)$, und die Indizes i und j seien bezüglich der gemischen Basis $q_1, \ldots, q_r$ dargestellt durch $i_\alpha, j_\alpha \in [0..q_\alpha)$ für $\alpha \in [1..r]$. Dann gilt für die Komponentendarstellungen

$$\begin{aligned} L^{-1} \cdot \mathrm{DFT}_N \cdot R^{-1} &= \left[\omega^{L^{-1}(i)\cdot R(j)} \mid i, j \in [0..N) \right] \quad \text{und} \\ \mathrm{DFT}_{q_1} \otimes \cdots \otimes \mathrm{DFT}_{q_r} &= \left[\prod_\alpha \omega_\alpha^{i_\alpha} \mid i, j \in [0..N) \right], \end{aligned}$$

wobei $\omega_\alpha = \omega^{\prod_{\beta\neq\alpha} q_\beta}$ die in DFT_{q_α} verwendete primitive q_α-te Einheitswurzel ist. Der Komponentenvergleich der beiden Matrizen liefert

$$\forall i, j : L^{-1}(i) \cdot R(j) \equiv_N \left(\sum_\alpha i_\alpha \prod_{\beta\neq\alpha} q_\beta \right) \left(\sum_{\alpha'} j_{\alpha'} \prod_{\beta'\neq\alpha'} q_{\beta'} \right).$$

Das letzte Produkt kann ausmultipliziert werden, wobei alle gemischten Terme mit $\alpha' \neq \alpha$ entfallen, da sie N als Faktor enthalten. Wir erhalten die folgende Charakterisierung

$$\begin{aligned} & \mathrm{DFT}_N = L \cdot (\mathrm{DFT}_{q_1} \otimes \cdots \otimes \mathrm{DFT}_{q_r}) \cdot R \\ \Leftrightarrow \quad & \forall i,j : L^{-1}(i) \cdot R(j) \equiv_N \sum_\alpha i_\alpha j_\alpha \prod_{\beta \neq \alpha} q_\beta. \end{aligned}$$

Da L und R nur bis auf die Perm-Perm-Symmetrie der DFT eindeutig bestimmt sind, nehmen wir an, es gäbe eine Paar L, R mit $L(N \Leftrightarrow 1) = N \Leftrightarrow 1$. Die Kongruenzen bestimmen L und R dann eindeutig in der im Lemma angegebenen Weise.

Es bleibt zu zeigen, daß die so erhaltenen Bedingungen auch tatsächlich Permutationen definieren, für die die charakteristischen Kongruenzen gelten. Seien L und R also definiert wie im Lemma. Dann sind L und R wegen dem CR wohldefinierte Permutationen. Für Indizes i, j sei $k = L^{-1}(i)$, also $L(k) = i$ und nach Definition von L ist daher $i_\alpha = (k \bmod q_\alpha)$. Werden L und R eingesetzt in die charakteristische Bedingung, so findet man

$$k \cdot \left(\sum_\alpha j_\alpha \prod_{\beta \neq \alpha} q_\beta \right) \quad \equiv_N \quad \sum_\alpha (k \bmod q_\alpha) j_\alpha \prod_{\beta \neq \alpha} q_\beta,$$

was mit dem CR leicht eingesehen werden kann. ∎

Die Methode von Cooley und Tukey Die Good-Thomas-FFT ist nicht anwendbar, wenn N eine Primzahlpotenz ist und daher nicht teilerfremd zerfällt. Die Methode nach Cooley und Tukey hat dieses Problem nicht. Mit ihr kann jede DFT_N auf mehrere DFTs von Teilern von N zurückgeführt werden. Der Preis dafür besteht in weiteren Multiplikationen mit den sogenannten *Twiddle-Faktoren.*

Obwohl die Zweierpotenz-FFT schon seit langem bekannt war, war es die Arbeit [22] von J. W. Cooley und J. W. Tukey (1965), die eine Art Goldrausch nach schnellen Algorithmen für die DFT ausgelöst hat. In der nur fünf Seiten langen Veröffentlichung stellen die Autoren die allgemeine Zerlegungsmethode dar und geben den Algorithmus für die Zweierpotenz-FFT an. Das folgende Lemma formuliert das allgemeine Zerlegungsprinzip der Cooley-Tukey-FFT.

7.4 Lemma Sei $N = n \cdot m$ mit $n, m > 1$ und ω bezeichne eine primitive N-te Einheitswurzel. Dann gilt

$$\mathrm{DFT}_N \quad = \quad L \cdot (\mathbf{1}_n \otimes \mathrm{DFT}_m) \cdot \mathrm{diag}\left(\omega^{t_k} \mid k \in [0..N)\right) \cdot (\mathrm{DFT}_n \otimes \mathbf{1}_m),$$

wobei die Exponenten t_k der Twiddle-Faktoren und die Permutation L der Menge $[0..N)$ definiert sind durch

$$\begin{aligned} t_k &= (k \operatorname{div} m)(k \bmod m) \quad \text{und} \\ L(k) &= (k \bmod n)m + (k \operatorname{div} n) \quad \text{für } k \in [0..N). \end{aligned}$$

Beweis In diesem Beweis werden die kompakten Notationen $x \uparrow m = (x \operatorname{div} m)$ und $x \downarrow m = (x \bmod m)$ verwendet. Damit kann die (i,j)-Komponente von L^{-1} mal der rechten Seite berechnet werden zu

$$\begin{aligned}
& \Big((\mathbf{1}_n \otimes \mathrm{DFT}_m) \cdot \mathrm{diag}\left(\omega^{t_k} \mid k \in [0..N)\right) \cdot (\mathrm{DFT}_n \otimes \mathbf{1}_m)\Big)_{i,j} \\
= & \sum_{k=0}^{N-1} \delta_{i\uparrow m, k\uparrow m} \left(\omega^n\right)^{(i\downarrow m)(k\downarrow m)} \cdot \omega^{(k\uparrow m)(k\downarrow m)} \cdot \left(\omega^m\right)^{(k\uparrow m)(j\uparrow m)} \cdot \delta_{k\downarrow m, j\downarrow m} \\
= & \sum_{k\uparrow m=0}^{n-1} \sum_{k\downarrow m=0}^{m-1} \delta_{i\uparrow m, k\uparrow m}\, \delta_{k\downarrow m, j\downarrow m}\, \omega^{n(i\downarrow m)(k\downarrow m)+(k\uparrow m)(k\downarrow m)+m(k\uparrow m)(j\uparrow m)} \\
= & \ \omega^{n(i\downarrow m)(j\downarrow m)+(i\uparrow m)(j\downarrow m)+m(i\uparrow m)(j\uparrow m)} = \omega^{(n(i\downarrow m)+i\uparrow m)\cdot(j\downarrow m+m(j\uparrow m))} \\
= & \ \omega^{(n(i\downarrow m)+i\uparrow m)\cdot j},
\end{aligned}$$

und mit Formel (A.1) ergibt sich daraus

$$\begin{aligned}
& L \cdot (\mathbf{1}_n \otimes \mathrm{DFT}_m) \cdot \mathrm{diag}\left(\omega^{t_k} \mid k \in [0..N)\right) \cdot (\mathrm{DFT}_n \otimes \mathbf{1}_m) \\
= & \ \left[\omega^{(n(L(i)\downarrow m)+L(i)\uparrow m)\cdot j}\right]_{i,j} = \left[\omega^{(n(i\uparrow n)+i\downarrow n)\cdot j}\right]_{i,j} = \mathrm{DFT}_N\,.
\end{aligned}$$

Damit ist die Behauptung bewiesen. ∎

q-Potenz-FFT Obwohl die Cooley-Tukey-FFT für jede Zerlegung $N = n \cdot m$ anwendbar ist, ist der Speziallfall $N = q^e$ für beliebige $q, e > 1$ von besonderem Interesse. Bei dieser q-Potenz-Methode, die eine mehrfache Anwendung der Zerlegung nach Cooley und Tukey ist, führt die spezielle Struktur zu einer starken Vereinfachung der Berechnung:

Die Perm-Perm-Symmetrie der DFT kann verwendet werden, um alle Permutationen an die rechte Seite der Zerlegung zu bewegen. Dies wurde bereits von Cooley und Tukey (1965) in [22] für den Spezialfall $q = 2$ verwendet. Der Spezialfall $q = 2$ ist die in der Praxis am häufigsten implementierte Methode der FFT. Die q^e-FFT wird in der Literatur auch oft als Radix-q-FFT bezeichnet, wobei q in der Regel als Zweierpotenz gewählt wird. Das folgende Lemma gibt die allgemeine Form der q-Potenz-FFT explizit an.

7.5 Lemma Sei $N = q^e$ für beliebige $q, e > 1$ und ω bezeichne eine primitive N-te Einheitswurzel. Dann gilt

$$\begin{aligned}
\mathrm{DFT}_N = & \ (\mathbf{1}_1 \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-1}}) \cdot \mathrm{diag}\left(\omega^{t(e-1,k)} \mid k \in [0..N)\right) \\
& (\mathbf{1}_q \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-2}}) \cdot \mathrm{diag}\left(\omega^{t(e-2,k)} \mid k \in [0..N)\right) \\
& \vdots \\
& (\mathbf{1}_{q^{e-2}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_q) \cdot \mathrm{diag}\left(\omega^{t(1,k)} \mid k \in [0..N)\right) \\
& (\mathbf{1}_{q^{e-1}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_1) \cdot R,
\end{aligned}$$

wobei die Exponenten $t(\alpha,k)$ der Twiddle-Faktoren und die Permutation R von $[0..N)$ bestimmt sind durch

$$\begin{aligned} t(\alpha,k) &= \Big((k \bmod q^{\alpha+1}) \operatorname{div} q^{\alpha}\Big) \cdot (k \bmod q^{\alpha}) \cdot q^{(e-1)-\alpha} \quad \text{und} \\ R(k) &= \sum_{\beta=0}^{e-1} \Big((k \operatorname{div} q^{\beta}) \bmod q\Big)\, q^{(e-1)-\beta} \quad \text{für } k \in [0..N),\ \alpha \in [1..e). \end{aligned}$$

(Die Permutation R invertiert die Wertigkeit der einzelnen Stellen von k, dargestellt im Zahlensystem zur Basis q.)

Beweis (Induktion über e.) Seien L und t_k wie bei der transponierten Cooley-Tukey-FFT mit $N = q^e$, $n = q$, $m = q^{e-1}$. Zur eindeutigen Kennzeichnung sei jedoch L_e statt L geschrieben. Es ist

$$\mathrm{DFT}_N = (\mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-1}}) \cdot \mathrm{diag}\left(\omega^{t_k} \mid k \in [0..N)\right) \cdot (\mathbf{1}_q \otimes \mathrm{DFT}_{q^{e-1}}) \cdot L_e^{-1}.$$

Offensichtlich ist $t_k = t(e \Leftrightarrow 1, k)$ für alle $k \in [0..N)$. Daher ist die Behauptung für $e = 2$ (Induktionsanfang) erfüllt und für $e > 2$ äquivalent zu

$$\begin{aligned} (\mathbf{1}_q \otimes \mathrm{DFT}_{q^{e-1}}) \cdot L_e^{-1} = {} & (\mathbf{1}_q \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-2}}) \cdot \mathrm{diag}\left(\omega^{t(e-2,k)} \mid k \in [0..N)\right) \\ & \vdots \\ & (\mathbf{1}_{q^{e-2}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_q) \cdot \mathrm{diag}\left(\omega^{t(1,k)} \mid k \in [0..N)\right) \\ & (\mathbf{1}_{q^{e-1}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_1) \cdot R. \end{aligned}$$

Die übriggebliebenen Twiddle-Matrizen zerfallen ebenfalls in Kroneckerprodukte der Form $\mathbf{1}_q \otimes X$, wie nun gezeigt wird. Dazu wird der Twiddle-Exponent $t(\alpha,k)$ vollständiger als $t(e,\alpha,k)$ bezeichnet.

Wie mit der Definition von t leicht eingesehen werden kann, gilt im Fall $1 \le \alpha \le e \Leftrightarrow 2$ die Rekursionsformel

$$t(e,\alpha,k) = q \cdot t(e \Leftrightarrow 1, \alpha, k \bmod q^{e-1}) \quad \text{für } k \in [0..q^e).$$

Damit zerfallen die Diagonalmatrizen in Kroneckerprodukte von der Form

$$\mathrm{diag}\left(\omega^{t(e,\alpha,k)} \mid k \in [0..q^e)\right) = \mathbf{1}_q \otimes \mathrm{diag}\left(\left(\omega^q\right)^{t(e-1,\alpha,k)} \mid k \in [0..q^{e-1})\right).$$

Als weitere Vorbereitung des Induktionsschrittes wird nun gezeigt, daß auch die Permutation ein Kroneckerprodukt der Form $\mathbf{1}_q \otimes X$ ist. R wird dazu vollständiger als R_e bezeichnet. Zur Erinnerung, es ist

$$\begin{aligned} L_e(k) &= \sum_{\beta=0}^{e-1} k_\beta\, q^{(\beta-1) \bmod e} \quad \text{und} \\ R_e(k) &= \sum_{\beta=0}^{e-1} k_\beta\, q^{(e-1)-\beta} \quad \text{für } k = \sum_{\beta=0}^{e-1} k_\beta\, q^{\beta} \text{ mit } k_\beta \in [0..q). \end{aligned}$$

Die Permutationen L_e und R_e operieren also auf den Tupeln $[k_0, \ldots, k_{e-1}]$ der Zahldarstellung von k zur Basis q, wie die Permutationen

$$\begin{aligned} \ell_e &= (e \Leftrightarrow 1, e \Leftrightarrow 2, \ldots, 2, 1, 0) \quad \text{und} \\ r_e &= (0, e \Leftrightarrow 1)(1, e \Leftrightarrow 2) \cdots (\lfloor e/2 \rfloor \Leftrightarrow 1, \lceil e/2 \rceil) \end{aligned}$$

auf der Menge $[0..e)$. Es gilt die Rekursionsformel

$$r_e \cdot \ell_e = r_{e-1} \quad \text{für } e > 1.$$

Die Permutation $R_e \cdot L_e$ stabilisiert daher k_{e-1} und operiert auf $[k_0, \ldots, k_{e-1}]$ wie R_{e-1}. In Matrixschreibweise

$$R_e \cdot L_e = \mathbf{1}_q \otimes R_{e-1} \quad \text{für } e > 1.$$

Nachdem die Twiddle-Matrizen und die Permutation als Kroneckerprodukte erkannt wurden, kann mit Hilfe der Induktionsvoraussetzung die Behauptung gezeigt werden. Dazu kann direkt berechnet werden:

$$\begin{aligned} & (\mathbf{1}_q \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-2}}) \cdot \mathrm{diag}\left(\omega^{t(e,e-2,k)} \mid k \in [0..q^e)\right) \\ & \vdots \\ & (\mathbf{1}_{q^{e-2}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_q) \cdot \mathrm{diag}\left(\omega^{t(e,1,k)} \mid k \in [0..q^e)\right) \\ & (\mathbf{1}_{q^{e-1}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_1) \cdot R \cdot L_e \\ = \; & (\mathbf{1}_q \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{e-2}}) \cdot \left(\mathbf{1}_q \otimes \mathrm{diag}\left((\omega^q)^{t(e-1,e-2,k)} \mid k \in [0..q^{e-1})\right)\right) \\ & \vdots \\ & (\mathbf{1}_{q^{e-2}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_q) \cdot \left(\mathbf{1}_q \otimes \mathrm{diag}\left((\omega^q)^{t(e-1,1,k)} \mid k \in [0..q^{e-1})\right)\right) \\ & (\mathbf{1}_{q^{e-1}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_1) \cdot (\mathbf{1}_q \otimes R_{e-1}) \end{aligned}$$

$$\begin{aligned} = \; & \mathbf{1}_q \otimes \\ & \Big[\left(\mathbf{1}_1 \otimes \mathrm{DFT}_q \otimes \mathbf{1}_{q^{(e-1)-1}}\right) \cdot \mathrm{diag}\left((\omega^q)^{t(e-1,(e-1)-1,k)} \mid k \in [0..q^{e-1})\right) \\ & \vdots \\ & \left(\mathbf{1}_{q^{(e-1)-2}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_q\right) \cdot \mathrm{diag}\left((\omega^q)^{t(e-1,1,k)} \mid k \in [0..q^{e-1})\right) \\ & \left(\mathbf{1}_{q^{(e-1)-1}} \otimes \mathrm{DFT}_q \otimes \mathbf{1}_1\right) \cdot R_{e-1} \Big] \\ = \; & \mathbf{1}_q \otimes \mathrm{DFT}_{q^{e-1}}, \end{aligned}$$

wobei im letzten Schritt die Induktionsvoraussetzung verwendet wurde mit der primitiven q^{e-1}-ten Einheitswurzel ω^q. ∎

An der Cooley-Tukey-FFT erkennt man deutlich die allgemeine Leistungssteigerung der Hardware: In [22] berichten Cooley und Tukey (1965) von 7.8 s Rechenzeit, die ein Rechner des Typs IBM 7094 für die DFT_{8192} benötigte. Die NUMERICAL RECIPES-Implementierung in [76], Kapitel 12.1, programmiert in BORLAND C++ auf einem PENTIUM-Prozessor mit der Taktfrequenz 150 MHz, benötigt für die gleiche Aufgabe 33.5 ms; das ist Faktor 230 schneller.

Die Methode von Rader Weder die Good-Thomas-Methode noch die Cooley-Tukey-Methode sind im Fall DFT_p für eine Primzahl p anwendbar. Für diesen Fall hat C. M. Rader in [78] gezeigt, daß DFT_p auf eine zyklische Faltung der Länge $p \Leftrightarrow 1$ zurückgeführt werden kann. Obwohl die sehr kurze und klare Veröffentlichung von Rader (1968) lange Jahre als Kuriosität gehandelt wurde, ist sie das entscheidende Bindeglied zu den anderen Zerlegungsmethoden: Wurde mit Cooley-Tukey auf Primzahllänge p reduziert, so wird mit Rader auf $p \Leftrightarrow 1$ reduziert, die Zahl $p \Leftrightarrow 1$ zerfällt und es sind wieder die anderen Methoden anwendbar.

Bemerkenswert ist außerdem, daß die Methode von Rader nicht wie die anderen FFT-Methoden (außer der Bluestein-FFT) auf der Perm-Irred-Symmetrie der DFT basiert. Vielmehr basiert die Rader-FFT auf der ebenfalls vorhandenen Perm-Perm-Symmetrie, wie bereits Beth in [10] erkannte. Die explizite Trennung der beiden Symmetriebegriffe geschah jedoch erst durch Minkwitz in [64].

7.6 Lemma Sei $p > 1$ prim, ω bezeichne eine primitive p-te Einheitswurzel und q sei der $[0..p)$-Repräsentant eines Erzeugers der multiplikativen Gruppe $(\mathbb{Z}/p\mathbb{Z})^\times$. Dann gibt es Permutationen L und R der Menge $[0..p \Leftrightarrow 1)$, so daß die DFT_p als $(1 + (p \Leftrightarrow 1)) \times (1 + (p \Leftrightarrow 1))$-Matrix aufgefaßt werden kann in der Form

$$\mathrm{DFT}_p \;=\; \left[\begin{array}{c|c} 1 & \mathbf{J}_{1\times(p-1)} \\ \hline \mathbf{J}_{(p-1)\times 1} & L \cdot \mathrm{circ}\left(w^{q^k} \mid k \in [0..p \Leftrightarrow 1)\right) \cdot R \end{array}\right].$$

Die Permutationen L und R sind eindeutig bestimmt, bis auf die Perm-Perm-Symmetrie der zirkulanten Matrix. Ein spezielles Paar L, R ist definiert durch

$$\begin{aligned} L^{-1}(k) &= (q^{(p-1)-k} \bmod p) \Leftrightarrow 1 \quad \text{und} \\ R(k) &= (q^k \bmod p) \Leftrightarrow 1 \quad \text{für } k \in [0..p \Leftrightarrow 1). \end{aligned}$$

Beweis Da $\mathrm{ggT}(p, q) = 1$ ist nach dem kleinen Satz von Fermat $q^{p-1} \overset{(p)}{\equiv} 1$. Wir betrachten die Komponentendarstellung des rechten unteren Teils der Matrizen.

$$\begin{aligned} & L^{-1} \cdot \left[\omega^{(i+1)(j+1)} \mid i, j \in [0..p \Leftrightarrow 1)\right] \cdot R^{-1} \\ = \; & \left[\omega^{(L^{-1}(i)+1)\cdot(R(j)+1)}\right]_{i,j} = \left[\omega^{q^{(p-1)-i}\cdot q^j}\right]_{i,j} = \left[\omega^{q^{j-i}}\right]_{i,j} = \mathrm{circ}(\omega^{q^k} \mid k). \end{aligned}$$

Damit ergibt sich die behauptete Darstellung von DFT_p. ∎

Zur praktischen Berechnung von DFT_p ist es also erforderlich, einen Erzeuger q der multiplikativen Gruppe $(\mathbb{Z}/p\mathbb{Z})^\times$ zu kennen. Das heißt, eine ganze Zahl $0 < q < p$ mit der Eigenschaft

$$\{1, q, q^2 \bmod p, \ldots, q^{p-2} \bmod p\} \quad = \quad \{1, \ldots, p \Leftrightarrow 1\}.$$

Ein Standardalgorithmus zur Bestimmung von q wird von Lipson (1981) in [60], Kap. IX.1.3., beschrieben. Das Verfahren basiert auf der hohen mittleren Dichte von $3/\pi^2$ der primitiven Elemente: Probiere der Reihe nach $q \in [2..p)$ und teste, ob q primitiv ist in $\mathbb{Z}/p\mathbb{Z}$ mit Hilfe der folgenden Aussage:

$$q \text{ primitiv} \quad \Leftrightarrow \quad q^{(p-1)/t} \not\equiv 1 \pmod{p} \text{ für alle Primfaktoren } t \text{ von } p \Leftrightarrow 1.$$

Die Methode von Bluestein Eine weitere Methode, um die DFT_n zu berechnen, ist ursprünglich von Bluestein (1970) in [14] verwendet worden, um zu zeigen, daß die DFT_n für *jedes* n in $O(n \log n)$ liegt. Dabei wird die DFT_n „zurückgeführt“ auf zwei größere DFTs der Länge $N \geq 2n$, bzw. auf eine zyklische Faltung der Länge N. Da N beliebig ist, kann insbesondere eine Zweierpotenz gewählt werden, und diese ist schnell realisierbar mit dem Verfahren von Cooley und Tukey. Genaugenommen entstehen allerdings nicht zwei DFT_N, sondern eine zyklische Faltung der Länge N. Die Bluestein-Methode gibt der symmetriebasierten Strukturzerlegung ein Problem auf: Sie ist bisher nicht durch darstellungstheoretische Methoden erklärt worden, denn sie stellt eine Verbindung her zwischen Gruppen unterschiedlicher Ordnung und Struktur.

7.7 Lemma Sei $N \geq 2n$ und ω eine primitive $(2n)$-te Einheitswurzel. Dann ist

$$\begin{aligned} \mathrm{DFT}_n \quad = \quad & \mathrm{diag}\left(\omega^{k^2} \mid k \in [0..n)\right) \\ & \mathbf{1}_{n\times N} \cdot \mathrm{circ}\left(v_k \mid k \in [0..N)\right) \cdot \mathbf{1}_{N\times n} \\ & \mathrm{diag}\left(\omega^{k^2} \mid k \in [0..n)\right), \end{aligned}$$

wobei der Faltungskern $[v_0, \ldots, v_{N-1}]$ definiert ist durch

$$v_k \quad = \quad \begin{cases} \omega^{-k^2} & \text{für } 0 \leq k < n, \\ 0 & \text{für } n \leq k < N \Leftrightarrow n \text{ und} \\ \omega^{-(k-(N-n))^2+n^2} & \text{für } N \Leftrightarrow n \leq k < N. \end{cases}$$

Beweis Vergleich der Komponenten liefert

$$\begin{aligned} \mathrm{diag}(\omega^{-k^2} \mid k) \cdot \mathrm{DFT}_n \cdot \mathrm{diag}(\omega^{-k^2} \mid k) \quad &= \quad \left[\omega^{-i^2} \cdot \left(\omega^2\right)^{ij} \cdot \omega^{-j^2}\right]_{i,j} \\ &= \quad \left[\omega^{-(j-i)^2}\right]_{i,j} \quad \text{und} \\ \mathbf{1}_{n\times N} \cdot \mathrm{circ}\left(v_k \mid k \in [0..N)\right) \cdot \mathbf{1}_{N\times n} \quad &= \quad \left[v_{(j-i) \bmod N} \mid i, j \in [0..n)\right]. \end{aligned}$$

Wegen $N \geq 2n$ gilt für die $i, j \in [0..n)$

$$(j \Leftrightarrow i) \bmod N \quad = \quad \begin{cases} j \Leftrightarrow i \in [0..n) & \text{für } j \geq i \text{ und} \\ j \Leftrightarrow i + N \in [N \Leftrightarrow (n \Leftrightarrow 1)..N) & \text{für } j < i. \end{cases}$$

Werden die Fälle einzeln behandelt, dann ergibt sich

$j \geq i$: Dann ist $v_{(j-i) \bmod N} = v_{j-i} = \omega^{-(j-i)^2}$ nach Definition von v_k.

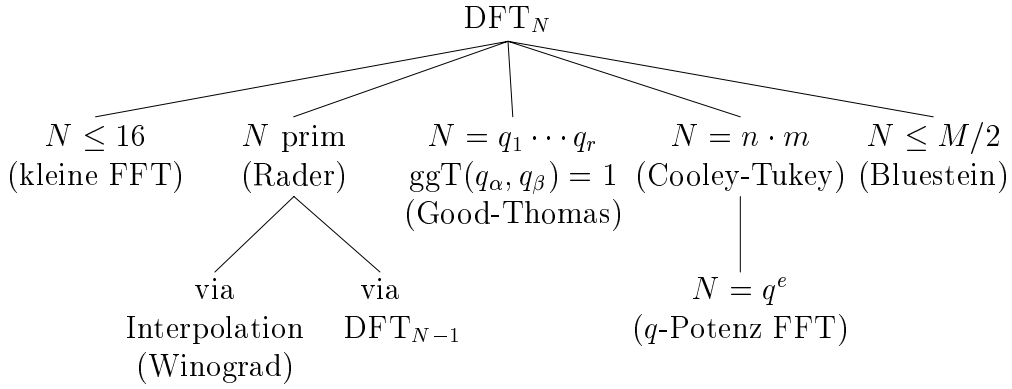
$j < i$: Dann folgt für den anderen Fall der Definition von v_k

$$v_{(j-i) \bmod N} \quad = \quad v_{j-i+N} = \omega^{-((N+j-i)-(N-n))^2+n^2} = \omega^{-(j-i)^2}.$$

Damit ist die behauptete Formel gezeigt. ∎

7.4 Zusammenspiel der Zerlegungsschritte

Offensichlich ist jede der bisher vorgestellten FFT-Methoden strukturell ein Teile-und-Herrsche Algorithmus. Daher ist die Auswahl der günstigsten Methode ein wesentlicher Punkt bei der Konstruktion eines schnellen Algorithmus. Zunächst ein Überblick über die Methoden, wobei zwei bisher noch nicht erwähnte Methoden hinzukommen: Die kleinsten FFTs, bis etwa $N \leq 16$, können als straight-line-Programme effizient kodiert werden. Eine Anzahl von kleinen FFTs kann dem Buch [67] von Nussbaumer (1982), Kapitel 5.5 (dort für $N \cdot \mathrm{DFT}_N^{-1}$), entnommen werden. Zudem kann die zyklische Faltung der Rader-FFT statt mit einer DFT auch durch Interpolation berechnet werden. Diese von Winograd in [94] beschriebene Methode kann für kleine N von Vorteil sein, da sie eine kleine Zahl von Multiplikationen bei nur wenig mehr Additionen benötigt. Die Methode von Winograd soll hier nicht weiter erläutert werden. Zunächst die Methoden im Überblick:



Die erste vorgestellte Methode zur Auswertung der DFT_N für alle $N > 0$ ist besonders einfach zu implementieren und verwendet trotzdem in jedem Fall einen schnellen Algorithmus.

7.8 Algorithmus (Einfach zu implementierende FFT) Ist $N = 2^e$, dann verwende eine Zweierpotenz-FFT. Anderenfalls wähle $M = 2^e \geq 2N$ und verwende die Bluestein-Methode, um DFT_N auf DFT_M zurückzuführen.

Im Gegensatz zu diesem einfachen Algorithmus ist in vielen Bibliotheken für numerische Programme nur die 2^e-FFT verfügbar, allerhöchstens eine Cooley-Tukey-FFT, die N bis auf Primzahlen zerkleinert. Bei einer derartigen „mixed radix FFT“ ist die Methode von Rader oft *nicht* implementiert. Dies erkennt man in der Praxis daran, daß die „schnelle“ Fourier-Transformation für $N = 65537$ (prim) etwa um den Faktor 30000 langsamer ist als für $N \Leftrightarrow 1 = 65536$ (Zweierpotenz). Wird die Rader-FFT verwendet, so ist nur eine Verlangsamung um den Faktor 2 zu erwarten; mit Algorithmus 7.8 ist der Faktor etwa 8.

Die nächste Kompositionstrategie ist leistungsfähiger und verwendet alle Methoden außer der von Bluestein. Sie hat den Vorteil, daß die DFT_N immer auf Transformationen kleinerer Länge reduziert wird.

7.9 Algorithmus (Greedy-FFT) Ist N klein und ist dafür eine direkt kodierte FFT verfügbar, so verwende diese. Ist N prim, dann verwende die Rader-FFT. Ist $N = 2^e$, dann verwende die Zweierpotenz-FFT. Ist $N = q^e$ und ist DFT_q direkt kodiert, dann verwende die q-Potenz-FFT. Ist $N = q_1 \cdots q_r$ mit teilerfremden q_α, so verwende die Good-Thomas-FFT. Andernfalls ist $N = p^e$ mit einer Primzahl p, die nicht direkt kodiert ist. In diesem Fall verwende die q-Potenz-FFT.

Die soeben vorgestellte Kompositionsstrategie ist nicht optimal. Namentlich kann mit der Methode von Bluestein für manche N mit besonders „primer“ Zerlegungsstruktur ein großer Gewinn verbunden sein. Für $N = 719$ beispielsweise ergibt Algorithmus 7.9 die folgende Zerlegungsstruktur:

```
rader[ (* N = 719 (prime) *)
  goodThomas[ (* N = 718 = 2 * 359 *)
    small[2],
    rader[ (* N = 359 (prime) *)
      goodThomas[ (* N = 358 = 2 * 179 *)
        small[2],
        rader[ (* N = 179 (prime) *)
          goodThomas[ (* N = 178 = 2 * 89 *)
            small[2],
            rader[ (* N = 89 (prime) *)
              goodThomas[ (* N = 88 = 8 * 11 *)
                small[8],
                rader[ (* N = 11 (prime) *)
                  goodThomas[ (* N = 10 = 2 * 5 *)
                    small[2],
                    small[5]
```

```
                    ]
                  ]
                ]
              ]
            ]
          ]
        ]
      ]
    ]
  ]
```

Diese FFT benötigt 304 ms auf einer 486-CPU bei 66 MHz Taktfrequenz. Nur 61.5 ms benötigt dagegen die folgende Methode

```
bluestein[ 719, (* = N <= 1440/2 *)
  goodThomas[ (* N = 1440 = 5 * 9 * 32 *)
    small[5],
    small[9],
    qPower[ (* N = 32 = 2^5 *)
      small[2],
      5
    ]
  ]
]
```

(Die Terme wurden mit dem FFT-Programm des Autors automatisch erzeugt; es ist im nächsten Abschnitt beschrieben.) Die soeben vorgestellte Strategie zur Wahl der jeweiligen Methode basiert auf der impliziten Annahme, daß sich für jede Zahl N aufgrund ihrer zahlentheoretischen Struktur a priori entscheiden läßt, welche Methode die schnellste sein wird. Diese Annahme ist in der Praxis nicht zulässig!

Tatsächlich hängt die Wahl der Methode extrem stark von der jeweiligen Implementierung und der zugrunde liegenden Architektur ab. Die Komplexitätsbetrachtungen etwa in [19], oder das Auszählen der arithmetischen Operationen in [67], sind daher für die Praxis nur sehr eingeschränkt von Nutzen. Theoretische Kostenmaße verleiten in diesem Fall sogar zu falschen Annahmen.

Dazu ein Beispiel: Eine sehr gute Zweierpotenz-FFT ist in den NUMERICAL RECIPES (1992) [76], Kapitel 12.1, veröffentlicht. Das dort angegebene unscheinbare C-Programm von 50 Zeilen Quelltext nutzt auf sehr geschickte Weise die Speicherhierachie der heute gängigen Rechner-Architekturen aus. Das Programm ist der Zahlenwolf im Schafspelz. (Die Implementierung basiert auf einem Programm von N. M. Brenner.) Namentlich wird die Zweierpotenz-FFT damit so schnell, daß Bluestein und 2^e-FFT für viele Fälle deutlich schneller ist, als eine

moderat gut implementierte Zerlegung in Primfaktoren. Und das, obwohl die Zerlegungsmethode aus theoretischer Sicht unbedingt vorzuziehen ist!

Die Wahl der optimalen Methode für die DFT_n hängt also so stark von der Implementierung und der Architektur der Hardware ab, daß sie nur anhand von Laufzeitinformationen gefällt werden kann. Ein naheliegender Ansatz dazu basiert auf dynamischer Programmierung:

7.10 Algorithmus (FFT mit dyn. Progr., ohne Bluestein) Bis zu einer vorgegebenen Schranke werden für alle $N = 1, 2, \ldots$ (in dieser Reihenfolge) jeweils die anwendbaren Methoden der Zerlegung (außer Bluestein) miteinander verglichen. Die jeweils schnellste wird für N in einer Tabelle gespeichert. Da die DFT_N auf kleinere DFTs zurückgeführt wird, sind diese schon optimal gewählt, wenn N betrachtet wird.

Die Methode von Bluestein mußte ausgespart werden, weil sie die Struktur der dynamischen Programmierung sprengt: Bluestein führt die DFT_N auf eine *größere* DFT zurück. Wie an dem obigen Beispiel $N = 719$ deutlich wurde, kann die Bluestein-Methode jedoch nicht einfach vernachlässigt werden, dazu ist sie zu wichtig. Einen Ausweg aus diesem Dilemma bietet die iterative Optimierung der von Algorithmus 7.10 gemessenen Tabelle. Die Grundannahme dabei ist, daß die Bluestein-Methode nur selten nutzbringend eingesetzt werden kann. Damit gelangt man im wesentlichen zu der folgenden dynamischen Kompositionsstrategie:

7.11 Algorithmus (FFT mit dyn. Progr., mit Bluestein) Bestimme mit Algorithmus 7.10 zu jedem N bis zu einer festen Schranke die jeweils schnellste FFT-Methode, ohne Berücksichtigung von Bluestein. Dann betrachte alle N, zu denen es ein $M > 2N$ gibt, so daß DFT_M mindestens Faktor 2 schneller ist als DFT_N. Vergleiche die Bluestein-Methode via M mit der bisher besten Methode für N. Wird eine weitere Iteration gewünscht, dann führe wieder Algorithmus 7.10 aus, usw.

Der Hauptnachteil der dynamischen Optimierung, wie in den Algorithmen 7.10 und 7.11 beschrieben, ist der hohe Bedarf an Laufzeitdaten. Soll etwa die FFT für $N \in [1..1024]$ optimiert werden, so müssen mindestens 10^4 Messungen vorgenommen werden. Dauert jede Messung 10 Sekunden, so kommen insgesamt schon 28 Stunden Rechenzeit zusammen. Die im nächsten Abschnitt besprochene FFT des Autors verwendet daher nur eine um Bluestein erweiterte Variante von Algorithmus 7.9.

Alle Erwägungen zur Wahl der FFT-Methode in diesem Abschnitt beziehen sich nur auf Software-Implementierungen mit einem einzigen Universalprozessor als aktive Resource. Werden parallele Architekturen betrachtet oder die ultimative Form der Parallelität, Entwurf von Hardware, so muß alles erneut überdacht werden. Dieser Aspekt der FFT-Methoden soll hier nicht betrachtet werden.

7.5 Eine konkrete Implementierung

Zur experimentellen Überprüfung der theoretischen Ergebnisse wurde vom Autor ein Programm zur Fourier-Transformation in der Programmiersprache C realisiert. Die Hauptleistungsmerkmale sind

- Generische Arithmetik für beliebigen Grundkörper bei hohem Durchsatz für numerische Berechnungen in Charakteristik 0 (komplexe Zahlen).
- Effiziente Algorithmen für beliebige Signallänge. Es wird in jedem Fall ein $N \log(N)$-Algorithmus verwendet, auch für große Primzahlen.
- Hohe Modularität. Das Programm ermöglicht es, die einzelnen FFT-Zerlegungsmethoden beliebig zu kombinieren. Die Zerlegungsmethode kann als Term in MATHEMATICA-Syntax ein- und ausgegeben werden. (Zwei Beispiele hierzu im vorigen Abschnitt.)
- Messung von Laufzeitdaten. Zur Bewertung alternativer Zerlegungen können Laufzeitdaten ermittelt werden. Insbesondere gibt es ein Programm, welches das FFT-Modul auf eine gegebene Architektur hin konfiguriert (siehe unten).
- Portabilität. Das Programm wurde unter mehreren C-Compilern und Betriebssystemen simultan entwickelt: GNU-C++ auf LINUX und auf SOLARIS sowie MS VISUAL C++ und BORLAND C++ auf WINDOWS NT, WINDOWS 95 und MS-DOS.
- Das FFT-Modul ist reentrant, kann also nebenläufig durch mehrere Aktivitätsbahnen des Anwendungsprozesses verwendet werden.

Das Programm ist so effizient und flexibel, daß es von H. Aagedal und M. Schmid in das DIGIOPT-System integriert wurde. Das DIGIOPT-System ist ein Programm mit graphischer Benutzeroberfläche zur Simulation und zum Entwurf diffraktiver optischer Elemente. Es wurde am Institut für Algorithmen und Kognitive Systeme der Universität Karlsruhe entwickelt und wird in der Dissertation von H. Aagedal beschrieben (in Vorbereitung).

Für die Simulation und den Entwurf diffraktiver optischer Elemente ist eine schnelle und flexible Fourier-Transformation von zentraler Bedeutung. In einem interaktiven System wie DIGIOPT gilt dies um so mehr. Mit dem FFT-Moduls des Autors ist es zudem möglich geworden, in DIGIOPT diffraktive Elemente von beliebiger Kantenlänge effizient zu transformieren. Die graphische Benutzeroberfläche von DIGIOPT verwendet Multi-Threading und erreicht dadurch eine hohe Nebenläufigkeit. Daher war Reentranz ein weiteres wesentliches Designziel für die Implementierung der Fourier-Transformation.

Wie bereits im vorigen Abschnitt ausgeführt, *kann die Entscheidung für oder gegen eine spezielle Zerlegungsmethode nur anhand von gemessenen Laufzeitdaten*

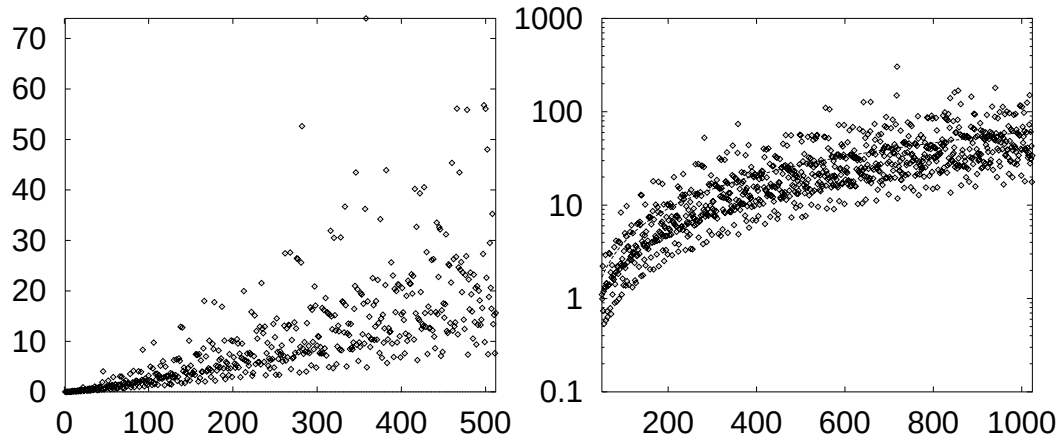


Abbildung 7.1: $t(N)$/ms mit Greedy-Strategie a) linear, b) logarithmisch.

erfolgen. Im folgenden werden daher einige Erkenntnisse aus gemessenen Laufzeitinformationen präsentiert. Die Daten wurden auf einem Arbeitsplatzrechner mit einer CPU vom Typ 486 bei der Taktfrequenz von 66 MHz mit 16 MB RAM unter dem Betriebssystem LINUX erhoben. Jede Einzelmessung erstreckte sich über mindestens 17 Sekunden. Bei einer geschätzten Genauigkeit der Zeitmessung von 100 ms ergibt sich eine eine relative Genauigkeit der Datenpunkte von 0.6%.

Die Abbildung 7.1a) gibt eine Vorstellung von der absoluten Laufzeit $t(N)$ einer FFT der Länge N, zerlegt nach der Greedy-Strategie 7.9. Bemerkenswert ist die große Fluktuation benachbarter Werte. Die große Variation ist zahlentheoretisch motiviert, denn $N+1$ zerfällt völlig anders in Primfaktoren als N. In der logarithmischen Auftragung von Abbildung 7.1b) wird das zu erwartende $O(N \log N)$-Gesetz sichtbar. Dieses Verhalten legt die folgende phänomenologische Theorie der Laufzeit nahe:

$$t(N) = c \cdot N \log_2(N) \cdot \kappa(N) \quad \text{für eine Konstante } c > 0.$$

Die Konstante enthält die absolute Geschwindigkeit, der Term $N \log_2(N)$ erfaßt den glatten Anteil des Wachstums und $\kappa(N)$ die schnell oszillierende Abweichung vom Mittelwert. Als Konstante wurde $c = 6.0135\,\mu$s gemessen. Die Funktion κ ist in Abbildung 7.2a) dargestellt. Sie zeigt, daß die Laufzeit der FFT um bis zu zwei Größenordnung vom Mittelwert abweichen kann.

Durch geschickte Verwendung der Methode von Bluestein kann der Variationsbereich von κ stark reduziert werden: Wie im vorigen Abschnitt gezeigt wurde, wird zum Beispiel die Laufzeit für $N = 719$ mit der Bluestein-Methode um den Faktor 5 verringert. Ein Profiling-Programm sucht systematisch Ausreißer dieser Art und realisiert die FFT für diese Längen mit der Bluestein-Methode. Das Ergebnis der Optimierung ist in Abbildung 7.2b) dargestellt. Die Messung der Laufzeitdaten benötigte zwei Tage CPU-Zeit.

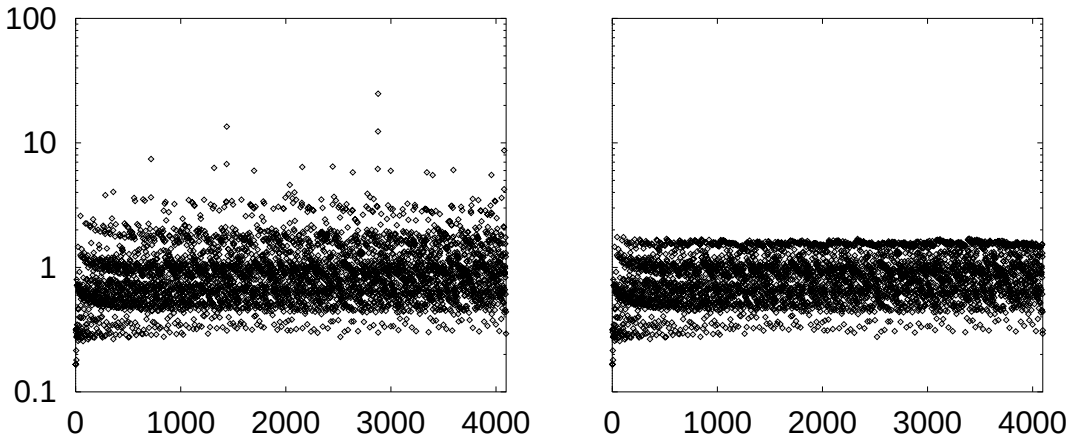


Abbildung 7.2: Abweichungsfaktor $\kappa(N)$ a) ohne, b) mit Bluestein-FFT.

7.6 Zusammenfassung

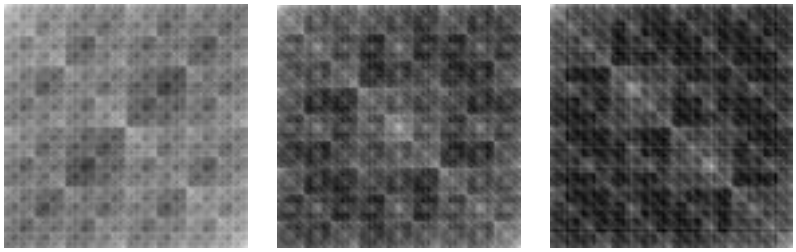
In diesem Kapitel wurden die klassischen Methoden der schnellen Fourier-Transformation (FFT) besprochen. Die Methoden basieren auf zwei wesentlichen Typen von Symmetrie: Einer Perm-Irred-Symmetrie vom Isomorphietyp Z_N und einer Perm-Perm-Symmetrie vom Isomorphietyp $\mathrm{Aut}(\mathsf{Z}_N)$. Dabei bezeichnet N die Anzahl der transformierten Einzelsignale (Abschnitt 7.2).

Die einzelnen Methoden führen eine DFT der Ordnung N auf mehrere DFTs anderer Ordnungen zurück. Obwohl nicht für jedes N jede Zerlegungsmethode anwendbar ist, ergänzen sich die Methoden passend. Damit ergibt sich strukturell ein Teile-und-Herrsche-Algorithmus. Die Strategie zur Auswahl des jeweils günstigsten Zerlegungsschritts ist in der Praxis ein großes Problem. Die Entscheidung kann nur anhand gemessener Laufzeitdaten verläßlich gefällt werden.

Um dies in der Praxis konkret nachzuprüfen, wurde eine hoch optimierte Fourier-Transformation in der Programmiersprache C realisiert. Das Programm hat einen Umfang von 4.8 kloc. Es ist effizient, flexibel, reentrant sowie portabel und wurde erfolgreich in den Kern des DigiOpt-Systems von H. Aagedal et al. (1993–97) zum Entwurf diffraktiver Optik integriert.

8

Integrabilität von Spin-Gitter-Modellen



Ising $L = 8$ *Potts* $q = 3, L = 5$ *Potts* $q = 4, L = 4$

Dieses Kapitel beschäftigt sich mit einer konkreten Verbindung zwischen der Symmetriesuche im Sinne dieser Arbeit und einem Teilgebiet der statistischen Physik. Es geht hierbei um sogenannte Spin-Gitter-Modelle und deren „experimentelle“ Untersuchung mit Hilfe von Computersimulationen. Namentlich kann die Symmetriesuche einige Strukturinformationen automatisch beschaffen und dadurch die numerische Simulation effizienter gestalten.

Um präzise darstellen zu können, an welcher Stelle die Symmetriesuche etwas beiträgt, ist es notwendig, ein gutes Stück weit in das Gedankengebäude der statistischen Physik einzudringen. Dies wird in dem nächsten Abschnitt passieren. Die dort verwendeten Begriffe der Hamiltonfunktion, der freien Energie, der Zustandssumme und der Transfermatrix sind Standardkonstrukte der statistischen Physik. Sie finden sich in jedem Einführungstext über statistische Physik, wie etwa dem Buch [80] von L. E. Reichl. Ebenso ist das hier beispielhaft erläuterte 2d-Ising-Modell ein wohluntersuchtes Standardmodell. Reichel gibt die Lösung für ein und zwei Dimensionen an, [80], 9.F.4.(a) und (b). Das erwähnte statistische Kriterium zur Integrabilität eines Spin-Gitter-Modells ist neueren Datums. Es fußt in der Theorie der Zufallsmatrizen und wurde von H. Meyer und J.-C. Anglès d'Auriac durch Computersimulationen „experimentell“ untersucht. Ein Standardwerk bezüglich Zufallsmatrizen ist das Buch [62] von M. L. Mehta

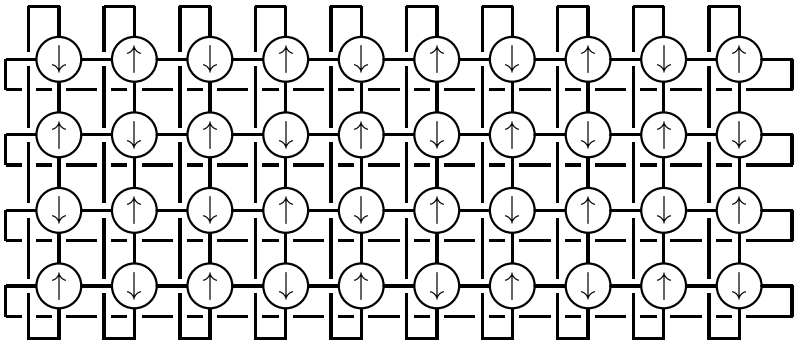


Abbildung 8.1: (4×10)-Ising-Modell, antiferromagnetischer Zustand

(1967). Es ist hauptsächlich die Dissertation [63] von H. Meyer (1996) am Centre de Recherche sur les Très Basses Températures (CRTBT) am CNRS Grenoble, die der Darstellung hier zugrunde liegt. Insbesondere wurde auch die hier präsentierte Sichtweise der Symmetriesuche im Kontext der statistischen Physik im Dialog mit Hendrik Meyer entwickelt.

Das Kapitel beginnt mit der Einführung der relevanten Grundbegriffe aus der statistischen Physik in Abschnitt 8.1. Danach folgt in Abschnitt 8.2 eine kurze Einführung in die wichtige physikalische Fragestellung der Integrabilität eines Systems. In Abschnitt 8.3 wird beispielhaft das $(L \times M)$-Ising-Modell für $L = 4$ ohne äußeres Feld betrachtet. Insbesondere wird erklärt, wie die Transfermatrix definiert ist und es wird die Symmetrie bestimmt. Danach wird in Abschnitt 8.4 die Symmetrie des Beispiels $L = 4$ bestimmt und anschaulich gedeutet. Dazu werden die in dieser Arbeit vorgestellten Symmetriesuchverfahren eingesetzt. In Abschnitt 8.5 werden die Ergebnisse der Symmetriesuche bei weiteren Modellen angegeben. In Abschnitt 8.6 findet sich eine Zusammenfassung des Kapitels.

8.1 Grundbegriffe

Ein *Spin-Gitter-Modell* beschreibt ein physikalisches System durch einen ungerichteten Graphen, an dessen Knoten sich identische Kopien eines Elementarsystems befinden, Abbildung 8.1. Ein Teilsystem wird gewöhnlich als *Spin* bezeichnet, denn bei den einfachsten Spin-Gitter-Modellen stellt man sich die einzelnen Teilsysteme als kleine Magnete vor. Ein Teilsystem ist in der Regel sehr einfach strukturiert; es hat meist nur wenige Zustände. Die einzelnen Kopien wechselwirken ausschließlich mit ihren nächsten Nachbarn, gemäß der Verbindungstopologie des Graphen, und eventuell noch mit einem homogenen äußeren Feld H_{mag}. Der Graph ist ein Gitter und die Wechselwirkung von nächsten Nachbarn ist homogen, das heißt, sie hängt nicht vom absoluten Ort ab. Das ganze Spin-Gitter-System befindet sich in Kontakt mit einem Wärmebad konstanter Temperatur $T > 0$, so daß sich ein thermodynamisches Gleichgewicht ausbildet.

Die Dynamik des Systems wird beschrieben durch seine *Hamiltonfunktion* $\mathsf{H}(\{\sigma_i\}, H_{\text{mag}})$, wobei $\{\sigma_i\}$ einen Zustand der Spins bezeichnet. Das Symbol $\{\sigma_i\}$ steht für das Tupel der einzelnen Spinzustände σ_i wobei i alle Gitterplätze durchläuft. Bei dem besonders einfachen Ising-Modell hat jeder Spin nur zwei mögliche Zustände, sie seien mit $\uparrow = +1$ und $\downarrow = \Leftrightarrow 1$ bezeichnet.

Die thermodynamischen Eigenschaften, wie die mittlere Ordnung, die Wärmekapazität, die Phasenübergänge, etc., werden vollständig durch die helmholtzsche *freie Energie* $F(T, H_{\text{mag}})$ beschrieben. Diese kann ausgedrückt werden durch die *Zustandssumme* (engl. partition function) $Z(T, H_{\text{mag}})$ gemäß der Relation

$$F(T, H_{\text{mag}}) = \Leftrightarrow k_{\text{B}} T \log Z(T, H_{\text{mag}}).$$

Da die Teilsysteme Boltzmann-verteilt sind, ist die Zustandssumme des Systems

$$Z(T, H_{\text{mag}}) = \sum_{\{\sigma_i\}} \exp\left(\Leftrightarrow \frac{\mathsf{H}(\{\sigma_i\}, H_{\text{mag}})}{k_{\text{B}} T} \right).$$

(Die Summation durchläuft alle möglichen Zustände des gesamten Spin-Gitter-Systems. k_{B} bezeichnet die Boltzmann-Konstante.) Die beiden letzten Gleichungen verknüpfen die Dynamik des Systems, beschrieben durch die Hamiltonfunktion H, mit dem thermodynamischen Verhalten, beschrieben durch F. Die dabei verwendete Zustandssumme Z ist die wesentliche kombinatorische und statistische Konstruktion. Beim Ising-Modell mit zwei Spin-Zuständen und $L \times M$ Gitterplätzen ist die Zustandssumme eine Summe von 2^{LM} vielen Exponentialfunktionen in T, J und H_{mag}.

Die Zustandssumme kann vereinfacht dargestellt werden durch die *Transfermatrix*. Sie erlaubt es, die Regularität des zugrundeliegenden Gitters zu nutzen. Auf einen zweidimensionalen Torus mit $(L \times M)$ Gitterplätzen ist die Zustandssumme gerade die Spur

$$Z = \text{tr}(T^M)$$

mit der Transfermatrix T. Die Transfermatrix erfaßt die Erweiterung des Gitters um eine weitere Spalte. Beim Ising-Modell ist sie eine $(2^L \times 2^L)$-Matrix. Mit der Darstellung der Zustandssumme durch die Transfermatrix wird die Abhängigkeit der Zustandssumme von M sehr einfach: Wird T durch U diagonalisiert, also $U^{-1} T U = \text{diag}(\lambda_i \mid i)$, dann ist

$$Z = \text{tr}(T^M) = \text{tr}\left(U^{-1} \,\text{diag}(\lambda_i^M \mid i) U \right) = \sum_i \lambda_i^M.$$

Die Eigenwerte der Transfermatrix zu festem L liefern daher direkt die Zustandssumme des Spin-Gitters der Größe $(L \times M)$ für jedes $M \geq 1$. Insbesondere ist im thermodynamischen Grenzwert $M \to \infty$ die freie Energie je Gitterplatz

$$f = F/(LM) = \Leftrightarrow k_{\text{B}} T \log(\lambda_{\max}),$$

wobei $\lambda_{\max}$ den betragsgrößten Eigenwert der Transfermatrix bezeichnet. Die Transfermatrix ist das wesentliche Studienobjekt bei der Symmetriesuche im Sinne dieser Arbeit.

8.2 Integrabilität

Eine wesentliche physikalische Fragestellung ist die Integrabilität eines Systems. Informell bedeutet Integrabilität, daß das System im Prinzip exakt gelöst werden kann. Für die hier betrachteten Spin-Gitter-Modelle gibt es einige äquivalente formale Kriterien der Integrabilität: Die Existenz einer Lösung der Yang-Baxter-Gleichungen, die Gültigkeit der Dreieck-Stern-Beziehungen, die Existenz einer einparametrigen Familie von kommutierenden Transfermatrizen und die Gültigkeit des Bethe-Ansatzes.

Im allgemeinen ist die Integrabilität mit den analytischen Kriterien sehr schwierig zu entscheiden und setzt eine gute Kenntnis des Systems voraus. Strukturell ist die Integrabilität eines Systems gleichbedeutend mit der Existenz einer ausreichend großen Symmetrie — etwa in Form der Dreieck-Stern-Beziehungen. Diese Art von Symmetrie ist allerdings nicht notwendigerweise von der in dieser Arbeit behandelten Form und kann deshalb auch nicht mit den in anderen Kapiteln dargestellten Verfahren gefunden werden.

Es gibt jedoch ein statistisches Kriterium zur Integrabilität, bei dem die Symmetriesuchverfahren nützlich sind. Das Kriterium entstand aus der Theorie der Zufallsmatrizen (engl. random matrix theory, RMT) und besagt: Die Eigenwerte bei Transfermatrizen integrabler Systeme sind statistisch anders verteilt als bei Transfermatrizen nicht-integrabler Systeme. Um also einen Aufschluß über die Integrabilität eines Systems zu erhalten, wird das Eigenwertspektrum numerisch bestimmt und statistisch gegen verschiedene Ensembles von Zufallsmatrizen getestet.

Die Theorie der Zufallsmatrizen beschäftigt sich mit der statistischen Verteilung der Eigenwerte von Matrizen aus Zufallsvariablen. Dabei wird die Verteilung der Komponenten der Matrix und eine Eigenschaft der ganzen Matrix vorgegeben (etwa Orthogonalität). Unter diesen Einschränkungen ergeben sich für die Eigenwerte ganz spezielle Klassen von Verteilungsfunktionen. Obwohl die Transfermatrix eines bestimmten Spin-Gitter-Modells keine Zufallsmatrix ist (sondern eine ganz bestimmte Matrix), kann die Menge der Eigenwerte als Stichprobe gegen die verschiedenen möglichen Verteilungen getestet werden.

Eine wichtige Rolle bei dieser Berechnung spielt die Symmetrie der Transfermatrix, diesmal in dem technischen Sinne der Symmetriesuche: Jeder bekannte Teil der Symmetrie kann verwendet werden, um die Transfermatrix auf eine Blockdiagonalform zu transformieren. Dann erst müssen die einzelnen Blöcke aufwendig numerisch diagonalisiert werden. Schließlich kann mit den in dieser Arbeit vorgestellten Verfahren bewiesen werden, daß eine vorgegeben Transfermatrix keine weiteren Symmetrieoperationen eines bestimmten Typs besitzt. Dies soll im folgenden an einem illustrativen und überschaubaren Beispiel vorgeführt werden. Es handelt sich dabei um das gut untersuchte zweidimensionale Ising-Modell.

8.3 Das $(L \times M)$-Ising-Modell ohne Feld

Die Hamiltonfunktion des Ising-Modells lautet

$$\mathsf{H} = \Leftrightarrow J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \Leftrightarrow H_{\mathrm{mag}} \sum_i \sigma_i.$$

Dabei ist k_{B} die Boltzmann-Konstante, $J > 0$ ist die Kopplungskonstante der Spin-Spin-Wechselwirkung und $\langle i,j \rangle$ bedeutet, daß i und j nächste Nachbarn sind. (Die Kopplungskonstante $g\mu_{\mathrm{B}}$ an das äußere Feld wurde wegnormiert.) Ab jetzt wird der Spezialfall ohne äußeres Feld, $H_{\mathrm{mag}} = 0$, betrachtet.

Beim $(L \times M)$-Ising-Modell ist der Nachbarschaftsgraph ein zweidimensionaler Torus mit L Zeilen und M Spalten. Eine Spalte hat 2^L Zustände, sie seien mit $\alpha \in \{1..2^L\}$ bezeichnet. Mit $\alpha_i \in \{\Leftrightarrow 1, 1\}$ wird der Zustand des Spins an Position $i \in \{1..L\}$ bezeichnet. Die symmetrisierte Transfermatrix T ist eine Funktion der dimensionslosen Größe $w = \exp(J/(k_{\mathrm{B}}T))$. Namentlich ist die (α, α')-Komponente von T die Potenz

$$T_{\alpha,\alpha'} = w^{t_{\alpha,\alpha'}},$$

mit der Matrix t der ganzzahligen Exponenten

$$t_{\alpha,\alpha'} = \sum_i (\alpha_i \alpha_{i+1}/2 + \alpha_i \alpha'_i + \alpha'_i \alpha'_{i+1}/2).$$

Für den Spezialfall $L = 4$ lautet die Exponentenmatrix der Transfermatrix

$$t = \begin{bmatrix}
8 & 4 & 4 & 2 & 4 & 0 & 2 & 0 & 4 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\
4 & 4 & 0 & 2 & 0 & 0 & -2 & 0 & 0 & 2 & -4 & 0 & -2 & 0 & -4 & 0 \\
4 & 0 & 4 & 2 & 0 & -4 & 2 & 0 & 0 & -2 & 0 & 0 & -2 & -4 & 0 & 0 \\
2 & 2 & 2 & 4 & -2 & -2 & 0 & 2 & -2 & 0 & -2 & 2 & -4 & -2 & -2 & 2 \\
4 & 0 & 0 & -2 & 4 & 0 & 2 & 0 & 0 & -2 & -4 & -4 & 2 & 0 & 0 & 0 \\
0 & 0 & -4 & -2 & 0 & 0 & -2 & 0 & -4 & -2 & -8 & -4 & -2 & 0 & -4 & 0 \\
2 & -2 & 2 & 0 & 2 & -2 & 4 & 2 & -2 & -4 & -2 & -2 & 0 & -2 & 2 & 2 \\
0 & 0 & 0 & 2 & 0 & 0 & 2 & 4 & -4 & -2 & -4 & 0 & -2 & 0 & 0 & 4 \\
4 & 0 & 0 & -2 & 0 & -4 & -2 & -4 & 4 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\
2 & 2 & -2 & 0 & -2 & -2 & -4 & -2 & 2 & 4 & -2 & 2 & 0 & 2 & -2 & 2 \\
0 & -4 & 0 & -2 & -4 & -8 & -2 & -4 & 0 & -2 & 0 & 0 & -2 & -4 & 0 & 0 \\
0 & 0 & 0 & 2 & -4 & -4 & -2 & 0 & 0 & 2 & 0 & 4 & -2 & 0 & 0 & 4 \\
2 & -2 & -2 & -4 & 2 & -2 & 0 & -2 & 2 & 0 & -2 & -2 & 4 & 2 & 2 & 2 \\
0 & 0 & -4 & -2 & 0 & 0 & -2 & 0 & 0 & 2 & -4 & 0 & 2 & 4 & 0 & 4 \\
0 & -4 & 0 & -2 & 0 & -4 & 2 & 0 & 0 & -2 & 0 & 0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 2 & 0 & 0 & 2 & 4 & 0 & 2 & 0 & 4 & 2 & 4 & 4 & 8
\end{bmatrix},$$

sie erfaßt die Thermodynamik aller $(4 \times M)$-Ising-Modelle für $M \geq 1$. Dieses Beispiel entstammt dem Exemple illustratif 7.2.3. aus [63].

8.4 Symmetrie des $(4 \times M)$-Ising-Modells

In diesem Abschnitt werden die Symmetriesuchverfahren der vorliegenden Arbeit angewendet auf die (16×16)-Exponentenmatrix t aus dem letzten Abschnitt.

Es wird der Mon-Mon-Symmetrietyp betrachtet. Bei dieser Verallgemeinerung der Perm-Perm-Symmetrie können die Matrizen links und rechts monomial sein. Das heißt, in jeder Zeile und jeder Spalte ist genau ein Eintrag ungleich Null. Da die Komponenten von t reell sind, kommen als Einträge der monomialen Matrizen nur die reellen Einheitswurzeln 1 und $\Leftrightarrow$1 in Frage. Damit kann das (16×16)-Mon-Mon-Problem elegant auf ein (32×32)-Perm-Perm-Problem reduziert werden. Das Perm-Perm-Problem wird mit der partitionsbasierten Backtracksuche aus Kapitel 3 schnell gelöst. Rückübersetzt in monomiale Matrizen ergibt sich die folgende Symmetriegruppe von Paaren (X, Y) mit der Eigenschaft $X \cdot t \cdot Y = t$:

$$G = \langle (\pi_T, \pi_T^{-1}), (\pi_R, \pi_R^{-1}), (\pi_C, \pi_C^{-1}), (\Leftrightarrow\!\!\!\pi_{AL}, \pi_{AR}), (\Leftrightarrow\!\mathbf{1}, \Leftrightarrow\!\mathbf{1}) \rangle.$$

Die in den Erzeugern von G auftretenden Permutationen sind explizit

$$\begin{aligned}
\pi_T &= (2,9,5,3)(4,10,13,7)(6,11)(8,12,14,15),\\
\pi_R &= (2,9)(3,5)(4,13)(6,11)(8,15)(12,14),\\
\pi_C &= (1,16)(2,15)(3,14)(4,13)(5,12)(6,11)(7,10)(8,9),\\
\pi_{AL} &= (1,11)(2,12)(3,9)(4,10)(5,15)(6,16)(7,13)(8,14),\\
\pi_{AR} &= (1,6)(2,5)(3,8)(4,7)(9,14)(10,13)(11,16)(12,15).
\end{aligned}$$

Als Symmetrieoperationen haben sie eine physikalische Bedeutung:

- T ist eine zyklische Verschiebung (engl. translation) der Spalte $\mid \alpha_1 \cdots \alpha_L \rangle$ in Richtung kleinerer Orte. Zum Beispiel wird $4 = |\uparrow\uparrow\downarrow\downarrow\rangle$ durch π_T in $|\uparrow\downarrow\downarrow\uparrow\rangle = 10$ verschoben.

- R ist die Spiegelung (engl. reflection) der Spalte α. Zum Beispiel wird $8 = |\uparrow\uparrow\uparrow\downarrow\rangle$ durch π_R zu $|\downarrow\uparrow\uparrow\uparrow\rangle = 15$ gespiegelt. Die Gruppe $\langle \pi_T, \pi_R \rangle$ ist die räumliche Symmetrie des Gitters; es ist genau die Automorphismengruppe der Spalte $\mid \alpha\rangle$. Es handelt sich dabei um eine Diedergruppe mit acht Elementen.

- C ist die Farbsymmetrie. Sie vertauscht die beiden möglichen Spin-Werte $\uparrow$ und $\downarrow$. Zum Beispiel wird $8 = |\uparrow\uparrow\uparrow\downarrow\rangle$ durch π_C auf $|\downarrow\downarrow\downarrow\uparrow\rangle = 9$ abgebildet.

- A ist eine Ferro/Antiferro-Symmetrie. Sie vertauscht eine ferromagnetische Phase (alle Spins in die gleiche Richtung) mit einer antiferromagnetischen Phase (alternierende Spins). Namentlich negiert π_{AR} den 1. und 3. Spin der Spalte α. Zum Beispiel wird $1 = |\downarrow\downarrow\downarrow\downarrow\rangle$ von AL auf $|\uparrow\downarrow\uparrow\downarrow\rangle = 6$ abgebildet. Diese Symmetrie existiert nur bei geradem L. Das Paar $(\Leftrightarrow\!\!\!\pi_{AL}, \pi_{AR})$ ist echt monomial. Da t die Exponenten der Transfermatrix sind, wird bei dieser Symmetrieoperation $w = \exp(J/(k_{\mathrm{B}}T))$ durch w^{-1} ersetzt. Dies paßt zur physikalischen Vorstellung: Wird die Kopplungskonstante J negiert, so wird aus einer ferromagnetischen Ordnung eine antiferromagnetische Ordnung und umgekehrt.

$\Leftrightarrow\mathbf{1}$ ist eine triviale Symmetrie. Sie liegt im Zentrum der $\mathsf{GL}_{16}(\mathbb{C})$ und hängt nicht von t ab. Sie tritt in jeder monomialen Symmetrie auf.

Die gesamte gefundene Symmetrie G des $(4 \times M)$-Ising-Modells war auch ohne automatische Symmetriesuche bekannt. Sie wurde von H. Meyer in seiner Disseration zur Diagonalisierung der Transfermatrix verwendet. Als neues Ergebnis ergibt sich hier jedoch, daß es auch keine weitere Symmetrie vom Mon-Mon-Typ gibt:

8.1 Satz *Sei* $(X, Y) \in \mathsf{GL}_{16}(\mathbb{C}) \times \mathsf{GL}_{16}(\mathbb{C})$ *ein Paar monomialen Matrizen mit der Eigenschaft*

$$X \cdot t \cdot Y = t.$$

Dann ist $(X, Y) \in G$. *(t ist wie im letzten Abschnitt, G wie in diesem.)*

Da das $(4 \times M)$-Ising-Modell ohne äußeres Feld mit Bethe-Ansatz integrabel ist, existiert weitere Symmetrie. Sie ist nach dem vorigen Satz keine Mon-Mon-Symmetrie der Exponenten der symmetrisierten Transfermatrix. Für die Physik stellt die eben getroffene Aussage nur einen recht eingeschränkten Spezialfall dar. Sie zeigt aber die Art der Aussagen, die durch die Symmetriesuche möglich werden. Die Symmetriesuchverfahren können daher nützlich sein, um sich einer Klasse von Spin-Gitter-Modellen experimentell zu nähern.

8.5 Ergebnisse weiterer Modelle

Nach der Einführung in die Symmetriebestimmung am Beispiel des $(4 \times M)$-Ising-Modells folgt nun eine Darstellung einiger massiverer Rechnungen. In diesem Abschnitt wird von den Ergebnissen der Symmetriebestimmung für einige Spin-Gitter-Modelle und einige Größen berichtet. Die verwendete Notation für Gruppen und deren Darstellungen wird in Anhang A erklärt.

$(L \times M)$-Ising-Modell ohne äußeres Feld Von der Mon-Mon-Symmetrie G_L des Ising-Modells ist eine Untergruppe B_L bekannt. Sie wird erzeugt durch die im letzten Abschnitt besprochenen speziellen Symmetrieoperationen. Diese sollten hier mit T, R, C, A und $\Leftrightarrow\mathbf{1}$ bezeichnet werden. Es ist

$$B_L = \begin{cases} \langle \Leftrightarrow\mathbf{1}; C; R, T \rangle & \text{für } L \text{ ungerade,} \\ \langle \Leftrightarrow\mathbf{1}; C; R, T, A \rangle & \text{für } L \text{ gerade.} \end{cases}$$

Die Ordnung der Erzeuger ist durch $\Leftrightarrow\mathbf{1}^2 = C^2 = R^2 = T^L = A^2 = \mathrm{id}$ gegeben. Außerdem erfüllen die Erzeuger die Relationen $(x^y = y^{-1}xy)$:

$$\begin{array}{rl} \Leftrightarrow\mathbf{1} & \text{kommutiert mit } C, R, T, A, \\ A & \text{operiert wie } C^A = C,\ R^A = CR,\ T^A = CT, \\ C & \text{kommutiert mit } R, T \text{ und} \\ R & \text{operiert wie } T^R = T^{-1}. \end{array}$$

Mit diesen Relationen ist der Isomorphietyp von B_L für alle L bekannt. In der folgenden Tabelle sind die mit den Methoden dieser Arbeit gefundenen Symmetriegruppen G_L sowie die direkt konstruierbaren Untergruppen B_L für die kleinsten L zusammengestellt.

L	$\lvert B_L\rvert$,	Nr.	$\lvert G_L\rvert$,	Nr.	G_L
1	4	2	8	3	D_8
2	16	11	384	5602	$\mathsf{Z}_2 \wr \mathsf{S}_4 = \mathsf{S}_4 \ltimes \mathsf{Z}_2^4$
3	24	14	24	14	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_6$
4	64	202	64	202	$\mathsf{Z}_2 \times (\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_8))$
5	40	13	40	13	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{10}$
6	96	209	96	209	$\mathsf{Z}_2 \times (\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_{12}))$
7	56	12	56	12	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{14}$
8	128	1728	128	1728	$\mathsf{Z}_2 \times (\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_{16}))$
9	72	17	72	17	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{18}$
10	160	217	160	217	$\mathsf{Z}_2 \times (\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_{20}))$

Diese Daten legen die folgende Vermutung nahe: *Die Mon-Mon-Symmetrie des Ising-Modells ist ab $L \geq 3$ genau die bekannte Symmetrie.* Mit dem Wissen über die Relationen der Erzeuger von B_L ergibt sich als Mon-Mon-Symmetrie des Ising-Modells

$$G_L = \begin{cases} G_1 & \cong & \mathsf{D}_8 & \text{für } L = 1 \\ G_2 & \cong & \mathsf{Z}_2 \wr \mathsf{S}_4 = \mathsf{S}_4 \ltimes \mathsf{Z}_2^4 & \text{für } L = 2 \\ \langle \Leftrightarrow\!\mathbf{1}; C; R, T\rangle & \cong & \mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{2L} & \text{für } L \geq 3 \text{ ungerade} \\ \langle \Leftrightarrow\!\mathbf{1}; A; C; R, T\rangle & \cong & \mathsf{Z}_2 \times (\mathsf{Z}_2 \ltimes (\mathsf{Z}_2 \times \mathsf{D}_{2L})) & \text{für } L \geq 4 \text{ gerade.} \end{cases}$$

$(L \times M)$-Potts-Standardmodell mit q Farben Beim Potts-Standardmodell mit $q \geq 2$ Farben handelt es sich um eine Verallgemeinerung des Ising-Modells. Der Zustand jedes Teilsystems (jedes Spins) ist aus der Menge $\{1..q\}$, den sogenannten Farben. Die Wechselwirkung zwischen benachbarten Spins σ_i und σ_j ist die Kroneckerfunktion $\delta_{\sigma_i,\sigma_j}$. Für $q = 2$ erhält man im wesentlichen wieder das Ising-Modell, bei dem die Farben allerdings durch $\Leftrightarrow\!\mathbf{1}$ und 1 codiert waren. Da beim Potts-Modell alle Exponenten der Transfermatrixkomponenten nicht-negativ sind, kann die Ferro/Antiferro-Symmetrie so nicht definiert werden. Andererseits sind die Farben alle gleichberechtigt, so daß als Farbsymmetrie die volle S_q auftritt. Die Farbsymmetrie wird hier daher durch $C(q)$ bezeichnet. Die bekannte Untergruppe $B_{q,L}$ der Symmetriegruppe $G_{q,L}$ ist daher beim Potts-Modell

$$B_{q,L} = \langle \Leftrightarrow\!\mathbf{1}, C(q), R, T\rangle.$$

Die folgende Tabelle stellt das Ergebnis der Symmetriesuche für kleine Werte von q und L dar. Die Zahlen wurden so gewählt, daß die Anzahl $(2q^L)^2$ der

Matrixkomponenten 10^7 nicht überschreitet.

L	$q=2$	$q=3$	$q=4$	$q=5$	$q=6$	$q=7$
1	$\mathsf{Z}_2 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_3$	$\mathsf{Z}_2 \times \mathsf{S}_4$	$\mathsf{Z}_2 \times \mathsf{S}_5$	$\mathsf{Z}_2 \times \mathsf{S}_6$	$\mathsf{Z}_2 \times \mathsf{S}_7$
2	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_3 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_4 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_5 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_6 \times \mathsf{Z}_2$	$\mathsf{Z}_2 \times \mathsf{S}_7 \times \mathsf{Z}_2$
3	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_6$	$\mathsf{Z}_2 \times \mathsf{S}_3 \times \mathsf{D}_6$	$\mathsf{Z}_2 \times \mathsf{S}_4 \times \mathsf{D}_6$	$\mathsf{Z}_2 \times \mathsf{S}_5 \times \mathsf{D}_6$	$\mathsf{Z}_2 \times \mathsf{S}_6 \times \mathsf{D}_6$	$\mathsf{Z}_2 \times \mathsf{S}_7 \times \mathsf{D}_6$
4	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_8$	$\mathsf{Z}_2 \times \mathsf{S}_3 \times \mathsf{D}_8$	$\mathsf{Z}_2 \times \mathsf{S}_4 \times \mathsf{D}_8$	$\mathsf{Z}_2 \times \mathsf{S}_5 \times \mathsf{D}_8$	$\mathsf{Z}_2 \times \mathsf{S}_6 \times \mathsf{D}_8$	
5	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{10}$	$\mathsf{Z}_2 \times \mathsf{S}_3 \times \mathsf{D}_{10}$	$\mathsf{Z}_2 \times \mathsf{S}_4 \times \mathsf{D}_{10}$			
6	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{12}$	$\mathsf{Z}_2 \times \mathsf{S}_3 \times \mathsf{D}_{12}$				
7	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{14}$					
8	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{16}$					
9	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{18}$					
10	$\mathsf{Z}_2 \times \mathsf{Z}_2 \times \mathsf{D}_{20}$					

Werden die gefundenen Gruppen $G_{q,L}$ verglichen mit den bekannten Gruppen $B_{q,L}$, so gelangt man zu folgender Vermutung: *Die Mon-Mon-Symmetrie des Potts-Modells ist genau die bekannte Symmetrie.* Diese Symmetrie ist gegeben durch

$$G_{q,L} = \begin{cases} \langle \Leftrightarrow\mathbf{1}; C(q) \rangle & \cong & \mathsf{Z}_2 \times \mathsf{S}_q & \text{für } L = 1 \\ \langle \Leftrightarrow\mathbf{1}; C(q); R \rangle & \cong & \mathsf{Z}_2 \times \mathsf{S}_q \times \mathsf{Z}_2 & \text{für } L = 2 \\ \langle \Leftrightarrow\mathbf{1}; C(q); R, T \rangle & \cong & \mathsf{Z}_2 \times \mathsf{S}_q \times \mathsf{D}_{2L} & \text{für } L \geq 3. \end{cases}$$

8.6 Zusammenfassung

Eine wichtige Frage der statistischen Physik ist die Integrabilität von Spin-Gitter-Modellen. Diese hängt mit der Existenz einer ausreichend großen Symmetrie des Systems zusammen. Suchverfahren zur automatischen Bestimmung der Mon-Mon-Symmetrie der Transfermatrix liefern einen Beitrag zur Untersuchung eines Spin-Gitter-Modells. Die Kenntnis der Mon-Mon-Symmetrie erleichtert die Bestimmung der Eigenwerte der Transfermatrix erheblich. (Wie dies im einzelnen geschieht, ist hier nicht ausgeführt worden.)

Das $(L \times M)$-Ising-Modell ohne äußeres Feld und das $(L \times M)$-Potts-Modell mit q Farben ohne äußeres Feld wurden betrachtet für kleine Werte von L und q. Es zeigt sich, daß die in der Literatur bekannten Symmetrieoperationen die gesamte Mon-Mon-Symmetrie erzeugen, außer beim Ising-Modell für $L = 1$ und $L = 2$. Die Gruppenstruktur und die konkrete Darstellung der Symmetrie werden für den allgemeinen Fall als plausible Vermutung formuliert.

Anhang A

Notationen und Symbole

QI'yaH, ghuy'cha', Qu'vatlh	`*?!#@, *@$%, #*@!`
Map[Function[x,Part[x,1]],Out[-1]]	`#[[1]]&/@%`

Klingonische Flüche mit Übersetzung und ein Ausdruck in MATHEMATICA *mit seiner Operatornotation*[1]

IIESEM ANHANG sind die grundlegenden Notationen und Konventionen zusammengetragen. Sie werden in der gesamten Arbeit benötigt und sind daher an einer Stelle gesammelt worden. Dieser Anhang hat den Charakter eines Nachschlagewerks für Notationen. Alle hier erklärten Notationen sind im Stichwortverzeichnis aufgeführt.

Allgemeines Es werden die folgenden kompakten Notationen verwendet:

$$
\begin{aligned}
\{n..m\} &= \{n, n+1, \ldots, m\} \\
[n..m) &= \{n, n+1, \ldots, m \Leftrightarrow 1\} \quad \text{als Menge oder Folge} \\
x \operatorname{div} m &= \lfloor x/m \rfloor \quad \text{ganzzahliger Quotient} \\
x \bmod m &= x \Leftrightarrow m \cdot (x \operatorname{div} m) \quad \text{Divisionsrest in } [0..m) \\
2^M &= \{t \mid t \subseteq M\} \quad \text{die Potenzmenge von } M \\
\binom{N}{k} &= \{K \subseteq N \mid |K| = k\} \quad k\text{-Teilmengen von } N.
\end{aligned}
$$

Die Notationen für diskrete Strukturen folgen im wesentlichen dem Buch von Graham, Knuth und Patashnik, [38]. Für die Konzepte der aufzählenden Kombinatorik ist das Buch von Kerber (1991), [50], nützlich. Das Buch von Cohen (1991), [20], ist eine Fundgrube für zahlentheoretische Algorithmen. Für spezielle Funktionen der Mathematik oder Physik wird das Referenzwerk von Abramowitz und Stegun (1970), [1], verwendet.

[1] Aus „The Klingon Dictionary, English/Klingon, Klingon/English", [69], Kapitel 5.5; MATHEMATICA ist ein Warenzeichen von Wolfram Research Inc.

Körper Das Symbol $\mathbb{F}_{p^e}$ bezeichnet den endlichen Körper mit p^e Elementen für eine Primzahl p und $e \geq 1$. Ein umfassendes Werk über endliche Körper ist das Buch [57] von Lidl und Niederreiter (1983). Für die Zwecke dieser Arbeit reicht eine Einführung aus, wie zum Beispiel in [52], Kapitel VIII, §1–3, von S. Lang.

Mit $\mathbb{Q}(\zeta_n)$ wird der Kreisteilungskörper der Ordnung $n \geq 3$ bezeichnet; dies ist der Zerfällungskörper von $X^n \Leftrightarrow 1$ über den rationalen Zahlen. Dabei bezeichnet ζ_n eine primitive n-te Einheitswurzel. Wird $\mathbb{Q}(\zeta_n)$ aufgefaßt als Teilkörper der komplexen Zahlen $\mathbb{C}$, so ist die Einheitswurzel von der Form $\zeta_n = \exp(2\pi \mathrm{i}\, k/n)$, für ein $k \in [0..n)$ mit $\gcd(k, n) = 1$. Eine Einführung in Kreisteilungskörper findet sich in [52], Kapitel VIII, §4–5, von S. Lang.

Ist L/K eine algebraische Körpererweiterung von K, so bezeichnet $[L : K]$ die Dimension von L, aufgefaßt als K-Vektorraum. Ist L/K galoisch, so bezeichnet $\mathsf{Gal}(L/K)$ die Menge aller Körperautomorphismen γ von L, die K punktweise stabilisieren. Das Bild eines Elements $\alpha \in L$ unter γ wird als α^γ geschrieben. Alle in dieser Arbeit benötigten Ergebnisse zur Theorie der Körpererweiterungen finden sich in [52], Kapitel VII.

Verbände Ein Verband ist eine partiell geordnete Menge, in der je zwei Elemente eine größte untere und eine kleinste obere Schranke besitzen (nach Jacobson [45], Defs. 8.1 und 8.2). Im allgemeinen wird die Ordnungsrelation mit $\sqsubseteq$, die größte untere Schranke (engl. meet) mit $\sqcap$ und die kleinste obere Schranke (engl. join) mit $\sqcup$ bezeichnet.

Von besonderem Interesse für diese Arbeit ist der Verband $\mathrm{Part}(M)$ aller Partitionen der endlichen Menge M (nach Jacobson [45], Ch. 8.6, Ex. 8). Die Partitionen von M sind durch die Verfeinerungsrelation $\sqsubseteq$ partiell geordnet. Beispiel: $(1|2\,3\,5|4) \leq (1\,4|2\,3\,5)$. Die größte untere Schranke von p und q wird $p \sqcap q$ und die kleinste obere mit $p \sqcup q$ bezeichnet.

Der andere für diese Arbeit wesentliche Verband ist der Untergruppenverband $\mathrm{Subgrp}(G)$ einer endlichen Gruppe G. Er ist durch die Untergruppenrelation $\leq$ partiell geordnet. Die größte untere Schranke von H und K ist $H \cap K$, die kleinste obere Schranke $\langle H \cup K \rangle$.

Abstrakte Gruppen Mit D_{2n} wird eine Diedergruppe mit $2n$ Elementen bezeichnet, mit Z_n eine zyklische Gruppe und mit E die triviale. Das Symbol $H \ltimes N$ bezeichnet das semidirekte Produkt mit dem Normalteiler N im Sinne von Huppert ([43], Bd. I, Kap. I, §14). Dies ist die zerfallende Gruppenerweiterung; sie wird in der Literatur meist als $N : H$ notiert (etwa im Atlas of Finite Groups, [21]). Das Symbol $G \wr \mathsf{S}_n$ bezeichnet das Kranzprodukt der Ordnung $|G|^n n!$ im Sinne von Huppert ([43], Bd. I, Kap. I, §15) oder A. Kerber ([50], sect. 1.2). Es ist von der Form $\mathsf{S}_n \ltimes G^n$. Das subdirekte Produkt mit vereinigter Faktorgruppe im Sinne von Huppert ([43], Bd. I, Kap. I, §9) wird mit $G \curlywedge H$ bezeichnet. Es wird in Definition 1.3 auf Seite 9 eingeführt und dort auch erläutert.

Für Gruppen bis zur Ordnung 1000, außer den Ordnungen 512 und 768, steht der neue Katalog der kleinen endlichen Gruppen [9] von H. U. Besche und B. Eick (1996) zur Verfügung. Dieser Katalog kann in das GAP-System, [82], geladen werden; er ist frei verfügbar. Eine Gruppe des Katalogs wird durch ihre Ordnung N und eine laufende Nummer K bezeichnet. Eine Ag- oder Permutations-Darstellung zu dem entsprechenden Isomorphietyp kann im GAP-System mit der Funktion `SmallGroup`(N, K) konstruiert werden. Die Funktion `IdGroup`(G) bestimmt zu einer Ag- oder Permutations-Darstellung G das Paar $[N, K]$.

Soll nicht nur der Isomorphietyp einer Gruppe angegeben werden, sondern auch ihre Darstellung, dann erfolgt dies durch eine Gleichsetzung der Form $\langle g_1, g_2, \dots \rangle$ = Isomorphietyp, wobei die Erzeuger durch Semikolons so gruppiert werden, wie es der Isomorphietyp angibt. Zum Beispiel: $\langle x; r, t \rangle = \mathsf{Z}_2 \times \mathsf{D}_8$ bedeutet, x ist der Erzeuger des direkten Faktors Z_2 und r, t sind Standarderzeuger der Diedergruppe. Als Standarderzeuger der Diedergruppe werden $\langle r, t \rangle = D_{2n}$ mit den Relationen $r^2 = t^n = rtrt = \mathrm{id}$ gewählt.

Ein Standardwerk über endliche Gruppen sind die Bücher von Huppert, [43]. Für diese Arbeit ist hauptsächlich Band I von Bedeutung. Als Einführung ist besonders das schöne Buch von M. Hall (1976) geeignet, [39]. Für die Computeralgebra von Gruppen ist die Sammlung von Atkinson (1984), [4], ein Klassiker. Seit 1984 sind jedoch etliche neue Methoden hinzugekommen. Für eine neuere und allgemeinere Übersicht der Computeralgebra sei auf den Bericht der GI, DMV und GAMM (1993) verwiesen, [31].

Permutationsgruppen Eine symmetrische Gruppe auf n Punkten, $n \geq 1$, wird mit S_n bezeichnet, eine alternierende mit A_n. Die Identitätspermutation wird mit id bezeichnet. *Das Bild des Punktes i unter der Permutation S wird in dieser Arbeit als i^S notiert.* Diese Konvention ist der Standard in der Gruppentheorie und wird auch im GAP-System, [82], verwendet. Zu dieser Konvention passen die Festlegungen

$$\begin{aligned} S^T &= T^{-1} \cdot S \cdot T \quad \text{die Konjugation von } S \text{ mit } T \\ [S, T] &= S^{-1} \cdot T^{-1} \cdot S \cdot T \quad \text{der Kommutator von } S \text{ und } T \\ \Pi_n(S) &= [\delta_{i^S, j} \mid i, j] \in \{0, 1\}^{n \times n} \quad \text{die Permutationsmatrix zu } S, \end{aligned}$$

wobei $S, T \in \mathsf{S}_n$ sind. Mit $\delta_{i,j}$ wird das Kronecker-Delta bezeichnet; es ist 1 für $i = j$ und ansonsten 0. *In dieser Arbeit wird eine Permutationsmatrix $\Pi_n(S)$ immer mit der Permutation S identifiziert.* Die wichtigste Formel in diesem Zusammenhang ist die Operation von Permutation(smatriz)en von links und von rechts auf einer beliebigen Matrix. Sei $M \in \mathrm{K}^{n \times m}$ sowie $L \in \mathsf{S}_n$ und $R \in \mathsf{S}_m$. Dann gilt

$$L \cdot M \cdot R = [\, M_{i^L, j^{R^{-1}}} \mid i, j \,]. \tag{A.1}$$

Die Young-Gruppe einer Partition $p = \{b_1, \ldots, b_r\}$ von $\{1..n\}$ ist die Permutationsgruppe $\mathsf{S}(p) = \mathsf{S}(b_1) \times \cdots \times \mathsf{S}(b_r)$. Die Young-Gruppe enthält genau die Permutationen aus S_n, die die Partition p stabilisieren. Für eine Young-Gruppe kann direkt ein starkes Erzeugendensystem mit Basis angegeben werden, denn $\{(1,n),(2,n),\ldots,(n \Leftrightarrow 1,n)\}$ ist ein starkes Erzeugendensystem der S_n mit Basis $\{1..n \Leftrightarrow 1\}$.

Ein neues Werk zur Theorie der Permutationsgruppen (auch der unendlichen) ist das Buch von Dixon und Mortimer (1996), [28]. Das Buch enthält im Anhang die Liste der primitiven Permutationsgruppen auf bis zu 1000 Punkten. Ein Standardwerk über die grundlegenden algorithmischen Verfahren für Permutationsgruppen ist das Buch von Butler (1991), [16]. In diesem Buch wird unter anderem das von Sims eingeführte Konzept der Stabilisatorkette einer Permutationsoperation erklärt. Klassische Texte über Permutationsgruppen sind das kompakte Buch von Wielandt (1964), [92], und das Buch von Passmann (1968), [71]. Für transitive Permutationsgruppen sei auf die kürzlich fertiggestellte Dissertation von A. Hulpke (1996) in Aachen verwiesen, [42]. Hulpke hat die transitiven Permutationsgruppen auf bis zu 31 Punkten konstruiert.

Matrizen Matrizen werden in dieser Arbeit mit $\{1..n\}$ oder mit $[0..n)$ indiziert, je nach Anwendung. Die grundlegende Notation um Matrizen anzugeben, ist die Komprehension. Sie faßt Komponenten $m(i,j)$ zu einer Matrix M zusammen: $M = [\, m(i,j) \mid i \in R, j \in C \,]$. Die Indexmengen R und C werden meist weggelassen.

Die (i,j)-Komponente von M wird als $M_{i,j}$ geschrieben. Die i-te Zeile von M wird mit dem „Wildcard"-Symbol $*$ als $M_{i,*}$ bezeichnet. Analog ist $M_{*,j}$ die j-te Spalte. Es wird außerdem eine kompakte Notation für Untermatrizen verwendet: Ist $I \subseteq R$ eine Teilmenge der Zeilenindizes, so bezeichnet $M_{I,*}$ die Untermatrix aus den Zeilen $\{M_{i,*} \mid i \in I\}$. Die Reihenfolge der Zeilen bleibt erhalten. Analog ist $M_{*,J}$ für $J \subseteq C$ eine Untermatrix aus Spalten von M. Darüber hinaus werden folgende Notationen verwendet:

$$
\begin{aligned}
\mathbf{1}_n &= \mathbf{1}_{n\times n} \quad \text{die } (n \times n)\text{-Einheitsmatrix} \\
\mathbf{1}_{n\times m} &= [\, \delta_{i,j} \mid i \in [0..n), j \in [0..m) \,] \\
\mathbf{J}_{n\times m} &= [\, 1 \mid i \in [0..n), j \in [0..m) \,] \quad \text{die „all-one"-Matrix} \\
\operatorname{diag}(x_k \mid k \in [0..n)) &= [\, x_i \cdot \delta_{i,j} \mid i,j \in [0..n) \,] \quad \text{eine Diagonalmatrix} \\
\operatorname{circ}(x_k \mid k \in [0..n)) &= \left[x_{(j-i) \bmod n} \,\middle|\, i,j \in [0..n) \right] \quad \text{eine zirkulante Matrix} \\
\operatorname{tr}(A) &= \sum_i A_{i,i} \quad \text{die Spur von } A \\
A^{\mathsf{T}} &= [\, A_{j,i} \mid i,j \,] \quad \text{die Transponierte von } A \\
A \oplus B &= \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix} \quad \text{direkte Summe von } A \text{ und } B
\end{aligned}
$$

$$\begin{aligned} A \otimes B &= \left[A_{i \operatorname{div} m,\, j \operatorname{div} m} \cdot B_{i \bmod m,\, j \bmod m} \;\middle|\; i, j \in [0..nm) \right] \\ \mathsf{GL}_n(\mathrm{K}) &= \{ M \in \mathrm{K}^{n \times n} \mid M \text{ invertierbar} \}. \end{aligned}$$

Im Kroneckerprodukt $A \otimes B$ bestimmt A die grobe Struktur und B die feine. Zum Beispiel

$$\mathbf{1}_2 \otimes \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \mathbf{1}_3 = \begin{bmatrix} 1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 3 & \cdot & \cdot & 4 \end{bmatrix}.$$

Darstellungen von Gruppen Eine Matrix-Darstellung ρ einer endlichen Gruppe G ist ein Gruppen-Homomorphismus von G in die Gruppe $\mathsf{GL}_n(\mathrm{K})$ der invertierbaren $(n \times n)$-Matrizen über dem Körper K. Die Zahl n heißt der Grad von ρ und wird mit $\deg(\rho)$ bezeichnet. Die Verkettung $g \mapsto \mathrm{tr}(\rho(g))$ von ρ mit der Spurabbildung tr heißt Charakter von ρ. Im folgenden sind die für diese Arbeit wesentlichen Konstruktionen von Darstellungen zusammengetragen.

- $\rho^A = g \mapsto A^{-1} \cdot \rho(g) \cdot A$ ist die mit $A \in \mathsf{GL}_n(\mathrm{K})$ konjugierte Darstellung.
- $\rho_G \oplus \varphi_G = g \mapsto \rho_G(g) \oplus \varphi_G(g)$ ist die direkte Summe der Darstellungen ρ_G und φ_G derselben Gruppe G.
- $\rho_G \otimes \varphi_G = g \mapsto \rho_G(g) \otimes \varphi_G(g)$ ist das innere Tensorprodukt der Darstellungen ρ_G und φ_G derselben Gruppe G. Das innere Tensorprodukt ist wieder eine Darstellung der Gruppe G. Das innere Tensorprodukt sollte nicht verwechselt werden mit dem wichtigeren äußeren Tensorprodukt:
- $\rho_G \# \varphi_H = (g, h) \mapsto \rho_G(g) \otimes \varphi_H(h)$ ist das äußere Tensorprodukt der Darstellungen ρ_G von G und φ_H von H. Das äußere Tensorprodukt ist eine Darstellung von $G \times H$.
- $\lambda_G \cdot \rho_G = g \mapsto \lambda_G(g) \cdot \rho_G(g)$ ist das lineare Vielfache der Darstellung ρ_G mit der Darstellung λ_G vom Grad Eins. (λ_G wird in der Literatur manchmal als „lineare Darstellung“ bezeichnet. Um Mißverständnisse zu reduzieren, wird diese Bezeichnung hier vermieden.)
- $\rho^\gamma = g \mapsto \rho(g)^\gamma$ ist die mit γ Galois-konjugierte Darstellung ρ. Die Konjugation γ ist ein Automorphismus des zugrunde liegenden Körpers K. Er wird auf jede Komponente der Matrix $\rho(g)$ angewandt.
- $\rho^\alpha = g \mapsto \rho(g^\alpha)$ ist die verkettete Darstellung von ρ mit α. Dabei ist α ein Isomorphismus von G nach G^α und ρ eine Darstellung von G^α.

- $\rho_G \downarrow H = h \mapsto \rho_G(h)$ ist die Restriktion der Darstellung ρ_G von G auf die Untergruppe $H \leq G$.

- $\rho_H \uparrow_T G$ bezeichnet die von ρ_H bezüglich T induzierte Darstellung auf G. Dabei ist H eine Untergruppe von G und $T = [t_1, \ldots, t_{|G/H|}]$ ist eine Transversale der Nebenklassenmenge G/H. Die Darstellung $\rho_H \uparrow_T G$ ist eine Darstellung von G vom Grad $|G/H| \cdot \deg(\rho_H)$. Sie ist definiert durch $|G/H|$-Matrizen von $\deg(\rho_H)$-Matrizen in der Form

$$\rho_H \uparrow_T G \;=\; g \mapsto \left[\dot{\rho}_H(t_i \cdot g \cdot t_j^{-1}) \;\middle|\; i,j \in \{1..|G/H|\}\right] \quad \text{mit}$$
$$\dot{\rho}_H(x) = \begin{cases} \rho_H(x) & x \in H \\ \mathbf{0} & \text{sonst.} \end{cases}$$

Die Darstellung ρ heißt Permutationsdarstellung, wenn $\rho(g)$ für alle $g \in G$ eine Permutationsmatrix ist. Die Darstellung ρ heißt monomiale Darstellung, wenn $\rho(g)$ für alle $g \in G$ eine monomiale Matrix ist. Die Darstellung ρ heißt induziert monomial, wenn ρ die Induktion einer linearen Darstellung einer Untergruppe ist.

In dieser Arbeit geht es ausschließlich um gewöhnliche Darstellungstheorie. Das heißt, $|G|$ wird von der Charakteristik des Grundkörpers nicht geteilt (die Bedingung von Maschke). In diesem Fall ist jede Darstellung konjugiert zu einer direkten Summe von irreduziblen Darstellungen (Satz von Maschke).

Ein Standardwerk über Darstellungstheorie endlicher Gruppen ist das Buch von Curtis und Reiner (1962/1981), [24], [25]. Für die Zwecke dieser Arbeit sind jedoch einführende Texte ausreichend, wie der von James und Liebeck (1993), [47]. Eine kompakte Einführung mit dem Blick auf das Wesentliche ist der Anfang des Buchs von Serre (1977), [83]. Eine sehr schöne Einführung in die Darstellungstheorie ist zudem das Skriptum [56] von H. W. Leopoldt (1979).

Literaturverzeichnis

[1] Abramowitz, M., und Stegun, I. A. *Handbook of Mathematical Functions*, 9 Aufl. Dover Publications, New York, 1970.

[2] Agarwal, R. C., und Cooley, J. W. New Algorithms for Digital Convolution. *IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-25* (1977), 392–410.

[3] Apple, G., und Wintz, P. Calculation of Fouriertransforms on Finite Abelian Groups. *IEEE Trans. Inform. Theory IT-16* (1970), 233–236.

[4] Atkinson, M. D. *Computational Group Theory.* Academic Press, 1984.

[5] Baum, U. Schnelle Algorithmen zur Spektraltransformation endlicher Gruppen (Dipl.-Arbeit), 1988.

[6] Beals, R. Quantum Computation of Fourier Transforms over Symmetric Groups. *Symposium on Theory of Computing, El Paso* (1997).

[7] Ben-Israel, A., und Greville, T. N. E. *Generalized Inverses: Theory and Applications.* Pure and applied mathematics. J. Wiley & Sons, 1974. ISBN 0-471-06577-3.

[8] Berlekamp, E. R., McEliece, R. J., und van Tilborg, H. C. A. On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory 24* (1978), 384–386.

[9] Besche, H. U., und Eick, B. The Groups of Order up to 1000, except 512 and 768. *ftp.math.rwth-aachen.de:/pub/incoming/grpcat* (1996).

[10] Beth, T. *Methoden der Schnellen Fouriertransformation.* Teubner Verlag, 1984.

[11] Beth, T. On the Computational Complexity of the General Discrete Fourier Transform. *Theoretical Computer Science 51* (1987), 331–339.

[12] Beth, T., und Hatz, V. *Computer Algebra, Skriptum zur Vorlesung.* Univ. Karlsruhe, 1988.

[13] Beth, T., Jungnickel, D., und Lenz, H. *Design Theory.* BI-Wiss.-Verl., 1985.

[14] Bluestein, L. I. A Linear Filtering Approach to the Computation of the Discrete Fourier Transform. *IEEE Trans. AU-18 18* (1970), 451–455.

[15] Burnside, W. On The Outer Isomorphisms of a Group. *Proc. Lon. Math. Soc. 11* (1913), 40–42.

[16] Butler, G. *Fundamental Algorithms for Permutation Groups.* Lecture Notes in Computer Science, 559. Springer, 1991.

[17] Clausen, M. *Beiträge zum Entwurf schneller Spektraltransformationen (Habilitationsschrift).* Univ. Karlsruhe, 1988.

[18] Clausen, M. Fast Generalized Fourier Transform. *Theoretical Computer Science 67* (1989), 55–63.

[19] Clausen, M., und Baum, U. *Fast Fourier Transforms.* BI-Wiss.-Verl., 1993.

[20] Cohen, H. *A Course in Computational Algebraic Number Theory.* Springer, 1991.

[21] Conway, J. H., Curtis, R. T., Norton, S. P., und Wilson, R. A. *Atlas of Finite Groups.* Clarendon Press, Oxford, 1985.

[22] Cooley, J. W., und Tukey, J. W. An Algorithm for the Machine Computation of Complex Fourier Series. *Mathematics of Computation 19* (1965), 297–301.

[23] Coppersmith, D. An Approximate Fourier Transform Useful in Quantum Factoring. Tech. Rep. RC 19642, IBM Research Division, Yorktown Heights NY, 7. December 1994.

[24] Curtis, W. C., und I., R. *Representation Theory of Finite Groups.* J. Wiley & Sons, 1962.

[25] Curtis, W. C., und I., R. *Representation Theory of Finite Groups*, Vol. I. J. Wiley & Sons, 1981.

[26] Diaconis, P., und Rockmore, D. Efficient Computation of the Fourier Transform on Finite Groups. *Journal of the AMS 3* (1990).

[27] Diaconis, P., und Rockmore, D. Efficient Computation of Isotypic Projections for the Symmetric Group. *DIMACS Series of Discrete Mathematics and Theoretical Computer Science 11* (1993), 87–104.

[28] Dixon, J. D., und Mortimer, B. *Permutation Groups.* Springer, 1996.

[29] Dornhoff, L. *Group Representation Theory.* Pure and Applied Mathematics. Dekker New York, 1971.

[30] Egner, S. Analyse und Synthese von Bewegungsvorgängen an einer Stewart-Plattform (Dipl.-Arbeit), 1994.

[31] Engeler, E., Hrsg. *Computer Algebra in Deutschland.* Gesellschaft für Informatik Deutsche Mathematiker Vereinigung und Gesellschaft für angewandte Mathematik und Mechanik, 1993.

[32] Fino, B. J., und Algazi, V. R. A Unified Treatment of Discrete Fast Unitary Transforms. *SIAM COMPUT. 6* (1977).

[33] Fulton, W., und Harris, J. *Representation Theory.* Springer, 1991.

[34] Gantmacher, F. R. *Matrizentheorie.* Springer, 1986.

[35] Garey, M. R., und Johnson, D. S. *Computers and Intractability.* W. H. Freeman and Co., 1979.

[36] Golub, G. H., und Van Loan, C. F. *Matrix Computations*, 2nd Aufl. Johns Hopkins Univ. Press, 1989.

[37] Good, I. J. The Relationship Between Two Fast Fourier Transforms. *IEEE Transactions on Computers C-20* (1971), 310–317.

[38] Graham, R. L., Knuth, D. E., und Patashnik, O. *Concrete Mathematics.* Addison-Wesley, 1992.

[39] Hall jr., M. *The Theory of Groups.* Chelsea Publ., 1976.

[40] Hegland, M., und Wheeler, W. W. Linear Bijections and the Fast Fourier Transform. *Applicable Algebra in Engineering, Communication and Computing 8* (1997), 143–163.

[41] Holt, D. F. The Computation of Normalizers in Permutation Groups. *Journal of Symbolic Computation 12* (1991), 499–516.

[42] Hulpke, A. *Konstruktion Transitiver Permutationsgruppen.* Diss., RWTH Aachen, 1996.

[43] Huppert, B. *Endliche Gruppen*, Vol. I. Springer, 1983.

[44] Iben, H. K. *Tensorrechnung.* Teubner Verlag, 1995.

[45] Jacobson, N. *Basic Algebra*, Vol. I. W. H. Freeman and Co., 1985.

[46] Jacobson, N. *Basic Algebra*, Vol. II. W. H. Freeman and Co., 1989.

[47] James, G., und Liebeck, M. *Representations and Characters of Groups.* Cambridge Univ. Press, 1993.

[48] James, G. D., und Kerber, A. *The Representation Theory of the Symmetric Group.* Addison-Wesley, 1981.

[49] Karpovsky, M. G., und Trachtenberg, E. A. Fast fourier transforms on finite non-abelian groups. *IEEE Trans. Comput.* (1977), 227–247.

[50] Kerber, A. *Algebraic Combinatorics Via Finite Group Actions.* BI-Wiss.-Verl., 1991.

[51] Korte, B., Lovász, L., und Schrader, R. *Geedoids.* Springer, 1991.

[52] Lang, S. *Undergraduate Algebra*, 2 Aufl. Springer, 1980.

[53] Laue, R. *Zur Konstruktion und Klassifikation endlicher auflösbarer Gruppen.* Bayreuther Mathematische Schriften. Bayreuth, 1982.

[54] Leedham-Green, C. R., Praeger, C. E., und Soicher, L. H. Computing with Group Homomorphisms. *Journal of Symbolic Computation 12* (1991), 527–532.

[55] Leon, S. J. Permutation Group Algorithms Based on Partitions, I: Theory and Algorithms. *Journal of Symbolic Computation 12* (1991), 533–583.

[56] Leopoldt, H. W. Darstellungstheorie endlicher Gruppen, Skriptum zur Vorlesung. Tech. Rep., Mathematik, Universität Karlsruhe, 1979.

[57] Lidl, R., und Niederreiter, H. *Finite Fields.* Addison-Wesley, 1983.

[58] Linton, S. A., Michler, G. O., und Olsson, J. Fast Fourier Transforms on Symmetric Groups. Tech. Rep., Institut für Experimentelle Mathematik, Universität GHS Essen, 1991.

[59] Linton, S. A., Michler, G. O., und Olsson, J. Fourier Transforms with Respect to Monomial Representations. *Mathematische Annalen 297* (1993), 253–268.

[60] Lipson, J. D. *Elements of Algebra and Algebraic Computing.* Benjamin/Cummings, 1981.

[61] Maslen, D. K., und Rockmore, D. Generalized FFTs — a Survey of Some Recent Results. *DIMACS Workshop in Groups and Computation* (1995).

[62] MEHTA, M. L. *Random Matrices and the Statistical Theory of Energy Levels*. Academic Press, 1967.

[63] MEYER, H. *Approches Numériques pour des Modèles de Physique Statistique*. Diss., Centre de Recherche sur les Trés Basses Températures (CRTBT), CNRS Grenoble, 1996.

[64] MINKWITZ, T. *Algorithmensynthese für lineare Systeme mit Symmetrie*. Diss., Univ. Karlsruhe, 1993.

[65] MINOUX, M. *Mathematical Programming. Theory and Applications*. J. Wiley & Sons, 1986.

[66] NÜCKEL, A. *Automatische Generierung von Schaltkreisen durch strukturelle Analyse algebraischer Systeme*. Diss., Univ. Karlsruhe, 1996.

[67] NUSSBAUMER, H. J. *Fast Fourier Transformation and Convolution Algorithms*, 2 Aufl. Springer, 1982.

[68] OBERST, U. Galois Theory and the Fast Gelfand Transform (FGT). Tech. Rep., Univ. Innsbruck, FB Mathematik, 1992.

[69] OKRAUD, M. *The Klingon Dictionary, English/Klingon, Klingon/English*. Pocket Books, New York, 1992.

[70] PAPOULIS, A. *Systems and Transforms with Applications in Optics*. McGraw-Hill, 1968.

[71] PASSMAN, D. *Permutation Groups*. Benjamin/Cummings, 1968.

[72] PICHLER, F. *Mathematische Systemtheorie; Dynamische Konstruktionen*. de Gruyter, Berlin, 1975.

[73] POHST, M., UND ZASSENHAUS, H. *Algorithmic Algebraic Number Theory*. Cambridge Univ. Press, 1989.

[74] POPPER, K. R. *Logik der Forschung*, 7. Aufl. J. C. B. Mohr, Tübingen, 1982.

[75] PRANGE, E. The Use of Information Sets in Decoding Cyclic Codes. *IRE Transactions on Information Theory IT-8* (September 1962).

[76] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., UND FLANNERY, B. P. *Numerical Recipes in C*, 2 Aufl. Cambridge Univ. Press, 1992.

[77] PUKELSHEIM, F. *Optimal Design of Experiments*. J. Wiley & Sons, 1993.

[78] Rader, C. M. Discrete Fourier Transforms When the Number of Data Samples is Prime. *Proceedings of the IEEE 56* (1968), 1107–1108.

[79] Rao, C. R., und Mitra, S. K. *Generalized Inverse of Matrices and its Applications.* J. Wiley & Sons, 1971.

[80] Reichl, L. E. *A Modern Course in Statistical Physics.* University of Texas Press, 1980.

[81] Rockmore, D. Efficient Computation of Fourier Inversion for Finite Groups. *Journal of the ACM 41*, 1 (1994), 31–66.

[82] Schönert, M., et al. GAP — Groups, Algorithms and Programming, v3.4.3. Tech. Rep., Lehrstuhl D für Mathematik, RWTH Aachen, 1995.

[83] Serre, J.-P. *Linear Representations of Finite Groups.* Springer, 1977.

[84] Shannon, C. E. A Symbolic Analysis of Relay and Switching Circuits. *Transactions of the IEEE 57* (1938).

[85] Shor, P. W. Algorithms for Quantum Computation: Discrete Logrithms and Factoring. *Foundations of Computer Science (FOCS)* (1994).

[86] Sturmfels, B. *Algorithms in Invariant Theory.* Springer, 1993.

[87] Thalmayr, A. *Das Wasserzeichen der Poesie.* Eichborn Verlag, Frankfurt, 1990.

[88] Thümmel, A. Computing Character Tables of p-Groups. In *Proceedings of ISSAC '96 (Zürich)* (1996), ACM.

[89] van Leeuwen, J., Hrsg. *Handbook of Theoretical Computer Science*, Vol. A/B. Elsevier, 1990.

[90] Walch, S. Die schnelle Gelfand- und Fouriertransformation Realisierung in AXIOM (Dipl.-Arbeit), 1994.

[91] Whitney, H. On the Abstract Properties of Linear Dependence. *American Journal of Mathematics 57* (1935), 509–533.

[92] Wielandt, H. *Finite Permutation Groups.* Academic Press, 1964.

[93] Winograd, S. Some Bilinear Forms Whose Multiplicative Complexity Depends on the Field of Constants. *Mathematical Systems Theory 10* (1977), 169–180.

[94] Winograd, S. On Computing the Discrete Fourier Transform. *Mathematics of Computation 32* (1978), 175–199.

Index

Zusammenfassung

In dieser Dissertationsschrift werden algorithmische Verfahren zur Bestimmung der Symmetrien einer gegebenen Matrix untersucht. Die betrachtete Form von Symmetrie wird beschrieben durch zwei Matrix-Darstellungen einer gemeinsamen endlichen Gruppe, in deren Intertwining-Raum die gegebene Matrix liegt. Es wurde von Minkwitz (1993) gezeigt, daß diese Form von Symmetrie die Konstruktion schneller Algorithmen für Signaltransformationen zu nicht-regulären, nichtabelschen Gruppen ermöglicht. Ein weiteres Anwendungsgebiet dieser Form von Symmetrie ist die Lösung bestimmter polynomialer Gleichungssysteme. Durch Egner (1993) wurde gezeigt, daß sich die direkten Kinematikgleichungen der 6/6-Stewart-Plattform durch eine lineare Transformation wesentlich vereinfachen lassen. Der tiefere Grund für diese Vereinfachung liegt in dem Vorhandensein einer großen Symmetrie von der hier betrachteten Form. Mit den in dieser Arbeit vorgestellten Verfahren kann die Symmetrie automatisch gefunden werden. Insbesondere gelang es, für den Perm-Block-Symmetrietyp ein in der Matrixgröße polynomiales Verfahren zu Symmetriebestimmung anzugeben, unter der Annahme einer oberen Schranke für die Größen der auftretenden Blöcke.

Abstract

In this Ph.D. thesis algorithmic methods to discover the symmetry of a given matrix are studied. The type of symmetry considered consists of two matrix representations of a common finite group such that the matrix is an element of the intertwining space of the representations. As Minkwitz (1993) has shown this type of symmetry can be used to construct signal transforms corresponding to non-regular non-abelian groups. Another field of application of the type of symmetry studied is the solution of certain polynomial systems of equations. It has been shown by Egner (1993) that the equations of the direct kinematics of the 6/6-Stewart platform can be simplified substantially by linear transformation. The deeper reason for this to be possible is the existence of a substantial symmetry of the form considered in this work. With the methods presented it is possible to obtain the symmetry automatically. In particular, an algorithm for the symmetry type Perm-Block is proposed which is polynomial time in the size of the matrix, for a fixed upper bound on the sizes of the resulting blocks.

Lebenslauf

5. April 1968	geboren in Hamburg
	Eltern: Dr. med. Ursula Egner, geb. Bohne und Dr. med. Wolfgang Dietrich Egner
1974 bis 1978	Besuch der Grundschule Alsterredder, Hamburg
1978 bis 1987	Gymnasium Oberalster, Hamburg Reifeprüfung: Juni 1987
1987 bis 1988	Wehrdienst in Essen und Glücksburg
1988 bis 1989	Informatikstudium an der Universität Fridericiana zu Karlsruhe Vordiplomprüfung: September 1989
1989 bis 1994	Physikstudium an der Universität Fridericiana zu Karlsruhe Diplomprüfung: September 1994
1991 bis 1995	Freiberufliche Tätigkeit am Zentrum für Kunst und Medientechnologie, Karlsruhe
seit Februar 1995	Stipendiat des Graduiertenkollegs „Beherrschbarkeit komplexer Systeme“ an der Fakultät für Informatik der Universität Fridericiana zu Karlsruhe