Numerische Mathematik 2, 280—292 (1960)

Untersuchungen des Untergruppenverbandes endlicher Gruppen auf einer programmgesteuerten elektronischen Dualmaschine

Von

J. NEUBÜSER

1. Problemstellung

Bei der Untersuchung gruppentheoretischer Probleme ist es oft nützlich, die Fragestellung an Hand eines größeren Beispielmaterials zu testen. Die genauere Diskussion spezieller Gruppen bedingt jedoch, wenn es sich nicht um einen wohlbekannten Typ, wie etwa abelsche Gruppen, handelt, einen unter Umständen recht erheblichen Rechenaufwand. Es liegt daher nahe zu versuchen, elektronische Rechenmaschinen zur Untersuchung und Diskussion speziell gegebener Gruppen zu verwenden.

In der Gruppentheorie liegen kaum systematische Methoden zur Strukturuntersuchung einer gegebenen Gruppe vor; häufiger benutzt werden nur einige Rechenverfahren wie etwa das der Restklassenabzählung [*1*], S. 12, für eine durch Erzeugende und definierende Relationen gegebene Gruppe, mit dem Ziel, eine Permutationsdarstellung für diese zu erhalten. Bei der Diskussion einer Gruppe macht man vielmehr gern Gebrauch von gruppentheoretischen Sätzen, die gerade auf die spezielle Gruppe anwendbar sind, um längere Rechnungen möglichst zu vermeiden. Die Benutzung einer elektronischen Rechenmaschine wird dagegen erst dann sinnvoll, wenn man ein systematisches Verfahren festlegt, welches auf beliebige Gruppen oder jedenfalls auf große Klassen von Gruppen anwendbar ist. In (durch die Speicherkapazität der Maschine) begrenztem Umfang können dabei Änderungen der Rechenmethode auf Grund schon bekannter Teilergebnisse unter Benutzung gruppentheoretischer Sätze durch Verzweigungen des Programms berücksichtigt werden.

Ein wichtiges Hilfsmittel, die Struktur einer endlichen Gruppe zu beschreiben, ist der Verband ihrer Untergruppen. Dieser kann durch die Eintragung der Klassen konjugierter Untergruppen und der Indizes der Untergruppen (vollständiger Situationsplan) ergänzt werden. Es ist zwar bekannt [*3*], daß es nichtisomorphe endliche Gruppen $\mathfrak{G}$ und $\mathfrak{H}$ gibt, deren Untergruppenverbände $V(\mathfrak{G})$ und $V(\mathfrak{H})$ durch einen Verbandsisomorphismus aufeinander abgebildet werden können, der Klassen konjugierter Untergruppen und Indizes erhält; doch sind diese Beispiele für kleinere Ordnungen recht selten. Die Bestimmung des vollständigen Situationsplanes ist um so mehr von Interesse, als man aus diesem auch wichtige charakteristische Untergruppen wie Kommutatorgruppe, Zentrum und Frattini-Gruppe ablesen kann.

Im folgenden soll ein System von Programmen beschrieben werden, das aus gegebenen Erzeugenden einer endlichen Gruppe nicht zu hoher Ordnung die obengenannten Größen für auflösbare Untergruppen berechnet und in übersichtlicher Form angibt. Die dabei benutzten gruppentheoretischen Sätze und Begriffe sind wohlbekannt und recht elementar, sie werden ohne Beweise im 2. Abschnitt zitiert, der zunächst eine allgemeine Beschreibung des Verfahrens gibt. Im 3. Abschnitt wird dann diskutiert, wie die speziellen Eigenschaften einer dualen Rechenmaschine mit hinreichend flexiblem Befehlscode zur möglichst zweckmäßigen Durchführung des Verfahrens benutzt werden können. Im 4. Abschnitt wird die Art der Ausgabe der Ergebnisse angegeben. Im 5. Abschnitt wird eine besonders schnell arbeitende Variante des Verfahrens für reguläre Permutationsgruppen beschrieben, während Abschnitt 6 über Anwendungsmöglichkeiten, bisherige und geplante Rechnungen mit den vorhandenen Programmen berichtet. In einem Anhang wird die benutzte Maschine, eine Z 22, kurz charakterisiert.

Die hier beschriebenen Programme wurden in der Zeit von April 1959 bis Februar 1960 am Rechenzentrum der Universität Kiel entwickelt, dessen Mitarbeitern, insbesondere Herrn Dr. B. SCHLENDER, ich für bereitwillig gewährten Rat und Unterstützung beim Programmieren herzlich danken möchte.

2. Das Rechenverfahren

Die Elemente einer zu untersuchenden Gruppe können in verschiedener Form gegeben sein, etwa

a) als Worte in abstrakten Erzeugenden, zu denen definierende Relationen vorliegen,

b) als Permutationen,

c) als Matrizen.

Von diesen Möglichkeiten erschien a) wenig geeignet, da es keinen Algorithmus gibt, um zu entscheiden, ob zwei formal verschiedene Worte das gleiche Gruppenelement darstellen; man müßte also spezielle Reduktionen für bestimmte Gruppenklassen einführen. Von den prinzipiell gleich geeigneten Möglichkeiten b) und c) wurde b) gewählt, da die Multiplikation zweier Permutationen weniger arithmetische Operationen benötigt und, jedenfalls bei regulären Darstellungen, die Anzahl der pro Gruppenelement zu speichernden Ziffern geringer ist; Rechenzeit und Speicherplatz unterliegen aber bei der Verwendung einer kleineren Maschine ziemlichen Beschränkungen.

Da jede endliche Gruppe einer Permutationsgruppe isomorph ist, bedingt diese Entscheidung keine Einschränkung bezüglich der zu untersuchenden Gruppen. Auch für die praktische Rechnung treten keine Schwierigkeiten auf, da für durch Erzeugende und definierende Relationen gegebene Gruppen im Anschluß an die hier beschriebenen Arbeiten bereits Programme [2] hergestellt sind, die mit dem oben erwähnten Verfahren der Restklassenabzählung Permutationsdarstellungen der gegebenen Gruppe nach beliebig vorgebbaren Untergruppen liefern.

Es seien also n Permutationen $G_1, \ldots, G_n$ vom Grad g vorgegeben. Die Untersuchung der von diesen erzeugten Gruppe $\mathfrak{G} = \{G_1, \ldots, G_n\}$ geschieht in mehreren größeren Schritten.

I. Die Gruppe $\mathfrak{G}$ wird erzeugt, d.h. es wird eine Liste aller Elemente von $\mathfrak{G}$ und ihrer Ordnungen hergestellt.

II. Die zyklischen Untergruppen von $\mathfrak{G}$ werden aufgesucht.

III. Die minimalen Untergruppen von $\mathfrak{G}$ werden in Klassen konjugierter Untergruppen geordnet.

IV. Der Halbverband aller auflösbaren Untergruppen von $\mathfrak{G}$ wird bestimmt.

Diese Schritte, denen vier größere Unterprogrammsysteme entsprechen, sollen nun im einzelnen beschrieben werden.

I. Um das Erzeugnis $\{G_1, \ldots, G_n\}$ zu erhalten, wird zunächst $\mathfrak{U}_0 = \{1\}$ und dann nacheinander $\mathfrak{U}_i = \{\mathfrak{U}_{i-1}, G_i\}$ für alle $i = 1, \ldots, n$ gebildet. Dabei erhält man das Erzeugnis $\{\mathfrak{U}, G\}$ einer Untergruppe $\mathfrak{U}$ und eines Elementes G folgendermaßen:

Die Elemente von $\mathfrak{U}$ seien, mit dem Einselement 1 beginnend, sonst beliebig, angeordnet:

$$U_1 = 1\,,\ U_2, \ldots, U_r.$$

Ist $\nu + 1$ die kleinste positive Zahl, für die $G^{\nu+1} \in \mathfrak{U}$, so wird $\mathfrak{V} = \mathfrak{U} + \mathfrak{U}G + \mathfrak{U}G^2 + \cdots + \mathfrak{U}G^\nu$ gebildet und der Anordnung von $\mathfrak{U}$ entsprechend geordnet:

$$V_1 = U_1, \ldots, V_r = U_r,\ V_{r+1} = U_1 G, \ldots, V_{2r} = U_r G, \ldots, V_{\nu r} = U_r G^\nu.$$

Dann werden die Elemente von $\mathfrak{V}$ der Reihe nach von links mit G multipliziert: $GV_1, GV_2, \ldots$. Nach jeder Multiplikation wird geprüft, ob $GV_j \in \mathfrak{V}$. Ist dies der Fall, so wird als nächstes GV_{j+1} untersucht. Ist dagegen $G^* = GV_j \notin \mathfrak{V}$ und $\nu_1 + 1$ die kleinste positive Zahl, für die $G^{*\nu_1+1} \in \mathfrak{V}$, so werden zu $\mathfrak{V}$ die neuen Restklassen $\mathfrak{U}G^*, \mathfrak{U}G^{*2}, \ldots, \mathfrak{U}G^{*\nu_1}$ hinzugefügt und ihre Elemente im Anschluß an die schon vorhandenen geordnet:

$$V_{\nu r+1} = U_1 G^*, \ldots, V_{(\nu+1)r} = U_r G^*,\ V_{(\nu+1)r+1} = U_1 G^{*2}, \ldots, V_{(\nu+\nu_1)r} = U_r G^{*\nu_1}.$$

Danach wird die Linksmultiplikation mit G bei V_{j+1} wieder begonnen und das Verfahren entsprechend fortgesetzt. Man gelangt zu einem Ende, wenn bei der Linksmultiplikation das letzte Element von $\mathfrak{V}$ erreicht wird, ohne daß Elemente $GV_j \notin \mathfrak{V}$ gefunden werden.

Auf diese Weise wird jedes Element von $\{\mathfrak{U}, G\}$ genau einmal erhalten: Die Menge $\mathfrak{V}$ besteht aus vollständigen Restklassen nach $\mathfrak{U}$, keine zwei dieser Restklassen können übereinstimmen, da vor der Bildung einer jeden Restklasse geprüft wird, ob ihr Repräsentant bereits in einer anderen Restklasse vorkommt. Ferner ist nach Konstruktion mit jedem Element $V \in \mathfrak{V}$ auch $GV \in \mathfrak{V}$. Daher erhält man durch Induktion nach der Länge l der Worte $U_1 G^{\nu_1} U_2 G^{\nu_2} \ldots U_l G^{\nu_l}$ des Erzeugnisses $\{\mathfrak{U}, G\}$, daß $\mathfrak{V}$ alle Elemente von $\{\mathfrak{U}, G\}$ enthält. Zugleich mit der Liste aller Elemente von $\mathfrak{G}$ wird die Liste ihrer Ordnungen angelegt; dabei erhält man die Ordnung einer Permutation P am einfachsten als kleinstes gemeinsames Vielfaches der Zyklenlängen in einer Zerlegung von P in elementfremde Zyklen.

II. Für die weitere Untersuchung der Gruppe sind die zyklischen Untergruppen von Primzahlpotenzordnung (im folgenden mit ZUPPO abgekürzt) ein wichtiges Hilfsmittel. Dies soll zunächst erläutert werden:

Im Verlauf der Rechnung werden häufig Untergruppen gebildet, die später nochmals gebraucht werden. Diese müssen in einer Form notiert werden, die

erstens gestattet, ohne längere Rechnung zwei Untergruppen zu vergleichen, zweitens mit wenig Angaben auskommt. Die Angabe aller Elemente der Untergruppen würde sicher die zweite Bedingung schlecht erfüllen. Wird dagegen nur ein aus möglichst wenigen Elementen bestehendes Erzeugendensystem notiert, so ist im allgemeinen die erste Bedingung nicht erfüllt, da man nicht ohne längere Rechnung feststellen kann, ob zwei Systeme von Elementen die gleiche Untergruppe erzeugen. Man erhält jedoch zu jeder Untergruppe $\mathfrak{U}$ folgendermaßen ein Erzeugendensystem, das $\mathfrak{U}$ eineindeutig zugeordnet ist:

In jeder ZUPPO wird ein erzeugendes Element ausgezeichnet. Eine Untergruppe wird durch die Menge $Z(\mathfrak{U})$ der in ihr enthaltenen ZUPPO erzeugt, diese Menge ist der Untergruppe eineindeutig zugeordnet. Die Menge $E(\mathfrak{U})$ der ausgezeichneten Erzeugenden der in $\mathfrak{U}$ enthaltenen ZUPPO bildet daher ein $\mathfrak{U}$ eineindeutig zugeordnetes System von Erzeugenden, das den genannten Ansprüchen weitgehend entspricht; denn es gilt für zwei Untergruppen $\mathfrak{U}$ und $\mathfrak{V}$ einer Gruppe $\mathfrak{G}$

1. $\mathfrak{U} \subset \mathfrak{V}$ dann und nur dann, wenn $E(\mathfrak{U}) \subset E(\mathfrak{V})$.
2. $\mathfrak{U} = \mathfrak{V}$ dann und nur dann, wenn $E(\mathfrak{U}) = E(\mathfrak{V})$.
3. $G^{-1}\mathfrak{U}G = \mathfrak{V}$ dann und nur dann, wenn $G^{-1}Z(\mathfrak{U})G = Z(\mathfrak{V})$.

Gerade diese letzte Eigenschaft erweist sich für die Rechnungen als sehr nützlich.

Im Teil II des Verfahrens werden daher die zyklischen Untergruppen aufgesucht und notiert. Einzelheiten, insbesondere über die Art der für jede Untergruppe zu speichernden Größen, werden in Abschnitt 3 besprochen, da sie weitgehend auf eine duale Rechenmaschine zugeschnitten sind.

Als nächstes soll der Verband der Untergruppen von $\mathfrak{G}$ untersucht werden. Die Schwierigkeit ist hierbei, ein praktisch brauchbares (d.h. mit möglichst geringem Zeitaufwand durchführbares) Verfahren zu finden, alle Untergruppen aufzusuchen.

So scheidet z.B. die Möglichkeit aus, alle Teilmengen durchzuprüfen. Man wird vielmehr versuchen, von schon bekannten Untergruppen ausgehend neue zu finden. Hierbei kann man mit den zyklischen Untergruppen beginnen und jeweils das Erzeugnis $\{\mathfrak{U}, G\}$ einer schon vorhandenen Untergruppe $\mathfrak{U}$ mit einem Element $G \notin \mathfrak{U}$ bilden. Doch ist auch dies ein übermäßiger Rechenaufwand, da man dieselbe Untergruppe auf vielerlei Weise erzeugt und erst hinterher feststellen kann, daß sie schon vorhanden ist. Man wird also anstreben, bereits von einer gegebenen Kombination $\mathfrak{U}, G$ festzustellen, ob ihr Erzeugnis gleich einer schon vorhandenen Untergruppe $\mathfrak{V}$ ist. Die hierfür notwendige Bedingung $\mathfrak{U} \subset \mathfrak{V}$ und $G \in \mathfrak{V}$ reicht nicht hin, da $\{\mathfrak{U}, G\}$ gleich einer noch nicht bekannten Untergruppe von $\mathfrak{V}$ sein kann. Um mit der Nachprüfung $\mathfrak{U} \subset \mathfrak{V}$ und $G \in \mathfrak{V}$ auszukommen, muß man also sicherstellen, daß bei jedem Schritt der Rechnung mit einer Untergruppe auch alle in ihr enthaltenen bekannt sind. Es ist zweckmäßig, zur Beschreibung dieser Situation eine neue Bezeichnung einzuführen.

Definition. Die Menge der Untergruppen von $\mathfrak{G}$, deren Ordnung Produkt von k Primfaktoren ist, werde die k-te Schicht Σ_k von $\mathfrak{G}$ genannt.

Offenbar erfüllt man die Forderung, daß mit einer Untergruppe auch alle in ihr enthaltenen bekannt sein sollen, wenn man die Untergruppen „schichtweise" konstruiert. Die Schicht Σ_1 ist bekannt, sie besteht aus den Untergruppen

von Primzahlordnung, die unter den ZUPPO vorkommen. Man hat also, von Σ_k ausgehend Σ_{k+1} zu konstruieren. Sei eine Untergruppe $\mathfrak{U} \in \Sigma_k$ und ein Element $G \notin \mathfrak{U}$ gegeben. $\{\mathfrak{U}, G\}$ ist genau dann noch nicht bekannt, wenn $\mathfrak{U} \not\subset \mathfrak{V}$ oder $G \notin \mathfrak{V}$ für jede schon vorhandene Untergruppe $\mathfrak{V} \in \Sigma_{k+1}$. Es braucht jedoch nicht $\{\mathfrak{U}, G\} \in \Sigma_{k+1}$ zu sein. Ob dies der Fall ist, kann im allgemeinen erst festgestellt werden, wenn $\{\mathfrak{U}, G\}$ gebildet wird.

Ist jedoch G im Normalisator $\mathsf{N}_{\mathfrak{U}}$ von $\mathfrak{U}$ enthalten, so ist genau dann $\{\mathfrak{U}, G\} \in \Sigma_{k+1}$, wenn es eine Primzahl p gibt, für die $G^p \in \mathfrak{U}$. Bei dem vorliegenden Programmsystem werden nun nur solche Elemente $G \in \mathsf{N}_{\mathfrak{U}}$ zur Erweiterung von $\mathfrak{U}$ benutzt. Da dann $\mathfrak{U}$ Normalteiler vom Index p in $\{\mathfrak{U}, G\}$ wird, beschränkt man sich damit auf das Aufsuchen auflösbarer Untergruppen von $\mathfrak{G}$. Auf die nicht auflösbaren Untergruppen konnte hier um so eher verzichtet werden, als die ersten einfachen Gruppen zusammengesetzter Ordnung die Ordnungen 60, 168, 360 haben, Kapazität und Rechengeschwindigkeit der benutzten Maschine aber die Ordnung der behandelbaren Gruppen auf etwa 300 beschränken.

Das benutzte Verfahren arbeitet im einzelnen folgendermaßen:

III. Man beginnt mit der Neuordnung der minimalen Untergruppen in Klassen Konjugierter. Dazu wird die erste der Untergruppen von Primzahlordnung in der Reihenfolge der ZUPPO herausgegriffen, sie und ihre Konjugierten in eine neue Liste eingetragen und die eingeordneten Gruppen in der alten Liste gestrichen. Unter den verbleibenden wird dann wieder die erste Untergruppe von Primzahlordnung gesucht usw., bis alle Untergruppen von Primzahlordnung behandelt sind. Dabei erhält man die Konjugierten einer Untergruppe $\mathfrak{U}$, indem man den Normalisator $\mathsf{N}_{\mathfrak{U}}$ bildet, $\mathfrak{G}$ in Restklassen nach $\mathsf{N}_{\mathfrak{U}}$ zerlegt und aus jeder Restklasse einen Repräsentanten wählt, mit dem man $\mathfrak{U}$ transformiert.

IV. Bei der Bildung von Σ_{k+1} aus Σ_k kann man die Rechnung weiter abkürzen, indem man die Einteilung in Klassen konjugierter Untergruppen zu Hilfe nimmt. Es gilt nämlich:

1. Jede auflösbare Gruppe $\mathfrak{G}_{k+1} \in \Sigma_{k+1}$ enthält mindestens eine Gruppe $\mathfrak{G}_k \in \Sigma_k$ mit $\mathfrak{G}_k \lhd \mathfrak{G}_{k+1}$.

2. Ist $\mathfrak{G}_k \lhd \mathfrak{G}_{k+1}$, $\mathfrak{G}_{k+1} : \mathfrak{G}_k = p$ und $\mathfrak{H}_k = X^{-1}\mathfrak{G}_k X$, $\mathfrak{H}_{k+1} = X^{-1}\mathfrak{G}_{k+1} X$, so ist $\mathfrak{H}_k \lhd \mathfrak{H}_{k+1}$ und $\mathfrak{H}_{k+1} : \mathfrak{H}_k = p$.

Man erhält daher alle auflösbaren Gruppen aus Σ_{k+1}, indem man für die Repräsentanten $\mathfrak{R}$ jeder Klasse konjugierter auflösbarer Untergruppen aus Σ_k alle Obergruppen aus Σ_{k+1} aufsucht, in denen $\mathfrak{R}$ normal ist und dann zu jeder von diesen alle ihre Konjugierten bestimmt. Man muß also zu einer gegebenen Untergruppe $\mathfrak{R}$ alle Untergruppen $\mathfrak{R}^*$ auffinden, für die

$$(*) \qquad 1.\ \mathfrak{R} \lhd \mathfrak{R}^*, \qquad 2.\ \mathfrak{R}^* : \mathfrak{R} = p$$

gilt. Jeder Repräsentant X einer erzeugenden Restklasse $X\mathfrak{R}$ der zyklischen Faktorgruppe $\mathfrak{R}^*/\mathfrak{R}$ hat die Eigenschaften

$$(**) \qquad 1.\ X \notin \mathfrak{R}, \qquad 2.\ X^p \in \mathfrak{R}, \qquad 3.\ X^{-1}\mathfrak{R}X = \mathfrak{R}, \qquad 4.\ \{X, \mathfrak{R}\} = \mathfrak{R}^*.$$

Ist die Ordnung von X gleich $p^\alpha r$, $\alpha \geqq 1$, $(p, r) = 1$, so hat X^{p^α} die Ordnung r, und aus $X^p \in \mathfrak{R}$ folgt $X^{p^\alpha} \in \mathfrak{R}$.

Ist $(\lambda, p) = 1$, so hat $X^{\lambda r}$ die Ordnung p^α, und da $\{X^{p^\alpha}, X^{\lambda r}\} = \{X\}$, hat mit X auch das Element $X^{\lambda r}$ die Eigenschaften (**). Man erhält daher schon alle

auflösbaren Gruppen $\mathfrak{R}^*$ mit (*), wenn man zur Erweiterung von $\mathfrak{R}$ nur die Erzeugenden der ZUPPO heranzieht. Ist eine solche Gruppe $\mathfrak{R}^*$ gefunden, so werden ihre Konjugierten in derselben Weise wie bei der untersten Schicht bestimmt.

Sind alle Repräsentanten von Klassen konjugierter auflösbarer Untergruppen aus Σ_k nach diesem Verfahren behandelt, so liegen damit auch alle auflösbaren Gruppen aus Σ_{k+1}, nach Klassen Konjugierter geordnet, vor, und der Vorgang kann mit Σ_{k+1} wiederholt werden, um Σ_{k+2} zu erhalten.

Das Aufsuchen der Untergruppen wird abgebrochen, wenn die Schicht Σ_{s-1} ($s =$ Anzahl der Primfaktoren der Ordnung von $\mathfrak{G}$) vollständig behandelt ist.

3. Zur Programmierung

Das oben beschriebene Verfahren benutzt eine Anzahl von Rechenvorgängen einfacher Art wie etwa die Multiplikation zweier Permutationen, aus denen kompliziertere, wie etwa das Aufsuchen des Normalisators einer Untergruppe, aufgebaut sind. Dementsprechend besteht das vorliegende Programmsystem aus (insgesamt 63) ineinander geschachtelten Unterprogrammen, die, zum Teil mittelbar, von einem Hauptprogramm aufgerufen werden. Man kann diese Unterprogramme in 5 Klassen einteilen:

1. Arithmetische Hilfsprogramme.
2. Grundprogramme zum Rechnen mit Permutationen.
3. Grundprogramme zum Rechnen mit Hilfsgrößen.
4. Gruppentheoretische Konstruktionen.
5. Oberprogramme I bis IV.

Die unter 1. fallenden Programme betreffen das Rechnen mit ganzen Zahlen, wie Multiplikation, Division mit Rest, Primfaktorzerlegung, Bildung des größten gemeinsamen Teilers mit dem Euklidischen Algorithmus, Bildung des kleinsten gemeinschaftlichen Vielfachen u. a., die im Laufe der Rechnung gebraucht werden.

Unter 2. sind die Unterprogramme zum Lesen und Drucken, Umspeichern, Vergleichen, Multiplizieren, Transformieren und Potenzieren von Permutationen gemeint. Hier war zunächst zu entscheiden, wie die Permutationen gespeichert werden sollten. Während man bei der Rechnung per Hand meist die Darstellung einer Permutation als Produkt ziffernfremder Zyklen (Zyklenschreibweise) bevorzugen wird, ist es in der Maschine zweckmäßiger, die Schreibweise als Abbildung $\begin{pmatrix} i \\ iP \end{pmatrix}$ zu wählen, da hier keine Klammersetzung mitgespeichert werden muß. Wählt man bei einer Gruppe vom Grad g als permutierte Objekte die natürlichen Zahlen $1, \ldots, g$, so kann man sich auf die Speicherung der Zahlenreihe iP beschränken, da i durch die Stellung in der Reihenfolge angegeben wird. Für kleine Grade hat jede dieser Zahlen iP nur wenige Binärstellen (Bits). Um den vorhandenen Speicherplatz möglichst gut auszunutzen, werden daher jeweils mehrere Zahlen hintereinander in dieselbe Zelle gespeichert. Die Anzahl der Zahlen pro Zelle und die Anzahl der Zellen pro Permutation werden vor Beginn der Rechnung aus dem anzugebenden Grad g und der Zellenlänge von 38 Bits von einem Unterprogramm ermittelt, das auch eine Reihe von Schiebebefehlen zum „Hervorholen" der einzelnen Zahlen aus einer Zelle herstellt (Fig. 1).

Die Multiplikation zweier Permutationen ist in der Schreibweise $\begin{pmatrix} i \\ iP \end{pmatrix}$ sehr einfach zu programmieren: Das Bild iPQ steht in Q an der iP-ten Stelle. Um aber bei einer Multiplikation nicht für jede Zahl i ausrechnen zu müssen, in welchen Bits welcher Zelle der Permutation Q die Zahl iPQ steht (dazu wäre eine Division mit Rest erforderlich), wird vor der Multiplikation die Permutation Q „ausgeschoben", d.h. die Zahlen iQ auf g feste Zellen verteilt und später das Resultat zur Speicherung wieder „zusammengeschoben".

Die Ausgabe von Permutationen kann durch Unterprogramme wahlweise in der Form $\begin{pmatrix} i \\ iP \end{pmatrix}$ oder in Zyklenschreibweise erfolgen.

Die unter 1. und 2. erwähnten Unterprogramme genügen bereits, um den Teil I des Verfahrens zu programmieren.

Bevor die nächsten Teile des Rechenverfahrens im einzelnen beschrieben werden können, muß noch einiges über die Speicherung von Untergruppen gesagt werden. Es war bereits unter II auf die Rolle der ZUPPO hierfür hingewiesen worden. Legt man eine Liste aller ZUPPO an, so genügt es zur Charakterisierung einer Untergruppe $\mathfrak{U}$, die Listennummern aller in $\mathfrak{U}$ enthaltenen ZUPPO zu kennen. Hier ist nun bei Benutzung einer Dualmaschine eine sehr elegante Form der Speicherung möglich. Sind z ZUPPO vorhanden, so wird jeder Untergruppe eine z-stellige Dualzahl zugeordnet, deren k-tes Bit 1 oder 0 ist, je nachdem, ob die k-te ZUPPO in $\mathfrak{U}$ enthalten ist oder nicht. Wir wollen diese, $\mathfrak{U}$ eineindeutig zugeordnete Dualzahl die *Kennzahl* $K(\mathfrak{U})$ nennen. Sie wird in $\left[\frac{z-1}{38}\right]+1$ Zellen gespeichert. Diese Kennzahlen dienen aber nicht nur zur platzsparenden Speicherung von Untergruppen; die in dualen Rechenmaschinen vorhandenen Befehle zur Intersektion und Disjunktion von Zelleninhalten (s. Anhang) erlauben es auch, mit ihnen in einfacher Weise zu rechnen. Es ist $K(\mathfrak{U}) \wedge K(\mathfrak{V}) = K(\mathfrak{U} \cap \mathfrak{V})$ und da genau dann $\mathfrak{U} < \mathfrak{V}$, wenn $\mathfrak{U} \cap \mathfrak{V} = \mathfrak{U}$, kann man mittels der Kennzahlen mit wenigen Befehlen feststellen, ob $\mathfrak{U} < \mathfrak{V}$ gilt. Der Disjunktion zweier Kennzahlen entspricht zwar nicht das Erzeugnis der zugehörigen Untergruppen, doch kann etwa die häufig vorkommende Frage, ob $\{\mathfrak{U}, \mathfrak{V}\} \subseteq \mathfrak{W}$ durch die Abfrage $(K(\mathfrak{U}) \vee K(\mathfrak{V})) \wedge K(\mathfrak{W}) \stackrel{?}{=} K(\mathfrak{U}) \vee K(\mathfrak{V})$ entschieden werden.

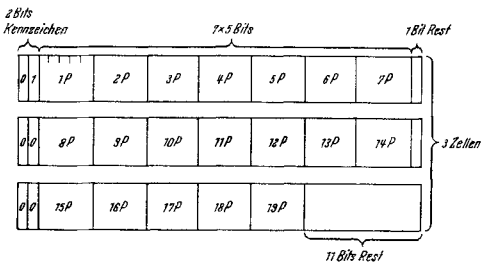

Fig. 1. Beispiel der Speicherung einer Permutation vom Grad 19

Im allgemeinen genügt es daher, von einer einmal gefundenen Untergruppe nur ihre Kennzahl zu speichern. Eine Ausnahme ist bei denjenigen Untergruppen gemacht, die als Repräsentant einer Klasse Konjugierter auftreten. Wie in Abschnitt 2 erläutert, wird unter gewissen Bedingungen aus einem Repräsentanten $\mathfrak{R} \in \Sigma_k$ und einem weiteren Element G eine neue Gruppe $\{\mathfrak{R}, G\}$ erzeugt. Dazu braucht man aber eine vollständige Liste der Elemente von $\mathfrak{R}$. Diese nach den Angaben aus $K(\mathfrak{R})$ mit Hilfe des Oberprogrammes I herzustellen, ist wegen der vielen Linksmultiplikationen und Vergleiche in diesem langwierig.

Es ist zweckmäßiger, sich zu erinnern, daß $\mathfrak{R}$ durch sukzessive Erweiterungen von Primzahlindex entstand. Kennt man die jeweils neu hinzugekommenen Elemente X_i, so kann man $\mathfrak{R}$ leicht durch wiederholte Bildung von $\mathfrak{R}_{i-1}+\mathfrak{R}_{i-1}X_i+\cdots+\mathfrak{R}_{i-1}X_i^{t_i-1}$ bei bekanntem p_i ohne Abfragen herstellen. Als X_i konnten nach Abschnitt 2 Erzeugende von ZUPPOs gewählt werden; deren Nummern (ebenfalls mehrere in einer Zelle) werden daher für jeden Repräsentanten gespeichert. (*Merkzahlen* $M(\mathfrak{R})$ für den Repräsentanten.) Eine Speicherung nach Art der Kennzahlen ist hier nicht möglich, da es auf die Reihenfolge der X_i ankommt.

Die zum Rechnen mit den Kennzahlen und zum Auswerten der Merkzahlen nötigen Unterprogramme sind unter 3. zusammengefaßt.

Das Oberprogramm II stellt die für die Rechnung mit diesen Hilfsgrößen nötigen Listen über die zyklischen Untergruppen von $\mathfrak{G}$ zusammen. Von I her liegt dazu eine Liste Λ_1 aller Elemente von $\mathfrak{G}$ und eine Parallelliste Λ_2 ihrer Ordnungen vor. Nun wird zunächst die Ordnung G von $\mathfrak{G}$ in Primfaktoren zerlegt.

$$G=p_1^{\alpha_1}p_2^{\alpha_2}\dots p_s^{\alpha_s}, \qquad p_1<p_2<\cdots<p_s.$$

Dann werden die ZUPPO in der Reihenfolge

$$p_1, p_1^2, \dots, p_1^{\beta_1}, p_2, \dots, p_2^{\beta_2}, \dots, p_s, \dots, p_s^{\beta_s}$$

($p_i^{\beta_i}$ Exponent einer p_i-Sylow-Gruppe von $\mathfrak{G}$) aufgesucht und für jede ZUPPO $\{P\}$ eingetragen:

1. ein ausgezeichnetes erzeugendes Element P in eine Liste Λ_3 der Erzeugenden von ZUPPO,

2. die Kennzahl $K(\{P\})$ in eine Parallelliste Λ_4 zu Λ_3, sowie, um schon behandelte Elemente zu kennzeichnen,

3. in der Liste Λ_2 der Ordnungen an der Stelle der ausgezeichneten Erzeugenden die Adresse derselben in Liste Λ_3,

4. an die Stelle der Ordnungen der zur Ordnung von P teilerfremden Potenzen von P eine Null.

Die verschiedenen ZUPPO werden gefunden, indem in der Liste Λ_2 unter den noch nicht abgeänderten Ordnungen der Reihe nach nach Elementen der Ordnungen $p_1, p_1^2, \dots, p_1^{\alpha_1}, p_2, \dots, p_2^{\alpha_2}, p_s, \dots, p_s^{\alpha_s}$ gesucht wird und für jedes gefundene die Operationen 1. bis 4. ausgeführt werden, bevor weitergesucht wird. Da der Exponent der p_i-Sylow-Gruppe kleiner sein kann als ihre Ordnung, wird die Suche nach Potenzen von p_i auch abgebrochen, wenn keine Elemente einer Ordnung $p_i^{\gamma_i}$ gefunden werden, und sogleich zur nächsten Primzahl übergegangen.

Nach den ZUPPO werden die restlichen zyklischen Gruppen in gleicher Weise gespeichert. Hierbei erhält man die Kennzahl einer zyklischen Untergruppe $\mathfrak{Z}$, die nicht von Primzahlpotenzordnung ist, als Summe der Kennzahlen der in $\mathfrak{Z}$ enthaltenen ZUPPO. Diese findet man, indem man alle Potenzen eines erzeugenden Elementes von $\mathfrak{Z}$ in der Liste Λ_1 aufsucht und nach Angabe in der Parallelliste Λ_2 feststellt, ob es sich um ein erzeugendes Element einer ZUPPO handelt.

Die Programme III und IV benutzen gemeinsam eine Reihe von Unterprogrammen, die gruppentheoretische Konstruktionen durchführen, bei denen sowohl mit Permutationen als mit Hilfsgrößen gerechnet wird. Solche werden gebraucht zum

Erzeugen einer Untergruppe nach gegebener Merkzahl,
Bilden von $\{\mathfrak{U}, X\}$, $K(\{\mathfrak{U}, X\})$, $M(\{\mathfrak{U}, X\})$ zu gegebenem $\mathfrak{U}$ und $X \in \mathsf{N}_{\mathfrak{U}}$ mit $X^p \in \mathfrak{U}$,
Bestimmen von $\mathsf{N}_{\mathfrak{U}}$ und $K(\mathsf{N}_{\mathfrak{U}})$ zu gegebenen $\mathfrak{U}$,
Zerlegen von $\mathfrak{G} = \mathsf{N}_{\mathfrak{U}} + \mathsf{N}_{\mathfrak{U}} X_1 + \cdots + \mathsf{N}_{\mathfrak{U}} X_r$,
Bilden von $K(X^{-1}\mathfrak{U}X)$ zu gegebenen $K(\mathfrak{U})$ und $X \in \mathfrak{G}$,
Aufsuchen der maximalen Untergruppen von $\mathfrak{U}$.

Bei den genannten Programmen treten keine neuen Typen von Größen auf, daher will ich auf ihre im einzelnen recht komplizierte Organisation nicht näher eingehen.

Um auch in den Programmen III und IV selbst möglichst weitgehenden Gebrauch vom Rechnen mit Kennzahlen machen zu können, werden außer einer Hauptliste, die die nach Klassen Konjugierter geordneten Kennzahlen aller Untergruppen enthält, drei Nebenlisten geführt, die für jede als Repräsentant ihrer Klasse ausgewählte Untergruppe zusätzlich ihre Adresse in der Hauptliste, ihre Merkzahl und die Kennzahl ihres Normalisators aufbewahren. Dies beschleunigt vor allem das Aufsuchen der Elemente X, die mit einem gegebenen Repräsentanten $\mathfrak{R} \in \Sigma_k$ eine Gruppe $\{\mathfrak{R}, X\} \in \Sigma_{k+1}$ erzeugen sollen. Wie oben erklärt, konnte man sich für deren Auswahl auf die Erzeugenden der ZUPPO beschränken. Von den an diese gestellten Bedingungen

1. $X \in \mathsf{N}_{\mathfrak{R}}$,
2. $X \notin \mathfrak{R}$,
3. $\{\mathfrak{R}, X\}$ noch nicht in Σ_{k+1} vorhanden,
4. $X^p \in \mathfrak{R}$

lassen sich die ersten 3 durch Vergleich von Kennzahlen entscheiden, da $K(\{X\})$, $K(\mathfrak{R})$, $K(\mathsf{N}_{\mathfrak{R}})$ und $K(\mathfrak{U})$ für alle $\mathfrak{U} \in \Sigma_{k+1}$ vorliegen.

4. Ausgabe der Ergebnisse

Die von der Maschine berechneten Daten werden während der Rechnung ausgegeben. Im Programmteil I wird die Liste Λ_1 aller Elemente von $\mathfrak{G}$ wahlweise in Zyklenschreibweise oder in der Form $\begin{pmatrix} i \\ i\,P \end{pmatrix}$ gedruckt, in II die Liste Λ_3 der zyklischen Untergruppen, wobei für jede zyklische Untergruppe die Λ_1-Nummern ihrer Elemente angegeben werden. Zur Beschreibung der Umordnung der minimalen Untergruppen in Klassen konjugierter wird in III für jede Untergruppe ihre neue Nummer in der Hauptliste (gekennzeichnet durch ein ′) und ihre alte Nummer in Liste Λ_3 ausgedruckt; für die vom Repräsentanten $\mathfrak{R}$ einer Klasse verschiedenen Untergruppen $\mathfrak{U}$ ferner das Element P_i, für das $P_i^{-1}\mathfrak{R}P_i = \mathfrak{U}$ ist. Bei den Untergruppen der weiteren Schichten werden in IV ihre Hauptlistennummern und die ihrer maximalen Untergruppen angegeben; für einen Repräsentanten $\mathfrak{R}_r \in \Sigma_{k+1}$ in der Form $r' = (s', z)$ außer seiner Hauptlistennummer r die seines Normalteilers $\mathfrak{R}_s$, aus der er zusammen mit der ZUPPO $\mathfrak{Z}_z$ erzeugt wurde, für die restlichen Untergruppen einer Klasse wieder die transformierenden Elemente. Nach Abschluß der Rechnung werden die Hauptlistennummern der Normalisatoren von Repräsentanten aufgesucht und gedruckt. Als einfaches Beispiel folgt das Rechenprotokoll und der danach gezeichnete Verband der Diedergruppe der Ordnung 12.

Erzeugende

(1 2 3) (4 5)
(1 2) (4 5)

Alle Elemente

$P\,2$	(1 2 3) (4 5)
$P\,3$	(1 3 2)
$P\,4$	(4 5)
$P\,5$	(1 2 3)
$P\,6$	(1 3 2) (4 5)
$P\,7$	(1 2) (4 5)
$P\,8$	(2 3)
$P\,9$	(1 3) (4 5)
$P\,10$	(1 2)
$P\,11$	(2 3) (4 5)
$P\,12$	(1 3)

Zyklische Untergruppen

0	:	$P\,4$
1	:	$P\,7$
2	:	$P\,8$
3	:	$P\,9$
4	:	$P\,10$
5	:	$P\,11$
6	:	$P\,12$
7	:	$P\,3$ $P\,5$
8	:	$P\,2$ $P\,3$ $P\,4$ $P\,5$ $P\,6$

Verband

$0' = 0$

—

$1' = 1$
$2' = 3$ TR $P\,8$
$3' = 5$ TR $P\,9$

—

$4' = 2$
$5' = 6$ TR $P\,7$
$6' = 4$ TR $P\,9$

—

$7' = 7$

—

$8' = (0'$ $1)$ $6'$ $1'$ $0'$
$9'$ TR $P\,8$ $5'$ $2'$ $0'$
$10'$ TR $P\,9$ $4'$ $3'$ $0'$

—

$11' = (0'$ $7)$ $7'$ $0'$

—

$12' = (7'$ $1)$ $7'$ $3'$ $2'$ $1'$

—

$13' = (7'$ $2)$ $7'$ $6'$ $5'$ $4'$

—

$14'$ $13'$ $12'$ $11'$ $10'$ $9'$ $8'$

Normalisatoren

$N(\ 0') = 14'$
$N(\ 1') = 8'$
$N(\ 4') = 10'$
$N(\ 7') = 14'$
$N(\ 8') = 8'$
$N(\ 11') = 14'$
$N(\ 12') = 14'$
$N(\ 13') = 14'$

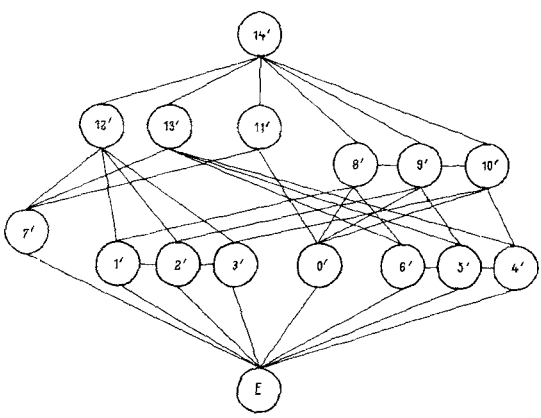

Fig. 2. Rechenprotokoll und Verband für die Diedergruppe der Ordnung 12

5. Vereinfachungen für reguläre Permutationsgruppen

Ein erheblicher Teil der Rechenzeit des obigen Programmes wird für das Multiplizieren von Permutationen und das Aufsuchen von Permutationen in Listen benötigt, wie es z. B. auftritt, wenn Kennzahlen zusammengestellt werden. Der Zeitbedarf beider Vorgänge wächst etwa quadratisch mit dem Grad der Permutation.

Für reguläre Permutationsgruppen, d. h. transitive Gruppen, deren Grad gleich ihrer Ordnung ist, kann man jedoch die Rechenzeit wesentlich verkürzen. In einer regulären Permutationsgruppe gibt es nämlich zu zwei vorgegebenen permutierten Objekten i und k genau eine Permutation P mit $iP=k$; d. h. jede Permutation P ist schon durch das Bild $1P$ eindeutig bestimmt. Ordnet man die Liste Λ_1 nach der Größe der Zahlen $1P$, so braucht man später nur die erste Zahl der neu errechneten Permutation zu untersuchen und kann in Parallellisten zu Λ_1 alle über dieses Element wünschenswerten Informationen speichern. In allen weiteren Listen genügt es dann, die Zahl $1P$ zu speichern. Insbesondere wird die Rechenzeit für die Multiplikation zweier Permutationen vom Grad der Permutationen unabhängig, da in jedem Fall nur $1PQ$ nach der früher geschilderten Methode gebildet wird.

Auch in einer ganzen Reihe von Unterprogrammen ergeben sich durch Benutzung von Parallellisten zu Λ_1 erhebliche Zeiteinsparungen, insbesondere bei der Bildung des Normalisators. Die Anwendung dieses Programmsystems ist dafür durch den hohen Speicherbedarf der Liste Λ_1 bei der Z 22 auf Gruppen bis etwa zum Grad 100 beschränkt.

6. Anwendungen

Die beiden beschriebenen Programmsysteme umfassen je etwa 3000 Befehle, so daß von den etwa 7000 freien Trommelzellen der Z 22 etwa 4000 zur Speicherung von Daten für die spezielle Gruppe zur Verfügung stehen. Speicherbedarf

Tabelle

Gruppe	Ordnung	Grad	Anzahl der ZUPPO	Anzahl der Untergruppen	Anzahl der Klassen konjugierter Untergruppen	Rechenzeit
$\mathfrak{S}^4$	24	4	16	30	11	18 min
$\mathfrak{A}^5$	60	5	31	59	9	40 min
$LF(2,7)$	168	7	78	179	15	3,5 Std
$\mathfrak{H}_{192}$	192	8	61	351	58	12,5 Std
$\mathfrak{G}_{192}$	192	8	89	469	78	22,5 Std

und Rechenzeit für die Diskussion einer Gruppe in der angegebenen Weise hängen ziemlich kompliziert von der Struktur der Gruppe (Ordnung, Grad, Anzahl der ZUPPO, Anzahl der Klassen konjugierter Untergruppen u. a.) ab. In einer Tabelle sind einige Beispiele zusammengestellt ($\mathfrak{S}^4$ symmetrische Gruppe vom Grad 4, $\mathfrak{A}^5$ alternierende Gruppe vom Grad 5). Das letzte Beispiel der Liste zeigt, daß wesentlich „größere" Gruppen schon aus Gründen der Rechenzeit kaum

noch behandelt werden können. Aber auch der Speicherbedarf beschränkt die Ordnung der vollständig behandelbaren Gruppen auf etwa 200—300. Mit Hilfe einiger verbindender Handrechnung kann man natürlich auch zur Untersuchung größerer Gruppen diese Programme zu Hilfe nehmen, die beiden Gruppen $\mathfrak{H}_{192}$ und $\mathfrak{G}_{192}$ sind z.B. maximale Untergruppen einer kristallographischen Gruppe der Ordnung 384, deren sämtliche Untergruppen gesucht werden sollten. Wird weniger Information über die von einer gegebenen Menge von Permutationen erzeugte Gruppe (z.B. nur deren Ordnung) verlangt, so können natürlich auch Gruppen höherer Ordnung untersucht werden.

Bisher sind mit den genannten Programmen eine Reihe von Gruppen durchgerechnet worden, die von speziellem Interesse waren. Ferner ist bereits begonnen, insbesondere die Variante für reguläre Darstellungen zur systematischen Durchrechnung der Gruppen kleiner Ordnung zu benutzen. Für diese existieren in der Literatur bisher nur eine Reihe von Aufzählungen, die jedoch höchstens Erzeugendensysteme angeben und sich überdies zum Teil widersprechen [*1*], S. 12.

Außer den hier beschriebenen sind am Rechenzentrum der Universität Kiel von H. Felsch zwei gruppentheoretische Programme entwickelt worden, von denen eines, wie erwähnt, aus gegebenen abstrakten Erzeugenden und definierenden Relationen eine Permutationsdarstellung der Erzeugenden ermittelt und damit die Untersuchung von durch Erzeugende und definierende Relationen gegebenen Gruppen ermöglicht.

Es wäre wünschenswert, die vorhandenen Programmsysteme weiter auszubauen, insbesondere, um Erweiterungsfragen untersuchen zu können, d.h. etwa zu gegebenen Gruppen $\mathfrak{N}$ und $\mathfrak{F}$ alle nichtisomorphen Gruppen $\mathfrak{G}$ zu finden, die einen Normalteiler $\mathfrak{N}' \cong \mathfrak{N}$ mit einer Faktorgruppe $\mathfrak{G}/\mathfrak{N}' \cong \mathfrak{F}$ besitzen. Ein erster hierzu nötiger Schritt, der auch für sich Interesse hätte, wäre die Bestimmung der Automorphismengruppe einer gegebenen Gruppe.

7. Anhang: Eigenschaften der Z 22

Die Z 22 der Firma Zuse KG ist eine programmgesteuerte Dualmaschine mit einem Magnettrommelspeicher von 8192 Zellen, deren mittlere Zugriffszeit 5 msec beträgt, sowie 25 Magnetkernspeichern. Jede Zelle enthält 38 Bits. Im Rechenwerk sind Addition (+), Intersektion (∧) und Disjunktion (∨) fest verdrahtet, die durch folgende Tabellen definiert sind: (* Übertrag)

+	0	1
0	0	1
1	1	0*

∧	0	1
0	0	0
1	0	1

∨	0	1
0	0	1
1	1	1

Negative Zahlen werden durch das in der letzten Dualstelle um 1 erhöhte Komplement dargestellt.

Die Befehle enthalten in den Bits 1—13 die Trommeladresse, in den Bits 14—18 die Schnellspeicheradresse. Die Bits 19—36 sind funktionelle Bits, die bis auf einige Ausnahmeschaltungen unabhängig voneinander wirken, die Bits 37 und 38 dienen zur Kennzeichnung. Jeder Befehl besteht aus einem der vier Grund-

befehle A (Addition), I (Intersektion), U (Umspeichern), E (Sprung) mit Zusätzen, die Stellen und Abfragen von Bedingungen, Indexsetzen, Verschiebungen und Indexzählungen im gleichen Befehl ermöglichen. Die zur Ausführung eines Befehls benötigte Zeit (Wortzeit) beträgt 0,3 msec.

Literatur

[1] COXETER, H. S. M., and W. O. J. MOSER: Generators and Relations for Discrete Groups. Berlin 1957.

[2] FELSCH, H.: Die Behandlung zweier gruppentheoretischer Verfahren auf elektronischen Rechenmaschinen. Diplomarbeit, Kiel 1960.

[3] ROTTLAENDER, A.: Nachweis der Existenz nicht isomorpher Gruppen von gleicher Situation der Untergruppen. Math. Z. **28**, 641–653 (1928).

Mathematisches Seminar der Universität Kiel
Olshausenstraße, Eingang C 4

(Eingegangen am 6. Mai 1960)