

Kohomologie und Normalisatoren in GAP

Diplomarbeit an der RWTH Aachen

vorgelegt von

Frank Celler

Februar 1992

Inhaltsverzeichnis

1	Endliche, polyzyklische Gruppen	5
1.1	Grundlegende Definitionen	5
1.2	Homomorphismen	8
1.3	Kleine Erzeugendensysteme	10
1.4	Orbit-Stabilisator-Algorithmus	11
2	Die 1-Kohomologiegruppe	15
2.1	Die n -Kohomologiegruppe $H^n(H, N)$	15
2.2	Die 1-Kohomologiegruppe $H^1(H, N)$	16
2.3	Berechnung der 1-Kohomologiegruppe	20
2.4	Reduktion der Unbekannten	25
2.5	Die H^1 einer endlichen, polyzyklischen Gruppe	27
2.6	Ein Beispiel: $H^1(S_3, Z_2^3)$	31
3	Der Normalisator	35
3.1	Der Normalisator als Stabilisator	35
3.2	Reduktion der Orbitlänge	36
3.3	Der lineare Fall	39
3.4	Reduktion der Orbitlänge im allgemeinen Fall	39
3.5	Der teilerfremde Fall	44
3.6	Endgültige Fassung	47
3.7	Ein Beispiel	48
4	Konjugiertenklassen von Komplementen	53
4.1	Das Homomorphieprinzip	53
4.2	Herunterziehen von Komplementen	55
4.3	S -Bahnen	56
4.4	Die affine Operation	58
4.5	Der zentrale Fall	61
4.6	Verfeinerung der Rekursion	64

4.7	Normale Komplemente	66
5	Kernfunktionen für den PQ	69
5.1	Endliche, polyzyklische Gruppen in GAP	69
5.2	Kommutator- und Konjugationsalgorithmus	71
5.3	Ag-Gruppen und Präsentationen	72
5.4	Pc-Präsentationen	73
5.5	GAP Funktionen	74
5.6	Ausblick	75
6	Funktionen der GAP-Bibliothek	77
6.1	Die Einskohomologiegruppe	77
6.2	Konjugiertenklassen von Komplementen	83
7	Zeiten	87
7.1	Kanonische Erzeugendesysteme	88
7.2	Normale Hülle	88
7.3	Konjugierte Untergruppen	88
7.4	Schnitte von Untergruppen	89
7.5	Kommutatoruntergruppen	90
7.6	Halluntergruppen	90
7.7	Normalisator	91

Vorwort

Unter den endlich präsentierten Gruppen besitzen die endlichen, polyzyklischen Gruppen eine besondere Bedeutung, da in diesen Gruppen die Elemente durch einen sogenannten Kollektor auf kanonische Form gebracht werden können, sofern die Gruppe durch eine AG-Präsentation gegeben ist. Ein solcher Kollektor wurde von Herrn Thomas Bishops in GAP implementiert und ermöglicht es in AG-präsentierten Gruppen in GAP zu rechnen. Die meisten der in S0G0S implementierten Algorithmen zum Rechnen in AG-präsentierten Gruppen mußten als Grundlage für die in Kapitel 1, 2 und 3 beschriebenen Algorithmen in GAP übertragen werden. Sie sind alle in der GAP Sprache geschrieben und dem Benutzer frei zugänglich. Eine vollständige Übersicht findet sich in [Gap91].

Die vorliegende Arbeit ist wie folgt gegliedert. Kapitel 1 gibt zunächst eine kurze Einführung. Hier werden die verwendeten Schreibweisen festgelegt und die grundlegenden Algorithmen für AG-Gruppen beschrieben. Kapitel 2 befaßt sich mit der Berechnung der 1-Kohomologiegruppe. Kapitel 3 beschreibt einen Algorithmus zur Berechnung des Normalisators einer Untergruppe einer AG-Gruppe mit Hilfe der in Kapitel 1 eingeführten 1-Kohomologiegruppe. Kapitel 4 beschreibt einen Algorithmus zur Berechnung von Konjugiertenklassen von Komplementen in einer AG-Gruppe. In Kapitel 5 findet sich eine Übersicht der GAP Kernfunktionen, welche für eine Implementation eines PQ in GAP notwendig wurden. Kapitel 6 gibt einen Überblick der Library-internen GAP Funktionen und Verbunde im Zusammenhang mit den Algorithmen zur Berechnung von Komplementen und der 1-Kohomologiegruppe. Kapitel 7 schließlich enthält Zeitvergleiche zwischen GAP, S0G0S und CAYLEY.

Der in Kapitel 3 beschriebene Algorithmus stellt eine Verallgemeinerung des Konjugiertenklassenalgorithmus von Elementen in AG-Gruppen dar. Die für diese Verallgemeinerung benötigte Theorie der 1-Kohomologiegruppe einer AG-Gruppe ermöglicht aber auch eine Verbesserung des in [GS92] beschriebenen Normalisatoralgorithmus.

Ich möchte mich an dieser Stelle bei allen herzlich bedanken, die durch ihre Unterstützung das Zustandekommen dieser Arbeit ermöglicht haben. Bei meinen Eltern und insbesondere bei meinem Großvater bedanke ich mich für die Unterstützung, ohne die ein Studium der Mathematik nicht möglich gewesen wäre. Ich danke Herrn Professor Neubüser für die interessante Aufgabenstellung und die Gelegenheit im Rahmen der Entwicklung des gruppentheoretischen Programmsystems GAP die vorliegende Arbeit anfertigen zu können. Ferner ermöglichte er mir, bei der Veröffentlichung einer Zusammenfassung meiner Arbeit mitzuwirken. Herrn Jürgen Mnich danke ich für die erfolgreiche Zusammenarbeit während der Prüfungsvorbereitungen. Mein besonderer Dank gilt auch Herrn Dr. Klaus Lux und Herrn Martin Schönert für aufschlußreiche und anregende Diskussionen.

Kapitel 1

Endliche, polyzyklische Gruppen

In diesem Kapitel werden die grundlegenden Definitionen und Algorithmen vorgestellt, welche in den folgenden Kapiteln zum Rechnen in endlichen, polyzyklischen Gruppen benötigt werden.

1.1 Grundlegende Definitionen

Definition 1.1 *Es sei H eine beliebige Gruppe.*

1. Für Elemente a, b von H wird $[a, b] := a^{-1}b^{-1}ab$ als der Kommutator von a und b bezeichnet.
2. Die von den Kommutatoren erzeugte Untergruppe $H' := \langle [a, b] \mid a, b \in H \rangle$ heißt Kommutatorgruppe von H .
3. Die i -te Kommutatorgruppe $H^{(i)}$ von H ist rekursiv definiert durch $H^{(0)} := H$ und $H^{(i+1)} := (H^{(i)})'$.

Es gilt für die Kommutatorgruppen folgendes

Lemma 1.2 *Es sei H eine beliebige Gruppe. Dann bilden die Kommutatorgruppen $H^{(i)}$ für $i = 0, 1, \dots$ eine absteigende Normalreihe $H =: H^{(0)} \geq H^{(1)} \geq \dots \geq H^{(j)} \geq \dots$, wobei jedes $H^{(i)}$ eine charakteristische Untergruppe von H ist.*

Definition 1.3 *Es sei H eine beliebige Gruppe. Genau dann heißt H auflösbar, wenn es ein $n \in \mathbb{N}$ gibt mit $H^{(n)} = \langle 1 \rangle$.*

Eine für das Rechnen wichtige äquivalente Definition gibt

Satz 1.4 *Es sei G eine beliebige endliche Gruppe. Dann sind folgende Aussagen äquivalent.*

1. G ist auflösbar.
2. G besitzt eine Subnormalreihe mit zyklischen Faktoren.
3. G besitzt eine Subnormalreihe mit abelschen Faktoren.
4. Die Kompositionsfaktoren von G haben Primzahlordnung.

Definition 1.5 *Es sei H eine beliebige Gruppe. Genau dann heißt H polyzyklisch, wenn H eine endliche Subnormalreihe S mit zyklischen Faktoren besitzt. Eine solche Reihe S heißt ab jetzt zyklische Subnormalreihe.*

Nach Satz 1.4 sind also für eine endliche Gruppe G die Begriffe „polyzyklisch“ und „auflösbar“ gleich. Ausgehend von einer Subnormalreihe mit zyklischen Faktoren einer endlichen, polyzyklischen Gruppe G erhält man die sogenannte Potenz-Kommutator Präsentation von G .

Definition 1.6 *Es sei G eine endliche Gruppe mit zyklischer Subnormalreihe $G = G_0 \geq G_1 \geq \dots \geq G_n = \langle 1 \rangle$.*

1. Für Elemente g_1, \dots, g_n aus G mit $G_i = \langle G_{i+1}, g_{i+1} \rangle$ für $i \in \{0, \dots, n-1\}$, heißt die Folge (g_1, \dots, g_n) eine AG-Erzeugenden-Folge für G oder ein AG-System von G .
2. Es sei (g_1, \dots, g_n) ein AG-System von G . Wenn alle Faktoren G_i/G_{i+1} für $i \in \{0, \dots, n-1\}$ Primzahlordnung haben, so heißt die Folge (g_1, \dots, g_n) ein PAG-System von G oder eine PAG-Erzeugenden-Folge für G .

Lemma 1.7 *Es sei G eine endliche, polyzyklische Gruppe mit AG-System (g_1, \dots, g_n) . Es sei o_i die Ordnung von G_{i-1}/G_i für $i \in \{1, \dots, n\}$. Dann besitzt G eine Präsentation von der Form*

$$\begin{aligned} g_i^{o_i} &= w_{ii}(g_{i+1}, \dots, g_n), \quad i \in \{1, \dots, n\}, \\ [g_i, g_j] &= w_{ij}(g_{j+1}, \dots, g_n), \quad 1 \leq j < i \leq n. \end{aligned}$$

Definition 1.8 *Es seien die Voraussetzungen wie in Lemma 1.7. Dann bezeichnet man die Präsentation aus Lemma 1.7 als AG-Präsentation beziehungsweise PAG-Präsentation, falls von einem PAG-System ausgegangen wird.*

Bezüglich eines AG-Systems (g_1, \dots, g_n) einer endlichen, polyzyklischen Gruppe G kann jedes Element $g \in G$ eindeutig dargestellt werden als *normiertes Wort*

$$g = g_1^{\nu_1} \cdots g_n^{\nu_n} \quad \text{mit} \quad 0 \leq \nu_i < o_i.$$

Zur Vereinfachung der Schreibweise noch

Definition 1.9 Die Voraussetzung seien wie in Lemma 1.7. Es sei ein Element $g \in G$ gegeben und $g = g_1^{\nu_1} \cdots g_n^{\nu_n}$ sei seine Darstellung als normiertes Wort.

1. $\nu_i(g) := \nu_i$ heißt der *i*-te Exponent von g .
2. Falls $\nu_i = 0$ für $i = 0, \dots, k-1$ und $\nu_k \neq 0$, so heißt $\lambda(g) := \nu_k$ der führende Exponent von g und $w(g) := k$ die Tiefe von g .
3. Der Index o_k von G_k in G_{k-1} heißt die relative Ordnung von g .

Da sich jedes endliche Produkt in den Erzeugern und ihren Inversen durch einen sogenannten Kollektor auf Normalform bringen läßt, können wir ab jetzt alle Worte als normiert betrachten, diese Wörter heißen „Ag-Wörter“. Insbesondere ist also das Wortproblem lösbar.

Nicht nur die Elemente einer endlichen, polyzyklischen Gruppe können bezüglich eines gegebenen PAG-Systems eindeutig dargestellt werden.

Definition 1.10 Es sei G eine endliche, polyzyklische Gruppe mit PAG-System (g_1, \dots, g_n) . Es sei eine Untergruppe U von G gegeben. Ein Erzeugendensystem (u_1, \dots, u_s) von U heißt kanonisches Erzeugendensystem beziehungsweise CGS von U bezüglich (g_1, \dots, g_n) , wenn folgende Eigenschaften erfüllt sind.

1. (u_1, \dots, u_s) ist ein PAG-System für U ,
2. $w(u_j) < w(u_i)$ für alle $1 \leq j < i \leq s$,
3. $\lambda(u_i) = 1$ für alle $1 \leq i \leq s$,
4. $\nu_{w(u_i)}(u_j) = 0$ für alle $i \neq j$.

Falls für ein Erzeugendensystem (u_1, \dots, u_s) nur die Eigenschaften 1 und 2 erfüllt sind, so heißt (u_1, \dots, u_s) ein induziertes Erzeugendensystem beziehungsweise IGS von U bezüglich (g_1, \dots, g_n) .

Lemma 1.11 Die Voraussetzungen seien wie in Definition 1.10. Es seien Untergruppen U und W mit kanonischen Erzeugendensystemen (u_1, \dots, u_s) und (w_1, \dots, w_t) gegeben. Aus $U = W$ folgt $s = t$ und $(u_1, \dots, u_s) = (w_1, \dots, w_t)$, das heißt, Untergruppen können eindeutig durch ihre kanonischen Erzeugendensysteme identifiziert werden.

Wie in [LNS84] beschrieben, läßt sich durch einen verallgemeinerten Gaußalgorithmus, welcher nicht kommutativ durchgeführt werden muß, zu jedem Erzeugendensystem ein kanonisches Erzeugendensystem berechnen. Insbesondere kann man also entscheiden, ob zwei Untergruppen gleich sind.

1.2 Homomorphismen

Folgende beiden Lemmata sind im Zusammenhang mit Homomorphismen nützlich, da sich mit ihnen Kerne und Bilder von Homomorphismen durch einen gewöhnlichen Gauß-Algorithmus bestimmen lassen.

Lemma 1.12 *Es sei G eine endliche, polyzyklische Gruppe mit PAG-System (g_1, \dots, g_n) und α ein Epimorphismus von G auf eine weitere Gruppe H . Dann enthält $(g_1\alpha, \dots, g_n\alpha)$ ein PAG-System von H , das heißt, es existiert eine Teilfolge $\{i_1, \dots, i_l\}$ von $\{1, \dots, n\}$, so daß $(g_{i_1}\alpha, \dots, g_{i_l}\alpha)$ ein PAG-System von H ist.*

Beweis: Beweis durch Induktion nach n . Falls $g_n\alpha$ trivial ist, so ist die Induktionsverankerung in Ordnung, da die leere Folge ein PAG-System von $\langle 1 \rangle = G_n\alpha$ ist. Ist $g_n\alpha$ nicht trivial, so hat $g_n\alpha$ eine Ordnung, welche die Primzahl o_n teilt. Somit hat aber $g_n\alpha$ Ordnung o_n und das PAG-System $(g_n\alpha)$ erzeugt $G_n\alpha$. Dies zeigt insgesamt die Induktionsverankerung. Es sei jetzt per Induktion eine Teilmenge $\{i_2, \dots, i_l\}$ von $\{2, \dots, n\}$ bekannt, so daß $(g_{i_2}\alpha, \dots, g_{i_l}\alpha)$ ein PAG-System von $G_2\alpha$ ist. Falls $g_1\alpha$ in $G_2\alpha$ liegt, so zeigt dies die Induktionsbehauptung, denn dann gilt $G_2\alpha = G_1\alpha$. Sollte $g_1\alpha$ nicht in $G_2\alpha$ liegen, so erzeugt $g_1\alpha$ zusammen mit $G_2\alpha$ die Gruppe $G_1\alpha$ und $g_1\alpha G_2\alpha$ hat Primzahlordnung. Weiter folgt aus $G_1 \triangleright G_2$, daß $G_1\alpha \triangleright G_2\alpha$. Damit erfüllt $(g_1\alpha, g_{i_2}\alpha, \dots, g_{i_l}\alpha)$ die Behauptung. $\ddagger\ddagger$

Lemma 1.13 *Es sei G eine endliche, polyzyklische Gruppe mit einer PAG-Erzeugenden-Folge (g_1, \dots, g_n) . Es sei weiter eine Untergruppe U von G mit einem PAG-System (u_1, \dots, u_l) gegeben. Dann entsteht aus (u_1, \dots, u_l) durch einen gewöhnlichen Gaußalgorithmus, das heißt, ohne Bildung von Kommutatoren und Potenzen, ein kanonisches Erzeugendensystem von U bezüglich (g_1, \dots, g_n) .*

Beweis: Wir führen eine Induktion nach l durch. Für $l = 1$ ist die Behauptung wahr, denn es gilt sicherlich $u_1 \neq 1$ und damit ist $\langle u_1 \rangle$ zyklisch von Primzahlordnung. Die Behauptung stimme also für $\{u_2, \dots, u_l\}$. Es sei \hat{U} die von $\{u_2, \dots, u_l\}$ erzeugte Untergruppe mit dem sortierten und normierten kanonischen Erzeugendensystem $(\hat{u}_2, \dots, \hat{u}_l)$, welches aus $\{u_2, \dots, u_l\}$ durch einen gewöhnlichen Gaußalgorithmus hervorgegangen ist. Dann hat \hat{U} insbesondere eine echt kleinere Ordnung als U . Dividiert man von u_1 das Elemente gleicher Tiefe aus $\{\hat{u}_2, \dots, \hat{u}_l\}$, falls vorhanden, ab und wiederholt diesen Vorgang solange, bis kein Element in $\{\hat{u}_2, \dots, \hat{u}_l\}$ mehr die gleiche Tiefe hat, und normiert das erhaltene Element noch bezüglich $(\hat{u}_2, \dots, \hat{u}_l)$, so erhält man ein neues Element \hat{u}_1 . Dieses Element ist insbesondere ungleich 1, denn sonst wäre $u_1 \in \hat{U}$ und damit $|\hat{U}| = |\hat{U}| < |U|$. Nun kann man \hat{u}_1 in $\{\hat{u}_2, \dots, \hat{u}_l\}$ entsprechend der Tiefe einfügen und die entstehende Folge normieren. Diese Folge erzeugt U und hat l nach der Tiefe sortierte und normierte Erzeuger. In einem nicht-kommutativen Gaußalgorithmus würden nun noch Kommutatoren und Potenzen gebildet. Da aber U Kompositionslänge l hat, können diese nichts neues mehr liefern und die entstandene Folge bildet ein kanonisches Erzeugendensystem. $\ddagger\ddagger$

Die beiden Lemmata liefern ein zur Berechnung der Normalisators benötigtes

Lemma 1.14 *Es seien G und U wie oben. Es sei weiter ein Element $g \in G$ gegeben. Dann ist $\{u_1^g, \dots, u_l^g\}$ nach einem gewöhnlichen Gaußalgorithmus ein kanonisches Erzeugendensystem von U^g .*

Das heißt also, daß ein CGS der zu U unter g konjugierten Untergruppe aus dem CGS von U berechnet werden kann, ohne daß ein nicht-kommutativer Gaußalgorithmus durchgeführt werden muss.

Andererseits läßt sich mit diesen Lemmata auch recht einfach der Kern und das Bild eines Homomorphismus mit endlichen auflösbarem Bild- und Definitionsbereich berechnen.

Lemma 1.15 *Es sei G eine endliche, polyzyklische Gruppe mit PAG-System (g_1, \dots, g_n) und α ein Homomorphismus von G auf eine weitere endliche, polyzyklische Gruppe H . Es bezeichne h_i die Bilder von g_i unter α für $i \in \{1, \dots, n\}$. Wenn man*

$$\left(\begin{array}{c|c} h_1 & g_1 \\ \vdots & \vdots \\ h_n & g_n \end{array} \right)$$

durch einen gewöhnlichen Gaußalgorithmus auf Dreiecksgestalt

$$\left(\begin{array}{c|c} u_1 & v_1 \\ \vdots & \vdots \\ u_l & v_l \\ 1 & v_{l+1} \\ \vdots & \vdots \\ 1 & v_n \end{array} \right)$$

bringt, so ist (v_{l+1}, \dots, v_n) ein kanonisches Erzeugendensystem des Kerns und (u_1, \dots, u_l) ein kanonisches Erzeugendensystem von $\text{Bild}(\alpha)$.

Beweis: Nach Lemma 1.12 und 1.13 ist (u_1, \dots, u_l) ein kanonisches Erzeugendensystem von $\text{Bild}(\alpha)$. Damit hat $G/\text{Kern}(\alpha)$ Kompositionslänge l . Da G Kompositionslänge n hat, muß der Kern von α Kompositionslänge $n - l$ haben. Andererseits liegt (v_{l+1}, \dots, v_n) im $\text{Kern}(\alpha)$ und die Erzeuger (v_{l+1}, \dots, v_n) sind normiert und nach der Tiefe sortiert. Da der Kern die Kompositionslänge $n - l$ hat, muß die Folge ein kanonisches Erzeugendensystem sein. $\ddagger\ddagger$

1.3 Kleine Erzeugendensysteme

Während kanonische Erzeugendensysteme zur Identifikation von Untergruppen besonders geeignet sind, ist für verschiedene Algorithmen eine kleine Anzahl von Erzeugenden wichtig. Dazu zuerst

Satz 1.16 *Es sei G eine endliche, polyzyklische Gruppe mit Hauptreihenlänge l . Dann gibt es ein Erzeugendensystem mit höchstens l Erzeugern.*

Beweis: Es sei $G = G_0 > G_1 > \dots > G_l = \{1\}$ eine Hauptreihe. Weiter seien für $i = 1, \dots, l$ Elemente $g_i \in G_{i-1} \setminus G_i$ gegeben. Dann folgt per Induktion, daß $\{g_1, \dots, g_i\}$ die Gruppe G erzeugt. Denn für $l = 1$ erzeugt $g_1 G_1$ eine normale Untergruppe von G/G_1 , welche nach Voraussetzung echt oberhalb von G_1 liegt. Also erzeugt $g_1 G_1$ die Gruppe G/G_1 . Nach Induktionsvoraussetzung erzeuge $\{g_1 G_i, \dots, g_i G_i\}$ die Gruppe G/G_i . Damit wird G/G_{i+1} von $\{g_1 G_{i+1}, \dots, g_i G_{i+1}\}$ und G_i/G_{i+1} erzeugt. Da G_i/G_{i+1} irreduzibel ist, gilt für jedes $g_{i+1} \in G_i \setminus G_{i+1}$

$$\begin{aligned} G/G_{i+1} \triangleright G_i/G_{i+1} &= \langle g_{i+1} G_{i+1} \rangle^{\langle g_1 G_{i+1}, \dots, g_i G_{i+1}, G_i/G_{i+1} \rangle} \\ &= \langle g_{i+1} G_{i+1} \rangle^{\langle g_1 G_{i+1}, \dots, g_i G_{i+1} \rangle}. \end{aligned}$$

Damit erzeugt aber nun $\{g_1 G_{i+1}, \dots, g_{i+1} G_{i+1}\}$ die Gruppe G/G_{i+1} und damit folgt die Behauptung. $\ddagger\ddagger$

Man beachte, daß im allgemeinen weit weniger Erzeuger benötigt werden, denn falls G_i/G_{i+1} ein \mathbf{Z}_p -Vektorraum ist und $g_i^p \neq 1$, so erzeugt g_i^p einen weiteren Hauptfaktor. Es ist auch weiterhin nicht notwendig eine Hauptreihe zu kennen, denn, falls eine Normalreihe mit elementar abelschen Faktoren bekannt ist, muß nur in jedem Schritt zusätzlich überprüft werden, ob schon der ganze Faktor aufgespannt wird oder ob noch weitere Basisvektoren hinzugenommen werden müssen. Insgesamt liefert dies

Algorithmus 1.1. (Kleines Erzeugendensystem)

Eingabe :

- Eine endliche, polyzyklische Gruppe G .
- Eine Normalreihe $E = [G = N_1, \dots, N_{r+1} = \{1\}]$ mit elementar abelschen Faktoren.

Ausgabe : Ein Erzeugendensystem für G .

```
SmallSystem := function( G, E )
  L := [];
  for i from 1 to Length(E)-1 do
    M := Urbilder der Erzeuger von E[i]/E[i+1];
    Append( L, M );
```

```

    od;
    U := TrivialSubgroup( U );
    S := [ ];
    i := 1;
    while U ≠ G do
        while L[i] ∈ U do
            i := i + 1;
        od;
        U := Closure( U, L[i] );
        Add( S, L[i] );
    od;
    return S;
end.

```

Man beachte, daß sich durch Multiplikation zweier Erzeuger, insbesondere solcher zu verschiedenen Primzahlen der Kompositionsfaktoren, die Anzahl der Erzeuger eventuell noch weiter reduzieren läßt. Es ist weiterhin möglich, die alten Erzeugenden in den neuen auszudrücken, indem man mit den neuen Erzeugenden einen nicht-kommutativen Gaußalgorithmus mit abstrakten Worten durchführt.

1.4 Orbit-Stabilisator-Algorithmus

Viele Berechnungen in einer endlichen, polyzyklischen Gruppe, wie zum Beispiel der Normalisator einer Untergruppe oder der Schnitt zweier Untergruppen, lassen sich auf eine Stabilisator-Berechnung zurückführen. Der grundlegende Algorithmus ist in [LNS84] Seite 111 beschrieben. Er läßt sich als reiner Stabilisator-Algorithmus oder als Orbit-Stabilisator-Algorithmus formulieren.

Folgender Algorithmus berechnet für eine endliche, polyzyklische Gruppe mit Operationsbereich Ω und einen Punkt ω den Orbit und den Stabilisator.

Algorithmus 1.2. (Stabilisator-Orbit)

- Eingabe* :
- Eine endliche, polyzyklische Gruppe G mit CGS (g_1, \dots, g_n) und Operationsbereich Ω .
 - Ein Punkt ω aus Ω .

Ausgabe : Ein kanonisches Erzeugendensystem für $Stab_G(\omega)$ sowie der Orbit ω^G .

```

StabilizerOrbit := function( G, ω )
    S := [ ];
    D := [ω];
    R := [1];

```

```

for i from n downto 1 do
  p :=  $\omega^{g_i}$ ;
  if p ∈ D then
    pos := Position( D, p );
    Add( S,  $g_i * R[pos]^{-1}$  );
  else
    oldD := D;
    oldR := R;
    for e from 1 to RelativeOrder(  $g_i$  )-1 do
      for p ∈ oldD do
        Add( D,  $p^{g_i^e}$  );
      od;
      for r ∈ oldR do
        Add( R,  $r * g_i^e$  );
      od;
    od;
  fi;
od;
return Reversed( S ), D;
end.

```

In diesem platzintensiven Algorithmus wird also der gesamte Orbit des Punktes ω sowie Repräsentanten der entsprechenden Restklassen des Stabilisators abgespeichert. Im Gegensatz dazu behält der zeitintensive Stabilisator-Algorithmus nur die Repräsentanten der Stabilisatorrestklassen.

Algorithmus 1.3. (Stabilisator)

Eingabe : • Eine endliche, polyzyklische Gruppe G mit PAG-System (g_1, \dots, g_n) und Operationsbereich Ω .

• Ein Punkt ω aus Ω .

Ausgabe : Ein kanonisches Erzeugendensystem für $Stab_G(\omega)$.

```

Stabilizer := function( G,  $\omega$  )
  S := [ ];
  R := [1];
  for i from n downto 1 do
    pos := 1;
    while pos < Length(R) and  $\omega^{g_i} \neq \omega^{R[pos]}$  do
      pos := pos + 1;
    od;
    if pos = Length(R) then

```

```

    oddR := R;
    for e from 1 to RelativeOrder( g_i )-1 do
        for r ∈ oddR do
            Add( R, r * g_i^e );
        od;
    od;
else
    Add( S, g_i * R[pos]^-1 );
fi;
od;
return Reversed( S );
end.

```

In diesem Algorithmus werden nur noch die Repräsentanten der Stabilisatorrestklassen abgespeichert. Auch auf diese Repräsentanten könnte noch verzichtet werden, da sie immer von der Form $g_{i_1}^{e_1} * \dots * g_{i_l}^{e_l}$ für $0 \leq e_j < o_j$ und Tiefen i_j , welche nicht in S vorkommen, sind.

Beide Algorithmen sind im allgemeinen jedoch immer noch zu platz- oder zeitintensiv, um direkt angewandt zu werden. In Gruppen G kleiner Ordnung lassen sich mit ihnen der Normalisator einer Untergruppe U als Stabilisator von U in G unter Konjugation oder der Schnitt zweier Untergruppen U und V als Stabilisator der Rechtsnebenklasse $U \cdot 1$ in V bei Multiplikation von rechts berechnen. In grösseren Gruppen sind jedoch weitere Methoden zur Reduktion der Orbitlänge notwendig. In den folgenden Kapiteln werden solche Methoden vorgestellt. Dazu benötigen wir folgende Abwandlung des Homomorphieprinzips aus [LNS84] Abschnitt 3.

Lemma 1.17 *Es sei H eine beliebige Gruppe mit Operationsbereich Ω und Normalteiler N . Es bezeichne $Stab_H(\omega)$ den Stabilisator für ein $\omega \in \Omega$ in H und $Stab_H(\omega^N) = \{h \mid h \in H, \omega^h \in \omega^N\}$ den Blockstabilisator von ω^N .*

1. *Es sei ein Punkt $\omega \in \Omega$ gegeben. Dann ist der Orbit ω^N ein Block für die Operation von H .*
2. *Für jedes $b \in Stab_H(\omega^N)$ existiert ein $n_b \in N$, so daß $\omega^b = \omega^{n_b}$.*
3. *Falls $Stab_{H/N}(\omega^N) = \langle b_1N, \dots, b_kN \rangle$ und $Stab_N(\omega) = \langle n_1, \dots, n_l \rangle$, so gilt*

$$Stab_H(\omega) = \langle b_1n_{b_1}^{-1}, \dots, b_kn_{b_k}^{-1}, n_1, \dots, n_l \rangle.$$

Literatur

Die hier vorgestellten Definitionen und Algorithmen umreißen nur die in den folgenden Kapiteln benötigten Werkzeuge. Eine weitergehende Einführung, deren Notation hier übernommen wurde, findet sich in [LNS84]. Eine erweiterte Version des Orbitalalgorithmus für den Fall, daß ein Homomorphismus in eine Matrixgruppe gegeben ist, welche die Operation liefert, befindet sich in [GS92].

Kapitel 2

Die 1-Kohomologiegruppe

In diesem Kapitel werden die Konjugiertenklassen von Komplementen eines elementar abelschen Normalteilers N im semidirekten Produkt von N mit einer endlichen, polyzyklischen Gruppe H beschrieben. Diese Beschreibung geschieht mit Hilfe der 1-Kohomologiegruppe $H^1(H, N)$.

2.1 Die n -Kohomologiegruppe $H^n(H, N)$

Definition 2.1 *Es sei H eine beliebige Gruppe und N eine abelsche Gruppe. Falls ein Homomorphismus $\alpha : H \rightarrow \text{Aut}(N)$ existiert, so heißt N ein H -Modul.*

Es sei ab jetzt H eine beliebige Gruppe mit einem additiv geschriebenen H -Modul N . Für nichtnegative ganze Zahlen $n = 0, 1, \dots$ bezeichne $C^n(H, N)$ die Menge der Abbildungen

$\overbrace{H \times \dots \times H}^n \rightarrow N$. Die Addition zweier Abbildungen aus $C^n(H, N)$ geschieht wie üblich argumentweise. Für $n = 0$ sind die Abbildungen also argumentlose, konstante Funktionen. Dann ist $C^n(H, N)$ bezüglich der argumentweisen Addition eine abelsche Gruppe.

Lemma 2.2 *Die Abbildung $\Delta_n : C^n(H, N) \rightarrow C^{n+1}(H, N)$, welche durch*

$$\begin{aligned} (\Delta_n \gamma)(g_1, \dots, g_{n+1}) &= \gamma(g_2, \dots, g_{n+1}) + \\ &\quad \sum_{i=1}^n (-1)^i \gamma(g_1, \dots, g_{i-1}, g_i g_{i+1}, \dots, g_{n+1}) + \\ &\quad (-1)^{n+1} \gamma(g_1, \dots, g_n)^{\alpha(g_{n+1})} \end{aligned}$$

für $\gamma \in C^n(H, N)$ definiert ist, beschreibt einen Homomorphismus von $C^n(H, N)$ nach $C^{n+1}(H, N)$.

Beweis: Siehe [Hup67] I.§16.12. ‡‡

Zur Vereinfachung der Bezeichnungen benötigen wir noch

Definition 2.3 *Es sei H eine beliebige Gruppe und N ein H -Modul.*

1. $Z^n(H, N) = \text{Kern}(\Delta_n)$ heißt die Gruppe der n -Kozykel und ein Element aus $Z^n(H, N)$ heißt n -Kozykel.
2. $B^n(H, N) = \text{Bild}(\Delta_{n-1})$ heißt Gruppe der n -Koränder und ein Element aus $B^n(H, N)$ heißt n -Korand.

Sowohl Z^n als auch B^n sind per Definition additiv geschriebene abelsche Gruppen. Da für alle $f \in C^{n-1}(H, N)$ gilt, daß $\Delta_n(\Delta_{n-1}(f)) = 0$, folgt $B^n(H, N) \subseteq Z^n(H, N)$ —siehe [Hup67] I.§16.11.

Definition 2.4 *Die Faktorgruppe $H^n(H, N) = Z^n(H, N)/B^n(H, N)$ heißt n -te Kohomologiegruppe von H mit Koeffizienten in N .*

Für die erste und zweite Kohomologiegruppe sind gruppentheoretische Interpretationen bekannt, siehe [Hup67]. Im folgenden werden die erste Kohomologiegruppe sowie Berechnungsmöglichkeiten weiter untersucht.

2.2 Die 1-Kohomologiegruppe $H^1(H, N)$

Es seien H, N und $\alpha : H \rightarrow \text{Aut}(N)$ wie im vorhergehenden Abschnitt.

Lemma 2.5 *Es sei H eine beliebige Gruppe, N ein H -Modul bezüglich α . Dann ist*

$$Z^1(H, N) = \{\gamma : H \rightarrow N \mid \forall h_1, h_2 \in H : \gamma(h_1 h_2) = \gamma(h_1)^{\alpha(h_2)} + \gamma(h_2)\}$$

und

$$B^1(H, N) = \{\gamma : H \rightarrow N \mid \exists n \in N \ \forall h \in H : \gamma(h) = (-n)^{\alpha(h)} + n\}.$$

Beweis: Nach Definition ist $\Delta_1 : C^1(H, N) \rightarrow C^2(H, N)$ gegeben durch

$$(\Delta_1 \gamma)(h_1, h_2) = \gamma(h_2) - \gamma(h_1 h_2) + \gamma(h_1)^{\alpha(h_2)}$$

für ein $\gamma \in C^1(H, N)$, beliebige $h_1, h_2 \in H$. Also liegt ein γ genau dann im Kern von Δ_1 , wenn $\gamma(h_2) + \gamma(h_1)^{\alpha(h_2)} = \gamma(h_1 h_2)$ für alle $h_1, h_2 \in H$ gilt. Andererseits ist $\Delta_0 : C^0(H, N) \rightarrow C^1(H, N)$ gegeben durch

$$(\Delta_0 \gamma)(h) = \gamma() - \gamma()^{\alpha(h)} = (-n)^{\alpha(h)} + n$$

für ein beliebiges festes $\gamma = (() \mapsto n)$ aus $C^0(H, N)$, beliebige $h \in H$. Dies zeigt auch den zweiten Teil der Behauptung. ‡‡

Es sei $G = \{(h, n) \mid h \in H, n \in N\}$ das semidirekte Produkt $H \ltimes N$ von H mit N mit der natürlichen Verknüpfung

$$(h_1, n_1)(h_2, n_2) = (h_1 h_2, n_1^{\alpha(h_2)} + n_2).$$

Dann gilt

$$\begin{aligned} H^* &= \{(h, 0) \mid h \in H\} \cong H, \\ G \triangleright N^* &= \{(1, n) \mid n \in N\} \cong N. \end{aligned}$$

Alle Komplemente von N^* in G lassen sich leicht durch die abelsche Gruppe $Z^1(H, N)$ beschreiben.

Lemma 2.6 *Es sei H eine beliebige Gruppe, N ein H -Modul, G das obige semidirekte Produkt von H mit N . Dann ist jedes Komplement K von N^* in G von der Form*

$$K = \{(h, \gamma(h)) \mid h \in H\}$$

für ein eindeutiges $\gamma \in Z^1(H, N)$ und umgekehrt, das heißt, die Komplemente K entsprechen eineindeutig den Elementen von $Z^1(H, N)$.

Beweis: Es sei ein beliebiges Komplement K von N^* in G gegeben. Da $H^* \leq G = KN^*$, gilt für alle $h \in H$

$$(h, 0) = k_h \cdot (1, n_h)$$

für eindeutig bestimmte $k_h \in K$ und $n_h \in N$, da $K \cap N^* = \{1\}$. Also ist $\gamma_K : H \rightarrow N$ durch $\gamma_K(h) := -n_h$ wohldefiniert und es gilt

$$\{(h, \gamma_K(h)) \mid h \in H\} \leq K. \quad (2.1)$$

Andererseits gilt für jedes $k \in K$, daß $k \in G = H^* N^*$, also $k = (h_k, 0) \cdot (1, n_k)$ beziehungsweise

$$(h_k, 0) = k \cdot (1, -n_k).$$

Da $k = (h_k, \gamma_K(h_k)) \in K$ beliebig war, gilt in 2.1 die Gleichheit. Da K eine Untergruppe von G ist, gilt für beliebige $h_1, h_2 \in H$

$$(h_1, \gamma_K(h_1))(h_2, \gamma_K(h_2)) = (h_1 h_2, \gamma_K(h_1)^{\alpha(h_2)} + \gamma_K(h_2)) \in K$$

und damit, da die Elemente in K , wie oben gesehen, durch den H -Anteil eindeutig festgelegt sind, daß

$$\begin{aligned} (h_1 h_2, \gamma_K(h_1 h_2)) &= (h_1, \gamma_K(h_1))(h_2, \gamma_K(h_2)) \\ &= (h_1 h_2, \gamma_K(h_1)^{\alpha(h_2)} + \gamma_K(h_2)). \end{aligned}$$

Also $\gamma_K \in Z^1(H, N)$ nach Lemma 2.5.

Andererseits liefert jedes Element $\gamma \in Z^1(H, N)$ ein Komplement

$$K_\gamma = \{(h, \gamma(h)) \mid h \in H\}.$$

Denn $K_\gamma N^* = G$ und $K_\gamma \cap N^* = \{1\}$ gilt per Konstruktion für die Menge K_γ . Wegen $0 = \gamma(1) = \gamma(hh^{-1}) = \gamma(h)^{\alpha(h^{-1})} + \gamma(h^{-1})$, gilt

$$\begin{aligned} (h_1, \gamma(h_1))(h_2, \gamma(h_2)) &= (h_1 h_2, \gamma(h_1)^{\alpha(h_2)} + \gamma(h_2)) \\ &= (h_1 h_2, \gamma(h_1 h_2)), \\ (h, \gamma(h))^{-1} &= (h^{-1}, (-\gamma(h))^{\alpha(h^{-1})}) \\ &= (h^{-1}, \gamma(h^{-1})) \end{aligned}$$

für beliebige $h_1, h_2, h \in H$, also ist K_γ eine Untergruppe von G .

Da offensichtlich für jedes Komplement K und jeden 1-Kozykel $\gamma \in Z^1(H, N)$ per Konstruktion $K = K_{f_k}$ und $\gamma = \gamma_{K_\gamma}$ gilt, handelt es sich bei der Zuordnung $K \mapsto \gamma_K$ um eine Bijektion.

‡‡

Definition 2.7 Die Voraussetzungen seien wie in Lemma 2.6. K sei ein beliebiges Komplement zu N^* und γ ein beliebiger 1-Kozykel aus $Z^1(H, N)$.

1. $\gamma_K : H \rightarrow N$ bezeichne den zu K gehörenden 1-Kozykel aus Lemma 2.6.
2. $K_\gamma \leq G$ bezeichne das zum 1-Kozykel γ gehörende Komplement zu N^* , gegeben durch

$$K_\gamma = \{(h, \gamma(h)) \mid h \in H\}.$$

Eine Konjugiertenklasse von Komplementen zu N^* läßt sich jedoch ebenso leicht durch die abelsche Gruppe $B^1(H, N)$ beschreiben.

Lemma 2.8 Die Voraussetzungen seien wie in Lemma 2.6. Zwei Komplemente K_1 und K_2 zu N^* sind genau dann konjugiert in G , wenn ein $n^* \in N^*$ existiert, so daß $K_1^{n^*} = K_2$ gilt. In diesem Fall gilt für die zu K_1 beziehungsweise K_2 gehörenden 1-Kozykel γ_1 und γ_2 , daß

$$\gamma_2 - \gamma_1 \in B^1(H, N).$$

Umgekehrt beschreibt jedes $\gamma \in \gamma_1 + B^1(H, N)$ ein zu K_1 konjugiertes Komplement.

Beweis: Es sei $K_1^g = K_2$ für ein $g \in G$. Da K_1 Komplement zu N^* ist, gilt $g \in G = KN^*$, also $g = kn^*$ für $n^* =: (0, n) \in N^*$, $k \in K_1$. Damit also

$$K_2 = K_1^g = K_1^{kn^*} = K_1^{n^*}.$$

Seien γ_1, γ_2 die zugehörigen 1-Kozykel, also

$$\begin{aligned} K_1^{n^*} &= \{(h, \gamma_1(h)) \mid h \in H\}^{n^*} \\ &= \{(1, -n)(h, \gamma_1(h))(1, n) \mid h \in H\} \\ &= \{(h, (-n)^{\alpha(h)} + \gamma_1(h) + n) \mid h \in H\} \\ &= K_2 \\ &= \{(h, \gamma_2(h)) \mid h \in H\}. \end{aligned}$$

Insbesondere gilt also $\gamma_2(h) = (-n)^{\alpha(h)} + \gamma_1(h) + n$ für beliebige $h \in H$. Damit

$$\gamma_2(h) - \gamma_1(h) = (-n)^{\alpha(h)} + n$$

für ein festes $n \in N$ und beliebige $h \in H$. Mit Lemma 2.5 folgt also nun $\gamma_2 - \gamma_1 \in B^1(H, N)$.

Es sei ein $\gamma \in \gamma_1 + B^1(H, N)$ gegeben, das heißt, es gibt ein $n \in N$, so daß $\gamma(h) = \gamma_1(h) + (-n)^{\alpha(h)} + n$ gilt. Damit ergibt sich

$$\begin{aligned} K_\gamma &= \{(h, \gamma(h)) \mid h \in H\} \\ &= \{(h, \gamma_1(h) + (-n)^{\alpha(h)} + n) \mid h \in H\} \\ &= \{(1, -n)(h, \gamma_1(h))(1, n) \mid h \in H\} \\ &= K_1^{(0, n)}. \end{aligned}$$

‡‡

Die beiden Lemmata 2.6 und 2.8 zusammen zeigen sofort

Satz 2.9 *Die Voraussetzungen seien wie in Lemma 2.6. Dann entsprechen die Konjugiertenklassen von Komplementen zu N^* in G eindeutig den Elementen der 1-Kohomologiegruppe $H^1(H, N)$, das heißt, den Restklassen von $Z^1(H, N)$ nach dem Teilraum $B^1(H, N)$. Jedem Repräsentanten $\gamma \in Z^1(H, N)$ einer Restklasse aus $H^1(H, N)$ wird dabei der Repräsentant*

$$K_\gamma = \{(h, \gamma(h)) \mid h \in H\}$$

einer Konjugiertenklasse von Komplementen zugeordnet.

2.3 Berechnung der 1-Kohomologiegruppe

Die in den vorhergehenden Abschnitten gemachten Sätze und Definitionen stellten keine Bedingungen in die Gruppe H . Im folgenden wollen wir uns auf endlich präsentierte Gruppen und elementar abelsche H -Module beschränken. Die folgenden Sätze sind immer dann anwendbar, wenn wir eine Präsentation für H kennen und die Operation von H auf N durch Matrizen gegeben ist.

Es sei jetzt H eine endlich präsentierte Gruppe mit einer gegebenen Präsentation

$$\langle h_1, \dots, h_l \mid r_i(h_1, \dots, h_l) = 1, i = 1, \dots, r \rangle,$$

N sei ein elementar abelscher H -Modul der Ordnung p^m zur Abbildung α mit Erzeugern b_1, \dots, b_m . G sei eine Erweiterung von H mit N . Dann existiert eine exakte Folge

$$\langle 1 \rangle \rightarrow N \xrightarrow{\mu} G \xrightarrow{\pi} H \rightarrow \langle 1 \rangle,$$

sowie ein Schnitt $\tau : H \rightarrow G$, das heißt, eine Abbildung mit $1_{H^\tau} = 1_G$ und $\tau \cdot \pi = id_h$. Dann gilt für $n \in N$, $h \in H$

$$(n^{\alpha(h)})^\mu = (n^\mu)^{h^\tau}.$$

Es soll nun untersucht werden, wann die Erweiterung G zerfällt und wie man in diesem Fall die Konjugiertenklassen von Komplementen zu N^μ beschreiben kann.

Satz 2.10 G , H und N seien wie oben angegeben. G zerfällt genau dann, wenn eine Funktion $f : \{h_1, \dots, h_l\} \rightarrow N$ existiert, so daß für alle $j \in \{1, \dots, r\}$

$$r_j(h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu) = 1$$

gilt. In diesem Fall ist ein Komplement gegeben durch

$$\langle h_i^\tau f(h_i)^\mu \mid i = 1, \dots, l \rangle \leq G.$$

Beweis: G zerfällt genau dann, wenn ein Komplement zu N^μ existiert.

“ \Leftarrow ”: Es existiere eine Funktion f mit $r_j(h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu) = 1$ für alle $j \in \{1, \dots, r\}$. Setze

$$\hat{H} = \langle h_i^\tau f(h_i)^\mu \mid i = 1, \dots, l \rangle \leq G.$$

Dann gilt $\hat{H}N^\mu = G$, denn die Faktorgruppe G/N^μ wird von den Elementen $h_i^\tau N^\mu$ für $i = 1, \dots, l$ erzeugt. Es gilt aber offensichtlich dann

$$\begin{aligned} G/N^\mu &= \langle h_1^\tau N^\mu, \dots, h_l^\tau N^\mu \rangle \\ &= \langle h_1^\tau f(h_1)^\mu N^\mu, \dots, h_l^\tau f(h_l)^\mu N^\mu \rangle. \end{aligned}$$

Also erzeugen die Elemente $h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu$ zusammen mit N^μ die Gruppe G .

Es sei $x \in \hat{H} \cap N^\mu$. Dann gilt $x = \prod (h_{i_j}^\tau f(h_{i_j})^\mu)^{e_j}$ für geeignete Exponenten $e_j \in \{-1, +1\}$. Da x in $N^\mu = \ker(\pi)$ liegt, gilt $x^\pi = 1$. Andererseits läßt sich nach Voraussetzung die Abbildung

$$\varphi : H \rightarrow \hat{H} : h_i \mapsto h_i^\tau f(h_i)^\mu$$

zu einem Homomorphismus fortsetzen, da die Bilder von h_i die Relationen von H erfüllen. Damit gilt also

$$\begin{aligned} 1 &= 1^\varphi = (x^\pi)^\varphi \\ &= \prod ((h_{i_j}^\tau)^\pi)^{e_j} = \prod (h_{i_j}^\varphi)^{e_j} \\ &= \prod (h_{i_j}^\tau f(h_{i_j})^\mu)^{e_j} = x, \end{aligned}$$

das heißt, $\hat{H} \cap N^\mu = \{1\}$. Somit ist \hat{H} ein Komplement zu N^μ und G eine zerfallende Erweiterung.

“ \Rightarrow ”: G zerfalle, das heißt, es existiert eine Untergruppe K mit $KN^\mu = G$ und $K \cap N^\mu = \{1\}$. Für alle h_i , $i \in \{1, \dots, l\}$ gibt es daher $n_i \in N$ und $k_i \in K$, so daß $h_i^\tau = k_i n_i^\mu \in G = KN^\mu$ gilt. Setze

$$f := (h_i \mapsto -n_i) : \{h_1, \dots, h_l\} \rightarrow N.$$

Für dieses f folgt dann für $j = 1, \dots, r$

$$r_j(h_i^\tau f(h_i)^\mu)^\pi = r_j(h_i^\tau)^\pi (f(h_i)^\mu)^\pi = r_j(h_i) = 1.$$

Also $r_j(h_i^\tau f(h_i)^\mu) \in \ker(\pi) = N^\mu$ und $r_j(h_i^\tau f(h_i)^\mu) = r_j(k_i) \in K$ und damit

$$r_j(h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu) \in K \cap N^\mu = \{1\}.$$

‡‡

Zum Beweis von Satz 2.10 hätte ein abelscher H -Modul N ausgereicht, wie das folgende Lemma jedoch zeigt, reduziert sich im Falle eines elementar abelschen H -Moduls die Berechnung einer Funktion f aus Satz 2.10 auf das Lösen eines inhomogenen Gleichungssystems. Da N elementar abelsch ist, kann N als ein \mathbf{Z}_p -Vektorraum aufgefaßt werden auf dem H linear operiert. Man kann also jede Gleichung der Form

$$n_1^{\alpha(h_1)} + \dots + n_t^{\alpha(h_t)} = n$$

für feste $h_i \in H$, $n \in N$ und Unbekannte $n_i \in N$ auffassen als lineare Gleichung in $t * m$ Unbekannten $x_{ij} \in \mathbf{Z}_p$ mit $n_i = b_1 x_{i1} + \dots + b_m x_{im}$.

Lemma 2.11 *Eine Funktion f aus Satz 2.10 läßt sich durch Lösen eines inhomogenen linearen Gleichungssystems mit $r \cdot m$ Gleichungen und $m \cdot l$ Unbekannten bestimmen.*

Beweis: Es sei ohne Einschränkung der Allgemeinheit $N = \mathcal{V}_{1 \times m}(\mathbf{Z}_p)$ und damit $\alpha : H \rightarrow \text{Aut}(N) = GL(m, p)$. Dann sind $m \cdot l$ Unbekannte $x_{ij} \in \mathbf{Z}_p$ gesucht, so daß für $f : \{h_1, \dots, h_l\} \rightarrow N$ mit $f(h_i) = (x_{i1}, \dots, x_{im})$

$$r_j(h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu) = 1$$

für $j = 1, \dots, r$ gilt. Sei $r = r_j$ ein beliebiger Relator. Es sei $r(y_1, \dots, y_l) = \prod_{j=1}^t y_{i_j}^{e_j}$ als Wort in den Erzeugern $\{y_1, \dots, y_l\}$ einer freien Gruppe für geeignete Exponenten $e_j \in \{-1, +1\}$ und Indizes $i_j \in \{1, \dots, l\}$. Es sei

$$GL(m, p) \ni B_j := \begin{cases} -\alpha(h_{i_j}^{-1}) & e_j = -1 \\ 1 & e_j = 1 \end{cases}$$

und $A_j := \alpha(h_{i_j}^{e_j}) \in GL(m, p)$. Mit dieser Setzung folgt dann

$$\begin{aligned} 0^\mu &= 1 \\ &= r(h_1^\tau f(h_1)^\mu, \dots, h_l^\tau f(h_l)^\mu) \\ &= \prod_{j=1}^t (h_{i_j}^\tau)^{e_j} (f(h_{i_j}) * B_j)^\mu \\ &= \left(\prod_{j=1}^t (h_{i_j}^\tau)^{e_j} \right) \cdot \left(\sum_{j=1}^t f(h_{i_j}) * (B_j \cdot \prod_{k=j+1}^t A_k) \right)^\mu, \\ &= \underbrace{r(h_1^\tau, \dots, h_l^\tau)}_{\in N^\mu} \cdot \left(\sum_{j=1}^t f(h_{i_j}) * (B_j \cdot \prod_{k=j+1}^t A_k) \right)^\mu. \end{aligned}$$

Da μ injektiv ist, folgt also

$$0 = r(h_1^\tau, \dots, h_l^\tau)^{\mu^{-1}} + \sum_{j=1}^t f(h_{i_j}) * (B_j \cdot \prod_{k=j+1}^t A_k),$$

also nach obiger Bemerkung ein inhomogenes System m linearer Gleichungen. $\ddagger\ddagger$

Der Beweis zu Satz 2.10 zeigt jedoch noch mehr.

Satz 2.12 *Die Voraussetzungen seien wie in Satz 2.10. G zerfalle und $L = f_p + L_{\text{hom}}$ sei die Lösungsgesamtheit des linearen Gleichungssystems aus Satz 2.10. Dann läßt sich jedem $f \in L$ eineindeutig ein $\gamma \in Z^1(H, N)$ mit $\gamma|_{\{h_1, \dots, h_l\}} = f - f_p$ zuordnen.*

Beweis:

1. Jeder 1-Kozykel γ aus $Z^1(H, N)$ ist festgelegt durch die Angabe der Bilder $\gamma(h_i)$, für $j = 1, \dots, l$. Denn wegen

$$\begin{aligned}\gamma(h_{i_1} h_{i_2}) &= \gamma(h_{i_1})^{\alpha(h_{i_2})} + \gamma(h_{i_2}), \\ \gamma(h_i^{-1}) &= -\gamma(h_i)^{\alpha(h_i^{-1})},\end{aligned}$$

sind mit $\gamma(h_i)$ auch die Bilder aller Produkte von Erzeugern h_i und h_i^{-1} festgelegt.

2. Einerseits ist

$$\hat{H} = \langle (h_i^\tau f_p(h_i)^\mu) \cdot \gamma(h_i)^\mu \mid i = 1, \dots, l \rangle$$

genau dann ein Komplement—wie der Beweis zu Satz 2.10 zeigt—, wenn die Funktion $f_p + \gamma|_{\{h_1, \dots, h_l\}}$ die Gleichungen aus Satz 2.10 löst. Andererseits ist \hat{H} nach Lemma 2.6 genau dann ein Komplement, wenn $\gamma|_{\{h_1, \dots, h_l\}}$ von einer Funktion γ aus $Z^1(H, N)$ —wie in Teil 1 beschrieben—kommt.

‡‡

Korollar 2.13 *Die Voraussetzungen seien wie in Satz 2.10. Jedes $\gamma \in Z^1(H, N)$ ist eindeutig festgelegt durch Angabe der Bilder $\gamma(h_1), \dots, \gamma(h_l)$ der Erzeuger von H .*

Zur Bestimmung der $Z^1(H, N)$ reicht es also nach Satz 2.12, ein System von $r * m$ linearen Gleichungen zu lösen, anstelle der $|H|^2 * m$ Gleichungen, welche Lemma 2.5 liefert. Wie der Beweis Teil 1 zu Satz 2.12 zeigt, sind die Elemente aus $Z^1(H, N)$ durch die Bilder von h_1, \dots, h_l festgelegt. Also gilt

Lemma 2.14 *Die Voraussetzungen seien wie in Satz 2.12 und es sei noch zusätzlich $N = \mathcal{V}_{1 \times m}(\mathbf{Z}_p)$. G zerfalle und K sei ein beliebiges Komplement zu N^μ . Eine $m \times m * l$ Matrix M sei gegeben durch*

$$M := (I - \alpha(h_1) \quad , \dots , \quad I - \alpha(h_l)).$$

Dann entspricht $B^1(H, N)$ dem Zeilenraum von M und die homogene Lösungsmenge der Gleichung $x \cdot M = 0$ dem Zentralisator $C_N(K)$ von K in N .

Beweis: Es sei $\beta : Z^1(H, N) \rightarrow \mathcal{V}_{1 \times m * l}(\mathbf{Z}_p)$ mit $\beta(\gamma) = (\gamma(h_1), \dots, \gamma(h_l))$ für $\gamma \in Z^1(H, N)$.

Dann ist β nach Korollar 2.13 injektiv. Es sei weiter U der Zeilenraum von M , dann gilt

$$\begin{aligned}U &= \{n * M \mid n \in N\} \\ &= \{(n * (I - \alpha(h_1)), \dots, n * (I - \alpha(h_l))) \mid n \in N\} \\ &= \{\gamma^\beta \mid \gamma \in C^n(H, N), \exists n \in N : \forall h \in H : \gamma(h) = n * (I - \alpha(h))\} \\ &= \{\gamma^\beta \mid \gamma \in B^1(H, N)\} \\ &= (B^1(H, N))^\beta.\end{aligned}$$

Also ist $\beta|_{B^1(H,N)} : B^1(H,N) \rightarrow U$ surjektiv und damit bijektiv.

Weiter sei $n \in N$ eine Lösung der homogenen Gleichung $x \cdot M = 0$, das heißt, $0 = n * (1 - \alpha(h_i)) = (-n) * \alpha(h_i) + n$ für $i = 1, \dots, l$. Dann ist für beliebiges $i \in \{1, \dots, l\}$

$$\begin{aligned}
(h_i^\tau f_p(h_i)^\mu \gamma(h_i)^\mu)^{n^\mu} &= (h_i^\tau)^{n^\mu} \underbrace{(f_p(h_i)^\mu \gamma(h_i)^\mu)}_{\in N, \text{ abelsch!}} \\
&= (-n)^\mu (h_i^\tau)^{n^\mu} (f_p(h_i)^\mu \gamma(h_i)^\mu) \\
&= h_i^\tau (-n^\mu)^{h_i^\tau} n^\mu (f_p(h_i)^\mu \gamma(h_i)^\mu) \\
&= h_i^\tau \underbrace{(-n * \alpha(h_i) + n)^\mu}_{=1} (f_p(h_i)^\mu \gamma(h_i)^\mu) \\
&= (h_i^\tau f_p(h_i)^\mu \gamma(h_i)^\mu).
\end{aligned}$$

Also zentralisiert n jeden Erzeuger von K , das heißt, $n \in C_{N^\mu}(K)$. Andererseits gilt für jedes $n \in C_{N^\mu}(K)$, daß

$$\begin{aligned}
0 &= ((-n^\mu)^{(h_i^\tau f_p(h_i)^\mu \gamma(h_i)^\mu)} (n^\mu))^{\mu^{-1}} \\
&= -n * \alpha(h_i) + n.
\end{aligned}$$

‡‡

Zum Schluß dieses Abschnitts noch ein für folgende Kapitel nützliches Lemma.

Lemma 2.15 *Die Voraussetzungen und Bezeichnungen seien wie in Lemma 2.14. Es seien zwei konjugierte Komplemente K_1 und K_2 zu N^μ gegeben. Für diese Komplemente sei der Vektor $v \in \mathcal{V}_{1 \times m * l}(\mathbf{Z}_p)$ gegeben durch*

$$v := (\gamma_{K_2}(h_1) - \gamma_{K_1}(h_1), \dots, \gamma_{K_2}(h_l) - \gamma_{K_1}(h_l)).$$

Dann existiert eine Lösung $n \in N$ von $v = x \cdot M$ und es gilt $K_1^{n^\mu} = K_2$.

Beweis: Es sei n eine Lösung von $v = x \cdot M$. Dann gilt also $\gamma_{K_2}(h_i) = n^{1-\alpha(h_i)} + \gamma_{K_1}(h_i)$ für alle $i \in \{1, \dots, l\}$. Damit ergibt sich aber

$$\begin{aligned}
K_1^{n^\mu} &= \langle h_i^\tau \cdot f_p(h_i)^\mu \gamma_{K_1}(h_i)^\mu \mid i = 1, \dots, l \rangle^{n^\mu} \\
&= \langle (h_i^\tau)^{n^\mu} \cdot f_p(h_i)^\mu \gamma_{K_1}(h_i)^\mu \mid i = 1, \dots, l \rangle \\
&= \langle h_i^\tau \cdot (n * (1 - \alpha(h_i)) + \gamma_{K_1}(h_i))^\mu \cdot f_p(h_i)^\mu \mid i = 1, \dots, l \rangle \\
&= \langle h_i^\tau \cdot \gamma_{K_2}(h_i)^\mu f_p(h_i)^\mu \mid i = 1, \dots, l \rangle \\
&= K_2.
\end{aligned}$$

Sind umgekehrt K_1 und K_2 konjugiert, so existiert nach Lemma 2.8 ein $n \in N$ mit $K_1^{n^\mu} = K_2$. Obige Rechnung zeigt dann, daß dieses n die Gleichung löst. $\ddagger\ddagger$

2.4 Reduktion der Unbekannten

Es sei H eine Gruppe mit einer endlichen Präsentation wie im vorhergehenden Abschnitt. Die Größe des zu lösenden Gleichungssystems zur Berechnung der 1-Kohomologiegruppe hängt dann nach Lemma 2.11 von der Anzahl der Relationen und Unbekannten ab. Diese lassen sich auf zwei Weisen reduzieren.

Falls eine endliche p' -Untergruppe $H_{p'}$ von H bekannt ist und durch $\{h_{i_1}, \dots, h_{i_{l'}}\}$ erzeugt wird, so läßt sich die Unzahl der Unbekannten des inhomogenen Gleichungssystems aus Lemma 2.11 von $m * l$ auf $m * (l - l')$ reduzieren. Denn es gilt

Satz 2.16 *Die Voraussetzungen seien wie in 2.10. Zusätzlich sei eine endliche p' -Untergruppe $H_{p'}$ von H bekannt und diese werden von $\{h_{i_1}, \dots, h_{i_{l'}}\}$ erzeugt. Dann existiert ein Schnitt $\tau : H \rightarrow G$, so daß $\{h_{i_1}^\tau, \dots, h_{i_{l'}}^\tau\}$ eine p' -Untergruppe von G erzeugt. Es sei*

$$\mathcal{V} = \{\gamma \in C^1(H, N) \mid \gamma(h_{i_j}) = 0 \forall j \in \{1, \dots, l'\}\}.$$

Wenn G eine zerfallende Erweiterung ist, dann existiert auch eine Lösung f aus Satz 2.10 mit $f(h_{i_j}) = 0$ für $j \in \{1, \dots, l'\}$ und es gilt

$$(Z^1(H, N) \cap \mathcal{V}) / (B^1(H, N) \cap \mathcal{V}) \cong H^1(H, N)$$

sowie $Z^1(H, N) \cap \mathcal{V} + B^1(H, N) = Z^1(H, N)$.

Beweis: Es sei G^* das vollständige Urbild von $H_{p'}$ in G . Dann ist N^μ ein p -Normalteiler von G^* und G^*/N^μ hat eine zu p teilerfremde Ordnung, das heißt, N^μ ist ein Hall-Normalteiler von G^* . Nach dem Satz von Schur-Zassenhaus besitzt aber N^μ ein Komplement $K_{p'}$ in G^* . Nach Satz 2.10 existiert dann einen Schnitt $\tau' : H_{p'} \rightarrow G^*$, so daß $\{h_{i_1}^{\tau'}, \dots, h_{i_{l'}}^{\tau'}\}$ das Komplement $K_{p'}$ zu N^μ in G^* erzeugt. Dieser Schnitt läßt sich zu einem Schnitt τ für H fortsetzen.

Da aus der letzten Aussage sofort die anderen beiden folgen, sei nun vorausgesetzt, daß G zerfällt, und γ sei ein beliebiger 1-Kozykel aus $Z^1(H, N)$. Das zu γ gehörende Komplement K_γ enthält die von $\{h_{i_1}^\tau \gamma(h_{i_1})^\mu, \dots, h_{i_{l'}}^\tau \gamma(h_{i_{l'}})^\mu\}$ erzeugte Untergruppe K^* . Da nach Konstruktion $\tau|_{H_{p'}}^{G^*} = \tau'$ gilt, folgt damit $(K^*)^\pi = H_{p'}$. Also ist insbesondere K^* ein Komplement zu N^μ in G^* und damit nach dem Satz von Schur-Zassenhaus konjugiert zu $K_{p'}$. Es sei $g \in G^*$ ein konjugierendes Element, das heißt, $K_{p'} = (K^*)^g$. Damit enthält dann K_γ^g die Untergruppe $K_{p'}$. Dann folgt aber per Konstruktion für den zum Komplement K_γ^g gehörenden 1-Kozykel $\gamma^g = \gamma_{K_\gamma^g}$, daß $\gamma^g \in \mathcal{V}$ und $\gamma - \gamma^g \in B^1(H, N)$. Dies zeigt aber schon

$$Z^1(H, N) \cap \mathcal{V} + B^1(H, N) = Z^1(H, N).$$

Es bleibt die Isomorphie der Vektorräume zu zeigen. Für Vektorräume gilt der Isomorphiesatz, das heißt, es gilt

$$\begin{aligned} (Z^1(H, N) \cap \mathcal{V}) / (B^1(H, N) \cap \mathcal{V}) &\cong (Z^1(H, N) \cap \mathcal{V}) / (B^1(H, N) \cap (Z^1(H, N) \cap \mathcal{V})) \\ &\cong (Z^1(H, N) \cap \mathcal{V} + B^1(H, N)) / B^1(H, N) \\ &\cong H^1(H, N), \end{aligned}$$

und damit die Aussage. ‡‡

Eine andere Möglichkeit zur Reduktion der Unbekannten ist gegeben, falls ein kleineres Erzeugendensystem für H bekannt ist. Denn es gilt offensichtlich

Satz 2.17 *Die Voraussetzungen seien wie in 2.10. Es sei weiter eine Teilmenge $E = \{i_1, \dots, i_{l'}\} \subset \{1, \dots, l\}$ bekannt, so daß $\{h_{i_1}, \dots, h_{i_{l'}}\}$ die Gruppe H erzeugt, sowie Worte w_s mit $h_s = w_s(h_{i_1}, \dots, h_{i_{l'}})$ für $s \notin E$. Dann läßt sich die 1-Kohomologiegruppe durch Lösen eines Gleichungssystems mit $r * m$ Gleichungen und $m * (l - l')$ Unbekannten bestimmen.*

Beweis: Tietze-Transformation für die Präsentation von H . ‡‡

Im Falle der endlichen, polyzyklischen Gruppen läßt sich meist ein kleineres Erzeugendensystem berechnen—siehe Abschnitt 1.3, jedoch enthält die Präsentation, welche aus der Potenz-Kommutator Präsentation entsteht, im allgemeinen wesentlich längere Worte. Es ist daher besser, die Tietze-Transformation erst implizit während der Berechnung der 1-Kohomologiegruppe durchzuführen. Dazu seien die Bezeichnungen wie im Beweis zu Lemma 2.11. Jede Gleichung der Form $h_s = w_s(h_{i_1}, \dots, h_{i_{l'}})$ liefert dann analog Lemma 2.11 ein Gleichungssystem

$$f(h_s) = \underbrace{((h_s^\tau)^{-1} w_s(h_{i_1}^\tau, \dots, h_{i_{l'}}^\tau))^\mu}_{=: n_s \in N} + \sum_{j \in E} f(h_j) * C_{sj}$$

für geeignete Matrizen C_{sj} . Wie im Beweis von Lemma 2.11 sei $r(y_1, \dots, y_l) = \prod_{j=1}^l y_{i_j}^{e_j}$ als Wort in den Erzeugern $\{y_1, \dots, y_l\}$ einer freien Gruppe für geeignete Exponenten $e_j \in \{-1, +1\}$ und Indizes $i_j \in \{1, \dots, l\}$ und es sei

$$GL(m, p) \ni B_j := \begin{cases} -\alpha(h_{i_j}^{-1}) & e_j = -1 \\ 1 & e_j = 1 \end{cases}$$

und $A_j := \alpha(h_{i_j}^{e_j}) \in GL(m, p)$. Mit dieser Setzung folgt dann insgesamt

$$0 = r(h_1^\tau, \dots, h_l^\tau)^{\mu^{-1}} + \sum_{j=1}^t f(h_{i_j}) * (B_j \cdot \prod_{k=j+1}^t A_k)$$

$$\begin{aligned}
&= \left(r(h_1^\tau, \dots, h_l^\tau)^{\mu^{-1}} + \sum_{i_j \notin E} n_{i_j} * (B_j \cdot \prod_{k=j+1}^t A_k) \right) + \\
&\quad \sum_{i_j \in E} f(h_{i_j}) * (B_j \cdot \prod_{k=j+1}^t A_k) + \sum_{i_j \notin E} \sum_{k \in E} f(h_k) * (C_{i_j, k} \cdot B_j \cdot \prod_{k=j+1}^t A_k).
\end{aligned}$$

Man beachte, daß die Summen schon während der Berechnung zusammengefaßt werden können.

2.5 Die H^1 einer endlichen, polyzyklischen Gruppe

Nachdem im vorhergehenden Abschnitt Möglichkeiten zur Berechnung der 1-Kohomologiegruppe im Falle einer endlich präsentierten Gruppe H aufgezeigt wurden, ziehen wir uns nun auf den Fall einer endlichen, polyzyklischen Gruppe H mit einer Potenz-Kommutator Präsentation zurück. Es sei also H gegeben durch

$$H = \langle h_1, \dots, h_l \mid h_i^{p_i} = w_{ii}(h_k) \forall 1 \leq i \leq l, [h_j, h_i] = w_{ji}(h_k) \forall 1 \leq i < j \leq l \rangle$$

für fest vorgebene Primzahlen p_i und Worte

$$\begin{aligned}
w_{ii}(x_1, \dots, x_l) &= \prod_{k=i+1}^l x_k^{\nu_{ii}^{(k)}}, \\
w_{ji}(x_1, \dots, x_l) &= \prod_{k=i+1}^l x_k^{\nu_{ji}^{(k)}}.
\end{aligned}$$

N sei der \mathbf{Z}_p -Vektorraum $\mathcal{V}_{1 \times m}(\mathbf{Z}_p)$ und zur Abkürzung sei $H_i := \alpha(h_i) \in GL(m, p)$ gesetzt. Die Elemente aus $Z^1(H, N)$ und $B^1(Z, N)$ werden als Tupel $(n_1, \dots, n_l) \in \mathcal{V}_{1 \times ml}(\mathbf{Z}_p)$ abgespeichert, so daß n_i das Bild des Erzeugers h_i unter einem 1-Kozykel ist, da nach Korollar 2.13 die 1-Kozykel dadurch eindeutig identifiziert werden können.

Das Gleichungssystem zur Bestimmung der $Z^1(H, N)$ wurde schon in Lemma 2.11 aufgestellt. Danach gehört zu jedem Relator r ein Gleichungssystem der Form

$$0 = n^{(r)} + n_1 * R_1^{(r)} + \dots + n_l * R_l^{(r)}$$

für Matrizen $R_i \in \mathcal{V}_{m \times m}(\mathbf{Z}_p)$. Im Fall einer Potenz-Kommutator Präsentation können beim Separieren des H und N -Anteils folgende Situationen vorkommen. Es sei $i, j \in \{1, \dots, l\}$ und $i \neq j$, A sei eine schon berechnete Matrix aus $\mathcal{V}_{m \times m}(\mathbf{Z}_p)$ und a sei eine positive ganze Zahl echt kleiner p_i .

$$\dots (n_i * A)^\mu (h_i^\tau n_i^\mu)^a \dots = \dots (n_i * A)^\mu h_i^\tau n_i^\mu \dots h_i^\tau n_i^\mu \dots$$

$$\begin{aligned}
&= \cdots (n_i * A)^\mu h_i^{\tau a} \left(n_i * \sum_{k=0}^{a-1} H_i^k \right)^\mu \cdots \\
&= \cdots h_i^{\tau a} \left(n_i * (A \cdot H_i^a + \sum_{k=0}^{a-1} H_i^k) \right)^\mu \cdots, \\
\cdots (n_i * A)^\mu (h_i^\tau n_i^\mu)^{-a} \cdots &= \cdots (n_i * A)^\mu n_i^{\mu-1} h_i^{\tau-1} \cdots n_i^{\mu-1} h_i^{\tau-1} \cdots \\
&= \cdots (n_i * A)^\mu h_i^{\tau-a} \left(n_i * \left(- \sum_{k=1}^a H_i^{-k} \right) \right)^\mu \cdots \\
&= \cdots h_i^{\tau-a} \left(n_i * \left((A - \sum_{k=0}^{a-1} H_i^k) \cdot H_i^{-a} \right) \right)^\mu \cdots, \\
\cdots (n_j * A)^\mu (h_i^\tau n_i^\mu)^a \cdots &= \cdots (h_i^\tau n_i^\mu)^a (n_j * (A H_i^a))^\mu \cdots, \\
\cdots (n_j * A)^\mu (h_i^\tau n_i^\mu)^{-a} \cdots &= \cdots (h_i^\tau n_i^\mu)^{-a} (n_j * (A H_i^{-a}))^\mu \cdots.
\end{aligned}$$

Folgender Algorithmus berechnet gemäß obigen Überlegungen für einen gegebenen Relator $r = x_i^{p_i}/w_{ii}$ oder $r = [x_j, x_i]/w_{ji}$ und eine Zahl t die Matrix R_t .

Algorithmus 2.1. (Gleichungsmatrix)

Eingabe : • Matrizen $[H_1, \dots, H_l] = M$, welche die Operation von h_1, \dots, h_l auf N beschreiben.

• Ein Relator $r(x_1, \dots, x_l) = \prod_{k=1}^t x_{j_k}^{\nu_k}$.

• Eine Nummer s , für welche die Matrix R_s berechnet wird.

Ausgabe : Die Matrix R_s des Gleichungssystems zum Relator r , wie oben beschrieben.

```

EquationMatrix := function( M, r, s )
  R := idmat;
  for i from 1 to t do
    if j_i = s then
      if nu_i > 0 then
        R := R * M[j_i]^nu_i + sum_{k=0}^{nu_i-1} M[j_i]^k;
      else
        R := (R - sum_{k=0}^{-nu_i-1} M[j_i]^k) * M[j_i]^nu_i;
      fi;
    else
      R := R * M[j_i]^nu_i;
    fi;
  od;

```

```

    return R;
end.

```

Die Gruppe der 1-Koränder läßt sich, falls eine zerfallende Erweiterung von H mit N vorliegt, nach Lemma 2.14 durch den Zeilenraum einer Matrix bestimmen. Dies geschieht durch

Algorithmus 2.2. (Gruppe der Einskoränder)

Eingabe : • Eine endliche, polyzyklische Gruppe G mit AG-System $[g_1, \dots, g_{l+m}]$ mit einem elementar abelschen Normalteiler N mit CGS $[n_1, \dots, n_m]$, so daß G eine zerfallende Erweiterung von G/N mit N ist.

Ausgabe : Eine Folge $B1 \subseteq \mathcal{V}_{1 \times ml}(\mathbf{Z}_p)$, welche eine Basis für die Gruppe der 1-Koränder bildet und eine Untergruppe CN mit CGS, welche den Zentralisator $C_N(K)$ für ein beliebiges Komplement K zu N beschreibt.

```

OneCoboundaries := function( G, N )
    M := Matrizen, welche die Operation von G/N auf N beschreiben;
    L := ( 1 - M[1]  ...  1 - M[l] );
    B1 := Basis der Zeilen von L;
    CN := Nullspace( L );
    return B1, CN;
end.

```

In nachfolgenden Kapiteln wird noch ein Algorithmus gebraucht, welcher zu konjugierten Untergruppen ein konjugierendes Element findet. Dies kann nach Lemma 2.15 berechnet werden.

Algorithmus 2.3. (Konjugierendes Element)

Eingabe : • Eine endliche, polyzyklische Gruppe G mit AG-System $[g_1, \dots, g_{l+m}]$ mit einem elementar abelschen Normalteiler N mit CGS $[n_1, \dots, n_m]$, so daß G eine zerfallende Erweiterung von G/N mit N ist.

• Ein Komplement K zu N mit CGS $[k_1, \dots, k_l]$.

• Ein zu K unter N konjugiertes Komplement K' mit CGS $[k'_1, \dots, k'_l]$.

Ausgabe : Ein Element $n \in N$ mit $K^n = K'$.

```

ConjugatingElement := function( G, N, K, K' )
    M := Matrizen, welche die Operation von G/N auf N beschreiben;
    L := ( 1 - M[1]  ...  1 - M[l] );
    R := ( k_1^{-1} * k'_1  ...  k_l^{-1} * k'_l );
    n := Solution( L, R );
    return n;
end.

```

Mit den beiden Algorithmen 2.1 und 2.2 läßt sich nun die 1-Kohomologiegruppe wie folgt berechnen.

Algorithmus 2.4. (Einskohomologiegruppe)

- Eingabe* :
- Eine endliche, polyzyklische Gruppe G mit AG-System $[g_1, \dots, g_{l+m}]$ mit einem elementar abelschen p -Normalteiler N mit CGS $[n_1, \dots, n_m]$.
 - Das System Rel_s der Potenz-Kommutator Relatoren für $H = G/N$ $\{r_1, \dots, r_{l+(l-1)l/2}\}$.

Ausgabe : Eine Folge $Z1 \subseteq \mathcal{V}_{1 \times ml}(\mathbf{Z}_p)$, welche eine Basis für die Gruppe der 1-Kozykel bildet, eine Folge $B1 \subseteq Z1$, welche eine Basis für die Gruppe der 1-Koränder bildet, ein Tupel f , wie in Satz 2.10 beschrieben, und eine Untergruppe CN mit CGS, welche $C_N(K)$ für ein beliebiges Komplement K zu N beschreibt, oder `false`, falls G eine nicht zerfallende Erweiterung ist.

```

OneCocycles := function( G, N, Rel_s )
  H := Repräsentanten für ein AG-System von G/N;
  L := ();
  R := ();
  for r ∈ Rel_s do
    L1 := ();
    for j ∈ [1..l] do
      R1 := EquationMatrix( M, r, j );
      L1 := ( L1
             R1 );
    od;
    L := ( L  L1 );
    R := ( R  -r(H[1], ..., H[l]) );
  od;
  f := Solution( L, R );
  if f = false then
    return false ;
  fi;
  Z1 := Nullspace( L );
  tmp := OneCoboundaries( G, N );
  B1 := tmp.B1;
  CN := tmp.CN;
  return f, Z1, B1, CN;
end.

```

Die im obigen Algorithmus beschriebene Situation, das heißt, eine endliche, polyzyklische

Gruppe mit einem elementar abelschen Normalteiler, wird in den folgenden Kapiteln noch weiter untersucht. Dafür ist folgendes Lemma hilfreich.

Lemma 2.18 *G sei eine endliche, polyzyklische Gruppe mit kanonischen Erzeugendensystem (g_1, \dots, g_{l+m}) und N ein Normalteiler von G mit einem kanonischen Erzeugendensystem $(g_{l+1}, \dots, g_{l+m})$. Es sei $H = \langle h_1, \dots, h_l \rangle$ eine endliche, polyzyklische Gruppe mit oben angegebener Potenz-Kommutator Präsentation, so daß $H \cong G/N$ gilt und es einen Schnitt $\tau : H \rightarrow G$ gibt mit $h_i^\tau = g_i$ für $i \in \{1, \dots, l\}$. Desweiteren sei G eine zerfallende Erweiterung und $f : \{h_1, \dots, h_l\} \rightarrow N$ eine Funktion, welche wie in Satz 2.12 beschrieben zu einem beliebigen Komplement*

$$K = \langle g_i f(h_i) \mid i = 1, \dots, l \rangle$$

gehört. Dann ist $(g_1 f(h_1), \dots, g_l f(h_l))$ ein kanonisches Erzeugendensystem für dieses Komplement.

Beweis: Es sei $k_i := g_i f(h_i)$ für $i = 1, \dots, l$. Nach Satz 2.10 erfüllen diese k_i die Potenz-Kommutator Relationen der h_i . Also existiert ein Epimorphismus $\varphi : H \rightarrow K$ mit $\varphi(h_i) = k_i$, wegen der Endlichkeit von $H \cong K$ ist φ ein Isomorphismus. Dann ist φ mit der Tiefenzuordnung normierter Worte vertauschbar.

Es reicht also zu zeigen, daß sich $k_i^{p_i}$ und $[k_j, k_i]$ für $i, j \in \{1, \dots, l\}$, $i < j$ in den Erzeugern k_{i+1}, \dots, k_l ausdrücken lassen. Dieses zeigt aber sofort der Isomorphismus φ , denn die h_1, \dots, h_l bilden nach Voraussetzung eine Potenz-Kommutator Präsentation, erfüllen also obige Bedingung. $\dagger\dagger$

Wie man am Beispiel $Z_2^4 = \langle g_1, g_2, g_3, g_4 \mid g_i^2 = [g_i, g_j] = 1 \rangle$ und dem Normalteiler $N = \langle g_1, g_2 \rangle$ sieht, kann auf die Bedingung an N in Lemma 2.18, daß also N von den letzten der g_i erzeugt wird, nicht verzichtet werden, denn zum Beispiel $\langle g_2 g_3, g_2 g_4 \rangle$, $\langle g_3, g_2 g_4 \rangle$ sind Komplemente zu N , aber weder $(g_2 g_3, g_2 g_4)$ noch $(g_3, g_2 g_4)$ sind kanonische Erzeugendensysteme, sondern müssen noch sortiert und normiert werden.

2.6 Ein Beispiel: $H^1(S_3, Z_2^3)$

Es sei jetzt S_4 die symmetrische Gruppe auf vier Punkten und G das direkte Produkt $S_4 \times Z_2$ von S_4 mit der zyklischen Gruppe der Ordnung 2. N sei der dreidimensionale Vektorraum über dem Körper mit zwei Elementen. Dann existiert ein Monomorphismus μ , welcher N auf das direkte Produkt $V_4 \times Z_2$ der normalen Kleinschen Vierergruppe V_4 von S_4 mit der zentralen Z_2 abbildet. Die Faktorgruppe $G/N^\mu = H$ ist dann isomorph zu der symmetrischen Gruppe S_3 auf 3 Punkten. Für dieses Beispiel soll nun die 1-Kohomologiegruppe berechnet werden.

G sei durch folgende Potenz-Kommutator Präsentation gegeben.

$$G = \left\langle a, b, c, d, e ; \begin{array}{l} a^2 = b^3 = c^2 = d^2 = e^2 = [c, d] = [d, a] = 1, \\ [b, a] = b, [c, a] = d, [c, b] = cd, [d, b] = c \\ [a, e] = [b, e] = [c, e] = [d, e] = 1 \end{array} \right\rangle.$$

Dann ist die Einbettung von N gegeben durch $(1, 0, 0)^\mu = c$, $(0, 1, 0)^\mu = d$ und $(0, 0, 1)^\mu = e$. Es sei $H = \langle \underline{a}, \underline{b} \rangle$ mit $\underline{a} := aN^\mu$, $\underline{b} := bN^\mu$. Desweiteren sei ein Schnitt $\tau : H \rightarrow G$ mit $\underline{a}^\tau = ac$ und $\underline{b}^\tau = b$ gegeben. Dann sind die Potenz-Kommutator Relatoren $r_1(\underline{a}, \underline{b})$, $r_2(\underline{a}, \underline{b})$ und $r_3(\underline{a}, \underline{b})$ für H bezüglich \underline{a} und \underline{b}

$$\begin{aligned} r_1(y_1, y_2) &= y_1^2, \\ r_2(y_1, y_2) &= y_2^3, \\ r_3(y_1, y_2) &= y_2^{-1} y_1^{-1} y_2 y_1 y_2^{-1}, \end{aligned}$$

und $\alpha : H \rightarrow \text{Aut}(N)$ ist gegeben durch

$$A := \alpha(\underline{a}) = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad B := \alpha(\underline{b}) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Zuerst stellen wir gemäß Algorithmus 2.1 die zu den Relatoren gehörenden Gleichungsmatrizen auf.

$$\begin{aligned} (0, 0, 0)^\mu &= (\underline{a}^\tau n_a^\mu)^2 \\ &= \underline{a}^\tau \cdot n_a^\mu \cdot \underline{a}^\tau \cdot n_a^\mu \\ &= \underbrace{(ac)^2}_{=d} \cdot (n_a * (A + 1))^\mu \\ &= ((0, 1, 0) + n_a \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix})^\mu, \\ (0, 0, 0)^\mu &= (\underline{b}^\tau n_b^\mu)^3 \\ &= \underline{b}^\tau \cdot n_b^\mu \cdot \underline{b}^\tau \cdot n_b^\mu \cdot \underline{b}^\tau \cdot n_b^\mu \\ &= (n_b * (B^2 + B + 1))^\mu \\ &= (n_b \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix})^\mu, \\ (0, 0, 0)^\mu &= (-n_b)^\mu \cdot \underline{b}^{-1\tau} \cdot (-n_a)^\mu \cdot \underline{a}^{-1\tau} \cdot \underline{b}^\tau \cdot n_b^\mu \cdot \underline{a}^\tau \cdot n_a^\mu \cdot (-n_b)^\mu \cdot \underline{b}^{-1\tau} \\ &= \underbrace{(b^{-1} c^{-1} a^{-1} b a c b^{-1})}_{=c} \cdot \\ &\quad (n_a * (-A^{-1} B A B^{-1} + B^{-1}))^\mu \cdot \\ &\quad (n_b * (-B^{-1} A^{-1} B A B^{-1} + A B^{-1} - B^{-1}))^\mu \\ &= ((1, 0, 0) + n_a \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + n_b \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix})^\mu \end{aligned}$$

für Unbekannte $n_a = (x_1, x_2, x_3), n_b = (x_4, x_5, x_6) \in N$. Das zum Bestimmen der 1-Kozykeln zu lösende Gleichungssystem lautet also nach Algorithmus 2.4 und Lemma 2.11

$$(x_1, x_2, x_3, x_4, x_5, x_6) \cdot \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} = (0, 1, 0, 0, 0, 0, 1, 0, 0).$$

Die Lösungsgesamtheit dieses inhomogenen Systems ist

$$(1, 0, 0, 0, 0, 0) + \langle (0, 1, 0, 0, 1, 0), (0, 0, 1, 0, 0, 0), (0, 0, 0, 1, 0, 0) \rangle.$$

Dies entspricht den vier Punktstabilisatoren der symmetrischen Gruppe auf vier Punkten zusammen mit den Punktstabilisatoren, bei denen der Erzeuger der Ordnung 2 mit dem Erzeuger e der zentralen Z_2 von G multipliziert wurde. Nach Satz 2.10 werden diese Untergruppen erzeugt durch

$$\begin{aligned} s_3^{(1)} &= \langle a, b \rangle, & s_3^{(5)} &= \langle ae, b \rangle, \\ s_3^{(2)} &= \langle ad, bd \rangle, & s_3^{(6)} &= \langle ade, bd \rangle, \\ s_3^{(3)} &= \langle ad, bc \rangle, & s_3^{(7)} &= \langle ade, bc \rangle, \\ s_3^{(4)} &= \langle ad, bcd \rangle, & s_3^{(8)} &= \langle ade, bcd \rangle. \end{aligned}$$

Die 1-Koränder entsprechen nach Lemma 2.14 dem Zeilenraum von

$$M = (I - A, I - B) = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

und der Zentralisator ist die Lösungsmenge von $xM = 0$. Damit ergibt sich sofort, daß

$$Z^1(H, N)/B^1(H, N) = (0, 0, 1, 0, 0, 0) + B^1(H, N).$$

Also ist die 1-Kohomologiegruppe $H^1(S_3, V_4)$ zyklisch von der Ordnung zwei und Repräsentanten für die Konjugiertenklassen von Komplementen sind

$$\begin{aligned} s_3^{(1)} &= \langle a, b \rangle, \\ s_3^{(5)} &= \langle ae, b \rangle. \end{aligned}$$

Literatur

Eine weiterführende Beschreibung der Kohomologiegruppe findet sich in [Hup67]. Die Idee der Berechnung der 1-Kohomologiegruppe durch lineare Gleichungssysteme lehnt sich an die Methoden im SQ zur Berechnung der 2-Kohomologiegruppe an, siehe [Weg89].

Kapitel 3

Der Normalisator

In diesem Kapitel wird ein Algorithmus zur Berechnung des Normalisators einer Untergruppe einer endlichen, polyzyklischen Gruppe vorgestellt. Dazu bezeichne ab jetzt G eine endliche, polyzyklische Gruppe mit Potenz-Kommutator Präsentation in den Erzeugenden g_1, \dots, g_t . Es sei weiter eine Normalreihe

$$G = N_1 \triangleright N_2 \triangleright \dots \triangleright N_r \triangleright N_{r+1} = \{1\}$$

mit elementar abelschen Faktoren bekannt, welche von der Kompositionsreihe

$$G = \langle g_1, \dots, g_t \rangle \triangleright \langle g_2, \dots, g_t \rangle \triangleright \dots \triangleright \langle g_t \rangle \triangleright \langle 1 \rangle$$

verfeinert wird. Dabei seien die Faktoren N_i/N_{i+1} für $1 \leq i \leq r$ elementar abelsche q_i -Gruppen für Primzahlen q_i . Eine Untergruppe U von G sei durch ein kanonisches Erzeugendensystem $\{u_1, \dots, u_s\}$ bezüglich $\{g_1, \dots, g_t\}$ gegeben. Für diese Untergruppe U soll der Normalisator $N_G(U)$ berechnet werden.

3.1 Der Normalisator als Stabilisator

Der Normalisator $N_G(U)$ kann als Stabilisator in G des Punktes U unter der Konjugation von Untergruppen aufgefaßt werden. Dies ist schon in [LNS84] beschrieben worden. Wir wollen hier nur kurz auf zwei unterschiedliche Ansätze eingehen.

Zum einen ist möglich, den Stabilisator zu berechnen ohne die bei dieser Berechnung entstehenden, zu U konjugierten Untergruppen abzuspeichern, indem man anstelle der konjugierten Untergruppen U' des Orbits nur Elemente $g_{U'} \in G$ abspeichert, so daß $U^{g_{U'}} = U'$ gilt. Dies entspricht Algorithmus 1.3 zur Berechnung des Stabilisators. Man braucht also eine Funktion, welche entscheidet, ob ein Element $g \in G$ die Untergruppe U normalisiert. Diese Funktion liefert

Lemma 3.1 *Es seien G und U wie oben. Dann normalisiert ein Element $g \in G$ die Untergruppe U genau dann, wenn*

$$[u_i, g] \in U \quad \forall i = 1, \dots, s$$

gilt.

Es ist im allgemeinen effektiver, $[u_i, g] \in U$ statt $u_i^g \in U$ zu überprüfen, da hierbei zentralisierte Elemente auf 1 abgebildet werden.

Zum anderen läßt sich der Stabilisator berechnen, indem man für alle konjugierten Untergruppen ein kanonisches Erzeugendensystem abspeichert. Dies entspricht Algorithmus 1.2 zur Berechnung des Stabilisators. Ein kanonisches Erzeugendensystem liefert dabei Lemma 1.14.

3.2 Reduktion der Orbitlänge

Die Methoden des vorhergehenden Abschnitts sind im allgemeinen nicht direkt anwendbar, da insgesamt $[G : N_G(U)]$ konjugierte Untergruppen zu U beziehungsweise konjugierende Elemente für diese abgespeichert werden müssen. Es ist daher nötig, die auftretenden Orbitlängen zu reduzieren. Die in diesem Abschnitt vorgestellten Methoden folgen dabei dem in [GS92] vorgestellten Algorithmus zur Berechnung des Normalisators einer Untergruppen einer endlichen polyzyklischen Gruppe.

Es sei $N = N_r$ und $M = N_i$ für ein festes $i \in \{1, \dots, r-1\}$. Eine erste Reduktion ermöglicht

Lemma 3.2 *Es sei H eine beliebige Gruppe mit einem Normalteiler L und Untergruppen V und W . Es sei S das vollständige Urbild des Normalisators $N_{V/L}(W/L)$. Dann enthält S den Normalisator $N_V(W)$ von W in V .*

Beweis: Es sei ein beliebiges $x \in N_V(W)$ gegeben. Dann gilt $(W/L)^{xL} = W^x L/L = W/L$, also normalisiert xL die Untergruppe W/L . Da aber x im vollständigen Urbild von xL liegt, folgt $x \in S$ und damit die Behauptung. ‡‡

Man beachte, daß in Lemma 3.2 im allgemeinen die Gleichheit nicht gilt. Dazu betrachte man die symmetrische Gruppe S_3 auf drei Punkten. Dann besitzt S_3 eine Untergruppe L der Ordnung 3, welche Normalteiler ist. Weiter sei W eine der drei zyklischen Gruppen der Ordnung 2. L und W erzeugen zusammen die S_3 , also ist der Normalisator $N_{S_3/L}(W/L) = S_3/L$. Andererseits ist V selbstnormalisierend. Es gilt aber

Lemma 3.3 *Es sei H eine beliebige Gruppe mit einem Normalteiler L und einer Untergruppe W , so daß W den Normalteiler L enthält. Dann gilt*

$$N_{H/L}(W/L) = N_H(W)/L.$$

Beweis: Da die Untergruppe W in ihrem Normalisator enthalten ist, enthält er auch den Normalteiler $L \leq W$. Mit Lemma 3.2 folgt sofort $N_H(W)/L \leq N_{H/L}(W/L)$. Sei also ein beliebiges $xL \in N_{H/L}(W/L)$ gegeben. Dann gilt

$$(W/L)^{xL} = W^x/L = W/L,$$

also insbesondere $W^x = W$. Das heißt, es gilt $x \in N_H(W)$ und somit $xL \in N_H(W)/L$.
‡‡

Für das vollständige Urbild S des Normalisators $N_{G/N}(UN/N)$ liefert Lemma 3.2

$$N_G(U) = N_S(U),$$

das heißt, statt den Normalisator in G zu berechnen, ist es möglich, ihn in der im allgemeinen kleineren Gruppe S zu berechnen. Dies hat Auswirkungen auf die auftretenden Orbitlängen und damit auf die abzuspeichernden Untergruppen beziehungsweise konjugierende Elemente. Bei der Berechnung von S tritt ein Orbit der Länge $[G/N : N_{G/N}(UN/N)] = [G : S]$ auf, bei der Berechnung von $N_S(U)$ analog einer der Länge $[S : N_S(U)]$. Insgesamt liefert dies

$$[G : S] + [S : N_S(U)]$$

zu berechnende konjugierte Untergruppen, anstelle der

$$[G : N_G(U)] = [G : S] * [S : N_S(U)]$$

bei der direkten Berechnung. Dieses Verfahren kann natürlich rekursiv bei der Berechnung von S mit $G \leftarrow G/N$ und $N \leftarrow N_{r-1}$ angewendet werden, das heißt, es wird der Reihe nach der Normalisator in $G/N_1 = \{1\}, G/N_2, \dots, G/N_r, G/N_{r+1} = G$ berechnet.

Wir wollen jetzt annehmen, daß der Normalisator $N_{G/N}(UN/N)$ bekannt sei und S sein vollständiges Urbild ist. Es ist—wie oben gesehen—nun $N_S(U)$ zu berechnen. Da jedoch $|N|$ oder mehr zu U konjugierte Untergruppen bei der Berechnung des Normalisators $N_S(U)$ auftreten, ist eine weitere Reduktion der Orbitlängen wünschenswert. Falls UN zum Beispiel das semidirekte Produkt $GF(p^r)^* \ltimes GF(p^r)^+$ der additiven Gruppe $GF(p^r)^+$ des Körpers $GF(p^r)$ mit seiner multiplikativen Gruppe $GF(p^r)^*$ ist, gibt es $|N|$ konjugierte Untergruppen.

Eine weitere Reduktion der Orbitlänge ermöglicht

Lemma 3.4 *Es sei H eine beliebige Gruppe mit Normalteiler L und Untergruppen V und W . Dann gilt*

$$N_V(W \cap L) \geq N_V(W).$$

Beweis: Es sei ein beliebiges $x \in N_V(W)$ gegeben, weiter sei ein beliebiges $w \in W \cap L$ gegeben. Dann gilt einerseits $w^x \in W$, da x im Normalisator von W liegt, und andererseits $w^x \in L$, da w in dem Normalteiler L liegt. Insgesamt gilt $w^x \in W \cap L$; da w beliebig war, folgt $x \in N_G(W \cap L)$. Dies zeigt die Behauptung. ‡‡

Analog obiger Reduktion nach Lemma 3.2 ermöglicht es Lemma 3.4, indem der Normalisator $N_S(U)$ als $N_{N_S(U \cap M)}(U)$ berechnet wird, die auftretenden Orbitlängen auf insgesamt

$$[S : N_S(U \cap M)] + [N_S(U \cap M) : N_S(U)]$$

anstelle der sonst benötigten

$$[S : N_S(U)] = [S : N_S(U \cap M)] * [N_S(U \cap M) : N_S(U)]$$

zu reduzieren.

Dieses Verfahren kann nun natürlich auch wieder rekursiv verwendet werden, indem der Reihe nach die Normalisatoren von

$$U \cap N_{r+1} = \{1\} \leq U \cap N_r \leq \dots \leq U \cap N_2 < U \cap N_1 = U$$

in S berechnet werden.

Wie in [GS92] lassen sich diese beiden Verfahren zusammenfassen, um den Normalisator einer Untergruppe U zu berechnen.

Satz 3.5 *Es sei G eine beliebige Gruppe mit einer Untergruppe U . Weiter sei eine Normalreihe $G = N_1 \triangleright \dots \triangleright N_{r-1} \triangleright N_r = \{1\}$ von G bekannt. Wenn N_{ii} das vollständige Urbild von $N_{G/N_{i-1}}(UN_{i-1}/N_{i-1})$ in G/N_i bezeichnet und man der Reihe nach die Normalisatoren N_{ij} von $T_{ij} = (UN_i \cap N_j)/N_i$ in $N_{i(j+1)}$ für $j = i-1, \dots, 1$ berechnet für $i = 2, \dots, r+1$, so gilt $N_G(U) = N_{(r+1)1}$.*

Beweis: Für $j = 1$ zeigt Lemma 3.2 die Behauptung für G/N_i und UN_i/N_i und für festes i zeigt Lemma 3.4 die Behauptung für beliebige j . ‡‡

Damit ergibt sich folgender Algorithmus.

Algorithmus 3.1. (Verbesserter Normalisator)

- Eingabe :*
- Eine endliche, polyzyklische Gruppe G .
 - Eine Normalreihe $E = [G = N_1, \dots, N_{r+1} = \{1\}]$ mit elementar abelschen Faktoren.
 - Eine Untergruppe U von G .

Ausgabe : Der Normalisator $N_G(U)$ von U in G .

```

Normalizer := function( G, E, U )
  S := G/E[2]; # = N_{G/N_2}(UN_2/N_2)
  for i from 3 to r+1 do
    S := vollständiges Urbild von S ≤ G/E[i-1] in G/E[i];
    for j from i-1 downto 1 do
      S := N_S(UE[i]/E[i] ∩ E[j]/E[i]);
    
```

```

        od;
    od;
    return ( S );
end.

```

Man beachte, da G/N_2 eine abelsche Gruppe ist, gilt $G/N_2 = N_{G/N_2}(W)$ für jede Untergruppe W von G/N_2 .

3.3 Der lineare Fall

Es sei $N = N_r$. Der Normalisator $N_{G/N}(UN/N)$ sei schon gemäß Algorithmus 3.1 berechnet worden. $S \leq G$ sei sein vollständiges Urbild. Es muß als nächster Schritt also der Normalisator $N_S(U \cap N)$ von $U \cap N$ in S berechnet werden.

In diesem Fall ist aber $U \cap N$ als Untergruppe von N ein Teilraum des Vektorraums N auf dem die Gruppe S linear operiert. Das heißt, der im allgemeinen recht aufwendige "collection"-Prozeß und nicht-kommutative Gauß zur Berechnung der zu $U \cap N$ konjugierten Untergruppen kann durch einfachere Matrixmultiplikationen, Vektoradditionen und einen kommutativen Gauß ersetzt werden.

Es sei $\mathcal{B} = (g_{i_r}, \dots, g_r)$ ein kanonisches Erzeugendensystem und damit eine Basis für N . Weiter sei $N \cap U$ durch das kanonische Erzeugendensystem (n_1, \dots, n_u) gegeben und $\alpha : G \rightarrow GL(r - i_r + 1, q_r)$ beschreibe die Operation von G auf dem als Vektorraum aufgefaßten elementar abelschen q_r -Normalteiler N . Dann läßt sich jede unter $g \in G$ zu $U \cap N$ konjugierte Untergruppe eindeutig durch die Matrix identifizieren, welche durch einen vollständigen Gaußalgorithmus aus

$$\begin{pmatrix} m(\mathcal{B}, n_1) * m(\mathcal{B}, \alpha(g), \mathcal{B}) \\ \vdots \\ m(\mathcal{B}, n_u) * m(\mathcal{B}, \alpha(g), \mathcal{B}) \end{pmatrix}$$

hervorgegangen ist.

3.4 Reduktion der Orbitlänge im allgemeinen Fall

Es sei $N = N_r$, $M_1 = N_{i-1}$ und $M_2 = N_i$ für ein festes i mit $i \in \{1, \dots, r-1\}$ und $M_1 \cap U \neq M_2 \cap U$. Der Normalisator $S = N_{G^*}(U \cap M_2)$ von $U \cap M_2$, für G^* als vollständiges Urbild von $N_{G/N}(UN/N)$, sei schon gemäß Algorithmus 3.1 berechnet worden. Es muß als nächster Schritt also der Normalisator $N_S(U \cap M_1)$ von $U \cap M_1$ in S berechnet werden.

Mit Abschnitt 3.1 ist dies direkt möglich, jedoch können die auftretenden Orbitlängen durch Anwendung des Homomorphieprinzips im allgemeinen weiter reduziert werden. Dabei ist zugelassen, daß die Ordnungen von $(U \cap M_1)/(U \cap M_2)$ und N nicht teilerfremd sind,

das heißt, $q_r = q_i$ ist möglich. Für den Fall, daß $q_r \neq q_i$ gilt, kann jedoch—wie der nächste Abschnitt zeigt—auf einen Orbitalgorithmus völlig verzichtet werden.

Die Orbitlängen werden durch Anwenden der Kohomologietheorie aus Kapitel 2 reduziert. Zuerst wollen wir uns jedoch die Beziehungen der auftretenden Untergruppen verdeutlichen.

Nach Konstruktion gilt $U \leq S \leq N_G(UN)$. Aus der Definition von S folgt sofort, daß $U \cap M_2$ ein Normalteiler von S ist. Ein weiterer Normalteiler von S ist durch $N \cap S$ gegeben. Für diese beiden Normalteiler gilt

$$\begin{aligned} (N \cap S) \cap (U \cap M_2) &= N \cap S \cap U \cap M_2 \\ &= (S \cap U) \cap (N \cap M_2) \\ &= U \cap N, \end{aligned}$$

siehe Figur Seite 42. Es sei $N^* := (N \cap S) \cdot (U \cap M_2)$. Dann ist $N^*/(U \cap M_2) \cong (N \cap S)/(N \cap U)$, also eine elementar abelsche q_r -Gruppe. Es gilt weiterhin

Lemma 3.6 *Es seien G, N, M_1, M_2 und U wie oben angegeben. Dann schneiden sich die Untergruppen $N^* = (N \cap S) \cdot (U \cap M_2)$ und $(U \cap M_1)$ in der Untergruppe $U \cap M_2$.*

Beweis:

“ \supseteq ”: Nach Definition von $N^* = (N \cap S) \cdot (U \cap M_2)$ liegt die Untergruppe $U \cap M_2$ in N^* . Andererseits ist M_1 nach Voraussetzung eine Obermenge von M_2 , also liegt $U \cap M_2$ auch im Schnitt von U und M_1 . Damit enthält der Schnitt von N^* und $U \cap M_1$ die Untergruppe $U \cap M_2$.

“ \subseteq ”: Es sei ein beliebiges $x \in N^* \cap (U \cap M_1)$ gegeben. Dann existieren Elemente $n \in (N \cap S)$, $m_2 \in (U \cap M_2) \leq U$ und $m_1 \in U \cap M_1 \leq U$ mit $x = nm_2 = m_1$. Also gilt $n = m_1/m_2 \in U$ und damit $n \in U \cap (N \cap S) = U \cap N \leq U \cap M_2$. Daher folgt $x = nm_2 \in U \cap M_2$. Die Untergruppe $U \cap M_2$ enthält damit den Schnitt von N^* und $U \cap M_1$.

Insgesamt gilt also die Gleichheit. ‡‡

Es sei eine weitere Untergruppe U^* von S gegeben durch $U^* := (U \cap M_1) \cdot N^*$. Für diese Untergruppe gilt dann sogar

Lemma 3.7 *Es sei G, N, M_1, M_2, U und S wie oben angegeben. Dann ist die Untergruppe $U^* = (U \cap M_1) \cdot N^*$ von S ein Normalteiler von S .*

Beweis: Man beachte, daß nach Algorithmus 3.1 schon $N_G(UN) \geq S$ gilt. Nach Definition von N^* ist $U^* = (U \cap M_1) \cdot (N \cap S) \cdot (U \cap M_2)$. Da aber die Untergruppe $(U \cap M_2)$ schon ein Normalteiler von S ist, gilt also

$$U^* = (U \cap M_1) \cdot (N \cap S) \cdot (U \cap M_2)$$

$$\begin{aligned}
&= (U \cap M_1) \cdot (U \cap M_2) \cdot (N \cap S) \\
&= (U \cap M_1) \cdot (N \cap S).
\end{aligned}$$

Wir müssen also zeigen, daß das Produkt $(U \cap M_1) \cdot (N \cap S)$ normal in S ist.

Dazu sei jetzt ein beliebiges $x \in (U \cap M_1) \cdot (N \cap S) \leq S$ und ein $s \in S$ gegeben. Dann läßt sich x in einen $U \cap M_1$ Anteil u und einen $N \cap S$ Anteil n aufspalten. Da s nach obiger Bemerkung schon UN normalisiert, liegt also insbesondere u^s in UN . Daher existieren Elemente $u' \in U$ und $n' \in N$ mit $u^s = u'n'$. Insgesamt gilt mit dieser Setzung

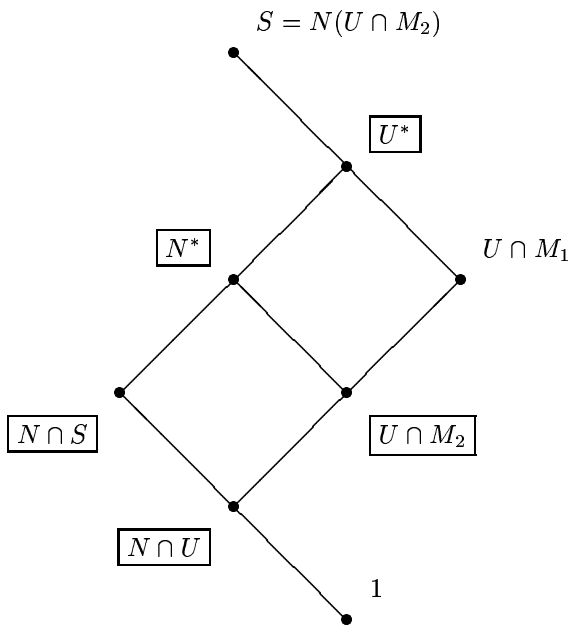
$$\begin{aligned}
x^s &= (u \cdot n)^s \\
&= u^s \cdot n^s \\
&= u' \cdot n'n^s.
\end{aligned}$$

Andererseits liegt aber nach Konstruktion sowohl $u \in U \cap M_1$ als auch $n' \in N < M_1$ in dem Normalteiler M_1 . Da M_1 ein Normalteiler ist, gilt also auch $u^s \in M_1$. Damit folgt aber $u' = u^s * n'^{-1} \in M_1$, und somit liegt u' in $U \cap M_1$. Da n in dem Normalteiler N und der Untergruppe S von G liegt, folgt auch hier $n^s \in (N \cap S)$. Da jetzt also x^s , u' und n^s in der Untergruppe S liegen, muß auch $n' = u'^{-1}x(n^s)^{-1} \in S$ gelten. Insgesamt zeigt dies aber $x^s \in (U \cap M_1) \cdot (N \cap S)$. $\ddagger\ddagger$

Anlog zu obiger Beweisführung in Lemma 3.7 gilt

Lemma 3.8 *Es sei eine beliebige Gruppe H mit Normalteiler L und einer Untergruppe U gegeben. Weiter sei S eine Untergruppe von H , so daß $N_H(UL) \geq S \geq U$ gilt. Dann folgt aus $L \cap S = L \cap U$, daß U ein Normalteiler von S ist.*

Es seien also die Untergruppen N^* und U^* von S definiert durch $N^* = (N \cap S) \cdot (U \cap M_2)$ und $U^* = (U \cap M_1) \cdot N^*$. Dann liegt nach obigen Überlegungen nun folgende Situation vor. Dabei sind die nichttrivialen Normalteiler von S umrandet. $N \cap U$ ist nach Algorithmus 3.1 ein Normalteiler von S .



Falls also die Faktorgruppe $(N \cap S)/(N \cap U) \cong N^*/(U \cap M_2)$ trivial ist, das heißt, $N \cap S = N \cap U$ gilt, folgt mit Lemma 3.8, daß $U \cap M_1$ ein Normalteiler von S ist. Damit ist der Normalisator von $U \cap M_1$ in S gegeben durch S . Da aber die Bedingung $N \cap S = N \cap U$ nur von S und nicht von M_1, M_2 abhängt, folgt sofort $S = N_S(U)$. Damit gilt aber $S = N_G(U)$ nach Lemma 3.4 und der Algorithmus 3.1 kann an dieser Stelle schon abgebrochen werden.

Es sei also $N \cap S \neq N \cap U$ vorausgesetzt. Damit ist dann insbesondere $N^*/(U \cap M_2)$ ein nichttrivialer elementar abelscher Normalteiler von $S/(U \cap M_2)$. Da $U \cap M_2 \subset U \cap M_1$ gilt und $U \cap M_2$ ein Normalteiler von S ist, folgt mit Lemma 3.3

$$N_S(U \cap M_1)/(U \cap M_2) = N_{S/(U \cap M_2)}((U \cap M_1)/(U \cap M_2)).$$

Wir können also ohne Beschränkung der Allgemeinheit zur Vereinfachung der Schreibweise $U \cap M_2 = \{1\}$ annehmen.

Es sei jetzt $\mathcal{K} = \{K < U^* \mid KN^* = U^*, K \cap N^* = \{1\}\}$. Dann operiert S via Konjugation auf \mathcal{K} , denn für ein beliebiges Komplement K zu N^* in U^* und ein $s \in S$ gilt $K^s < (U^*)^s = U^*$ und $K^s N^* = (KN^*)^s = (U^*)^s = U^*$ und somit $K^s \in \mathcal{K}$. Da U^* ein Normalteiler von S ist, ist die U^* -Bahnzerlegung von \mathcal{K} eine Blockzerlegung von \mathcal{K} für

die Operation von S auf \mathcal{K} . Durch Berechnung des Blockstabilisators $N_S((U \cap M_1)^{U^*})$ läßt sich die Orbitlänge weiter reduzieren. Diesen Weg wollen wir nun einschlagen.

Der Operationsbereich \mathcal{K} , nämlich die Menge aller Komplemente in U^* zu N^* , entspricht nach Kapitel 2 den 1-Kozykeln aus $Z^1(U \cap M_1, N^*)$, die verschiedenen Blöcke entsprechen den Elementen der 1-Kohomologiegruppe $H^1(U \cap M_1, N^*)$. Der Stabilisator von $U \cap M_1$ unter Konjugation mit Elementen aus S läßt sich nach dem Homomorphieprinzip aus dem Blockstabilisator $N_S((U \cap M_1)^{U^*})$ und dem Stabilisator $N_{U^*}(U \cap M_1)$ von $U \cap M_1$ unter Konjugation mit Elementen aus U^* zusammensetzen. Daher zuerst

Lemma 3.9 *Es sei H eine beliebige Gruppe mit Normalteiler L und einem Komplement K zu L . Dann gilt*

$$\begin{aligned} N_H(K) &= K \cdot N_L(K) \\ &= K \cdot C_L(K). \end{aligned}$$

Beweis: Einerseits gilt $K \cdot N_L(K) \subseteq N_H(K)$, andererseits läßt sich jedes Element $h \in N_H(K) \subseteq H = KL$ zerlegen in einen K -Anteil $k \in K$ und L -Anteil l mit $h = kl$. Dann gilt aber $K^h = K^{kl} = K^l = K$ und damit $h \in K \cdot N_L(K)$.

Es bleibt also $N_L(K) = C_L(K)$ zu zeigen. Es sei ein beliebiges $k \in K$ und $n \in N_L(K)$ gegeben. Dann liegt $(n^{-1})^k$ in L und k^n in K . Also

$$k^{-1}n^{-1}kn = (n^{-1})^k n = k^{-1}k^n \in K \cap L = \{1\},$$

das heißt n zentralisiert k . Da k beliebig war, gilt also $n \in C_L(K)$, da $n \in N_L(K)$ beliebig war, auch $N_L(K) \subseteq C_L(K)$. Dies zeigt insgesamt die Aussage. $\ddagger\ddagger$

Den Zentralisator $C_{N^*}(U \cap M_1)$ erhalten wir aber schon bei der Berechnung der 1-Kohomologiegruppe mit Algorithmus 2.4.

Um den Normalisator nach Lemma 1.17 rekonstruieren zu können, müssen wir in der Lage sein, zu zwei Elementen eines Blockes ein konjugierendes Element zu finden. Mit Lemma 2.18 läßt sich ein solches Element als Lösung einer inhomogenen Gleichung berechnen.

Insgesamt läßt sich der Normalisator $N_S(U \cap M_1)$ mit folgendem Algorithmus berechnen.

Algorithmus 3.2. (Nächster Normalisatorschritt (allgemein))

- Eingabe :*
- Zwei CGS der Schnitte $U \cap M_1$ und $U \cap M_2$, dabei gelte $U \cap M_1 \neq U \cap M_2$.
 - Ein CGS des Normalisators S von $U \cap M_2$ im vollständigen Urbild von $N_{G/N}(UN/N)$ und des Schnitts $N \cap S$, so daß $N \cap S \neq N \cap U$ gilt.

Ausgabe : Ein CGS des Normalisators $N_S(U \cap M_1)$.

```

NextStep := function( S, N ∩ S, U ∩ M1, U ∩ M2 )
  N* := (N ∩ S) · (U ∩ M2);
  U* := (U ∩ M1) · N*;
  rels := Potenz-Kommutator Relationen von (U ∩ M1)/(U ∩ M2);
  tmp := OneCocycles( U*/(U ∩ M2), N*/(U ∩ M2), rels );
  B1 := tmp.B1; # Gruppe der 1-Koränder
  CN := tmp.CN; # Zentralisator CN*/(U ∩ M2)((U ∩ M1)/(U ∩ M2))
  if B1 = {0} then
    Stab := Stabilisator von U ∩ M1 in S/U*;
    NN := Vollständiges Urbild von Stab ≤ S/U* in S;
  else
    BS := Stabilisator von γ(U ∩ M1)/(U ∩ M2) + B1 in S/U*;
    for i from 1 to Length(BlockStab) do
      BS[i] := Ein Repräsentant von BS[i] ∈ S/U*;
    od;
    Stab := [];
    for i from 1 to Length(BS) do
      n := ConjugatingElement(
        U*/(U ∩ M2),
        N*/(U ∩ M2),
        (U ∩ M1)/(U ∩ M2),
        ((U ∩ M1)/(U ∩ M2))BS[i]);
      Stab[i] := BlockStab[i] * n-1;
    od;
    Stab := Closure( Stab, (U ∩ M1)/(U ∩ M2), CN );
    NN := Vollständiges Urbild von Stab ≤ S/(U ∩ M2) in S;
  fi;
  return NN;
end.

```

Die Stabilisatoren seien dabei als Stabilisatoren unter Konjugation beziehungsweise der entsprechenden Operation auf der 1-Kohomologiegruppe zu verstehen. Ist die 1-Kohomologiegruppe trivial, gibt es nur einen Block, auf dem S/U^* operieren kann, und damit ist der Stabilisator dieses Block gleich ganz S/U^* .

Man beachte, daß man für den Algorithmus nicht explizit die Gruppe der 1-Kozykeln benötigt.

3.5 Der teilerfremde Fall

Die Bezeichnungen seien wie im vorhergehenden Abschnitt. Zusätzlich gelte $q_r \neq q_i$, das heißt, $N^*/(U \cap M_2)$ ist ein Hallscher Normalteiler von $U^*/(U \cap M_2)$. Damit sind aber nach

dem Satz von Schur-Zassenhaus, siehe [Hup67] I.§18, alle Komplemente zu $N^*/(U \cap M_2)$ konjugiert. Wir können in Algorithmus 3.2 also von vornherein annehmen, daß die 1-Kohomologiegruppe trivial ist und die Berechnung entsprechend fortsetzen.

Der Vektorraum $N^*/(U \cap M_2)$ habe die Dimension m und das Komplement $(U \cap M_1)/(U \cap M_2)$ habe das kanonische Erzeugendensystem k_1, \dots, k_l . Dann läßt sich ein Element des Normalteilers, welches ein Komplement in ein anderes konjugiert, nach Lemma 2.15 durch Lösen eines inhomogenen linearen Gleichungssystems mit m Unbekannten und $m * l$ Gleichungen bestimmen.

Eine andere Möglichkeit, ein solches Element zu finden, liefert folgender Satz von S. P. Glasby und M. C. Slattery in [GS92], welcher auf einen Satz von Kantor und Taylor ([KT88]) zurückgeht.

Satz 3.10 *Es sei G eine endliche polyzyklische Gruppe mit einer Potenz-Kommutator Präsentation in den Erzeugern (g_1, \dots, g_t) . Weiter seien K und H zwei Untergruppen, welche eine gemeinsame Untergruppe L enthalten, so daß der Index $[K : L] = [H : L] = p$ eine Primzahl ist und L normal in H und K ist. Dann gibt es Elemente $k \in K$ und $h \in H$ mit $K = \langle L, k \rangle$ und $H = \langle L, h \rangle$. Für diese Elemente sei ein elementar abelscher q -Normalteiler N bekannt, so daß $hN = kN$ und $q \neq p$ gilt. Dann gilt für*

$$x = (m^h \cdot (m^2)^{(h^2)} \dots (m^{p-1})^{(h^{p-1})})^t$$

mit $m = h^{-1}k$ und t eine ganze Zahl, so daß $-pt \equiv 1 \pmod p$ gilt, daß

$$H^x = K.$$

Beweis: Siehe [Gla87] 48-50. ‡‡

Dieser Satz ermöglicht

Algorithmus 3.3. (Konjugierendes Element (teilerfremd))

- Eingabe :*
- Eine endliche, polyzyklische Gruppe G mit AG-System $\langle g_1, \dots, g_t \rangle$ und einem elementar abelschen q -Normalteiler N mit CGS $\langle g_1, \dots, g_t \rangle$.
 - Untergruppen W und V gleicher Ordnung von G , so daß $WN = VN$ gilt.
 - Einen Normalteiler K von G , welcher in $W \cap V$ enthalten ist, so daß W/K und damit V/K p -Gruppen für eine Primzahl $p \neq q$.

Ausgabe : Ein Element $n \in N$ mit $W^n = V$.

```
ConjugatingElementCoprime := function( N, W, V, K )
  w := Repräsentanten eines kanonischen Erzeugendensystems von W/K;
  v := [ ];
```

```

for i from 1 to Length( w ) do
  v[i] := Ein Element v aus V mit vKN = w[i]KN;
od;
x := 1_N;
o := Eine ganze Zahl mit -op ≡ 1 modulo q;
for i from Length( w ) downto 1 do
  tmp := (w[i]^x)^{-1} * v[i];
  y := Ein Element y ∈ V mit tmp ∈ yN;
  z := y^{-1} * tmp;
  x := x * (∏_{j=1}^{p-1} (z^j)^{v[i]^j})^t;
od;
return ( x );
end.

```

Mit $N \leftarrow (N \cap S)$, $K \leftarrow (U \cap M_2)$, $W \leftarrow (U \cap M_1)$ und V der zu W konjugierten Untergruppe berechnet dieser Algorithmus dann das gesuchte Element. Man beachte, daß die Komplexität dieses Algorithmus im wesentlichen von der Länge l des kanonischen Erzeugendensystem für $(U \cap M_1)/(U \cap M_2)$ und der Primzahl p abhängt und nicht wie der Algorithmus nach Lemma 2.15 von l und der Dimension des Normalteilers N .

Insgesamt läßt sich der Normalisator von $U \cap M_1$ wie folgt berechnen.

Algorithmus 3.4. (Nächster Normalisator (teilerfremd))

Eingabe : • Zwei CGS der Schnitte $U \cap M_1$ und $U \cap M_2$, dabei gelte $U \cap M_1 \neq U \cap M_2$.

- Ein CGS des Normalisators S von $U \cap M_2$ im vollständigen Urbild von $N_{G/N}(UN/N)$ und des Schnitts $N \cap S$, so daß $N \cap S \neq N \cap U$ gilt und N und $(U \cap M_1)/(U \cap M_2)$ teilerfremde Ordnungen haben.

Ausgabe : Ein CGS des Normalisators $N_S(U \cap M_1)$.

```

NextStepCoprime := function( S, N ∩ S, U ∩ M_1, U ∩ M_2 )
  N* := (N ∩ S) · (U ∩ M_2);
  U* := (U ∩ M_1) · N*;
  CN := OneCoboundaries( U*/(U ∩ M_2), N*/(U ∩ M_2) ).CN;
  Stab := Repräsentanten für S/U*;
  for i from 1 to Length( Stab ) do
    n := ConjugatingElementCoprime(
      U*/(U ∩ M_2),
      N*/(U ∩ M_2),
      (U ∩ M_1)/(U ∩ M_2),
      ((U ∩ M_1)/(U ∩ M_2))^{Stab[i]};
  end;
end;

```

```

     $Stab[i] := Stab[i] * n^{-1};$ 
  od;
   $Stab := \text{Closure}( Stab, (U \cap M_1)/(U \cap M_2), CN );$ 
   $NN := \text{Vollständiges Urbild von } Stab \leq S/(U \cap M_2) \text{ in } S;$ 
  return  $NN;$ 
end.

```

3.6 Endgültige Fassung

Bei der Implementation des Algorithmus ist zu beachten, daß aus $U \leq N_i$ für ein $i \in \{1, \dots, r+1\}$ sofort $(UN_i \cap N_j) = N_i \triangleleft G$ für alle $j \in \{1, \dots, i\}$ folgt. Gilt andererseits aber $N_{i-1} \leq HN_i$, das heißt, H deckt N_{i-1}/N_i , so folgt $HN_i = HN_{i-1}$. Mit Lemma 3.3 ist dann $N_{G/N_i}(HN_i/N_i)$ gleich dem vollständigen Urbild von $N_{G/N_{i-1}}(HN_i/N_{i-1})$.

Zusammengefaßt ergibt sich für den Normalisator folgender Algorithmus.

Algorithmus 3.5. (Normalisator)

- Eingabe* :
- Eine endliche, polyzyklische Gruppe G .
 - Eine Normalreihe $E = [G = N_1, \dots, N_{r+1} = \{1\}]$ mit elementar abelschen Faktoren.
 - Eine Untergruppe U von G .

Ausgabe : Der Normalisator $N_G(U)$ von U in G .

```

Normalizer := function( G, E, U )
  S := G/E[2];
  for i from 3 to r+1 do
    S := vollständiges Urbild von  $S \leq G/E[i-1]$  in  $G/E[i]$ ;
    if  $E[i-1] \not\subseteq U \cdot E[i]$  and  $U \not\subseteq E[i]$  then
      S := Stabilisator von  $(U \cdot E[i] \cap E[i-1])/E[i]$  unter Verwendung linearer Operationen;
      j := i-2;
      k := i-1;
      while j > 0 and  $U \cdot E[i] \cap E[k] = U \cdot E[i] \cap E[j]$  do
        j := j-1;
      od;
      while j > 0 and  $E[i-1]/E[i] \cap S \neq (E[i-1] \cap U \cdot E[i])/E[i]$  do
        if  $q_{i-1} \neq q_j$  then
          S := NextStepCoprime(S,  $E[i-1] \cap S, U \cap E[j], U \cap E[k]$ );
        else
          S := NextStep(S,  $E[i-1] \cap S, U \cap E[j], U \cap E[k]$ );
        fi;
      od;
    fi;
  od;
end;

```


$$M_B := \alpha(B) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \text{ und}$$

$$M_C := \alpha(C) = 2 \cdot I_{GL(11,5)},$$

welche im wesentlichen das Kranzprodukt von Q vom Grad 11 mit einem 1-dimensionalen Z_5 -Vektorraum beschreibt. So eine Darstellung existiert, da die Matrizen M_A und M_B gerade die Permutationsmatrizen von A und B sind, und M_C beide zentralisiert.

Es sei G das semidirekte Produkt von H mit dem Normalteiler $N = \mathcal{V}_{1 \times 11}(Z_5)$, wobei die Operation von H auf N durch obige Matrizen beschrieben wird. Diese Gruppe ist, da Q auflösbar ist, ebenfalls auflösbar. Es sei $\{v_1, \dots, v_{11}\}$ die Standardbasis von N , dann besitzt G bezüglich der Erzeuger $C_1 = C, C_2 = C^2, B, A, v_1, \dots, v_{11}$ eine Potenz-Kommutator Präsentation, denn für $i \in \{1, \dots, 11\}$ gilt

$$\begin{aligned} C_1^2 &= C_2 \\ C_2^2 &= 1 \\ B^5 &= 1 \\ A^{11} &= 1 \\ v_i^5 &= 1 \\ [A, B] &= A^3 \\ [v_i, C] &= v_i^{-1} * v_i^{M_C} \\ [v_i, B] &= v_i^{-1} * v_i^{M_B} \\ [v_i, A] &= v_i^{-1} * v_i^{M_A} \end{aligned}$$

und alle anderen Kommutatoren sind trivial.

Eine Normalreihe mit elementar abelschen Faktoren ist durch

$$\begin{aligned} N_1 &= \langle C_1, C_2, B, A, v_1, \dots, v_{11} \rangle \triangleright \\ N_2 &= \langle B, A, v_1, \dots, v_{11} \rangle \triangleright \\ N_3 &= \langle A, v_1, \dots, v_{11} \rangle \triangleright \end{aligned}$$

$$\begin{aligned} N_4 &= \langle v_1, \dots, v_{11} \rangle \triangleright \\ N_5 &= \{1\} \end{aligned}$$

gegeben.

Will man nun den Normalisator der Untergruppe $U = \langle B * v_1 \rangle$ berechnen, so sieht man, daß $N_{G/N_4}(UN_4/N_4) = \langle C_1, C_2, B \rangle N_4/N_4$ gilt, denn C_1 und C_2 zentralisieren B , während aber $B^A = B * A^{-3}$ nicht in UN_4 liegt.

Es sei also nun $S = \langle C_1, C_2, B, N_4 \rangle$ das vollständige Urbild des schon berechneten Normalisators $N_{G/N_4}(UN_4/N_4)$. Als nächstes soll nun $N_S(U)$ berechnet werden. Nach Algorithmus 3.1 müssen nun die Normalisatoren von $U \cap N_4$, $U \cap N_3$, $U \cap N_2$ und $U \cap N_1$ in S berechnet werden. Da $U \cap N_4$ und $U \cap N_3$ trivial sind, beginnen wir mit $U \cap N_2 = U$. U und N_4 sind beides 5-Gruppen, das heißt, wir befinden uns im allgemeinen Fall von Algorithmus 3.5.

Da

$$I - M_B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{pmatrix}$$

Rang 8 hat, gibt es nach Lemma 2.14 insgesamt $5^8 \approx 4 * 10^5$ unter N_4 zu U konjugierte Untergruppen. Insgesamt würde S bei Konjugation einen Orbit der Länge $\approx 16 * 10^5$ erzeugen. Wie man mit Algorithmus 2.4 nachrechnet, sind $\langle B \rangle$, $\langle B * v_1 \rangle$, $\langle B * v_1^2 \rangle$, $\langle B * v_1^3 \rangle$, und $\langle B * v_1^4 \rangle$ Repräsentanten für die Konjugiertenklassen von Komplementen zu N_4 in $\langle N_4, B \rangle$. Die 1-Kohomologiegruppe ist also von der Dimension 1 und S erzeugt auf ihr einen Orbit der Länge 4. Insgesamt gesehen muß also bei der Verwendung der Kohomologietheorie maximal ein Orbit der Länge 4 berechnet werden, während bei einer direkten Berechnung mehr als eine Million konjugierter Untergruppen zu berechnen wären.

Literatur

Die grundlegende Idee der rekursiven Berechnung des Normalisators findet sich in [GS92], der hier erbrachte Beweis sowie die Verwendung der Kohomologietheorie ist neu. Eine

Zusammenfassung kann auch in [CNW90] gefunden werden, dort sind jedoch leicht andere Beweise für die Lemmata in Abschnitt 3.4 gegeben.

Kapitel 4

Konjugiertenklassen von Komplementen

In diesem Kapitel wollen wir die Konjugiertenklassen von Komplementen eines Normalteilers in einer endlichen, polyzyklischen Gruppe untersuchen. Dies stellt somit eine Fortsetzung von Kapitel 2 dar, in dem der Normalteiler elementar abelsch sein mußte. Auf diese Forderung werden wir nun verzichten. Das hier beschriebene Verfahren ist eine Verallgemeinerung eines Algorithmus zur Berechnung der Konjugiertenklassen von Elementen einer auflösbaren Gruppe—siehe [MN89].

Es bezeichne G in diesem Kapitel eine endliche, polyzyklische Gruppe mit Potenz-Kommutator Präsentation in den Erzeugenden g_1, \dots, g_t . Es sei weiter eine Normalreihe

$$G = N_1 \triangleright N_2 \triangleright \dots \triangleright N_r \triangleright N_{r+1} = \{1\}$$

mit elementar abelschen Faktoren bekannt. Dabei seien die Faktoren N_i/N_{i+1} für $1 \leq i \leq r$ elementar abelsche q_i -Gruppen für Primzahlen q_i . Diese Reihe sollte zur Beschleunigung des Algorithmus, insbesondere der darin auftretenden Faktorgruppenbildung, von der Kompositionsreihe

$$G = \langle g_1, \dots, g_t \rangle \triangleright \langle g_2, \dots, g_t \rangle \triangleright \dots \triangleright \langle g_t \rangle \triangleright \langle 1 \rangle$$

verfeinert werden. Dies ist für den Algorithmus jedoch nicht notwendig.

4.1 Das Homomorphieprinzip

Es sei H eine beliebige Gruppe mit einem Normalteiler M , zu welchem die Konjugiertenklassen von Komplementen berechnet werden sollen. Es sei ein weiterer Normalteiler N von H gegeben, welcher in M liegt.

Um die Konjugiertenklassen von Komplementen zu M zu berechnen, müßten wir eine H -Bahnzerlegung der Menge der Komplemente zu M durchführen. Nach Lemma 2.8 sind jedoch zwei Komplementen genau dann unter H konjugiert, wenn sie auch schon unter M konjugiert sind. Damit gilt

Satz 4.1 *Es sei H eine beliebige Gruppe mit Normalteiler M . \mathcal{K} die Menge aller Komplemente zu M in H und H operiere auf \mathcal{K} per Konjugation. Dann gilt für jedes Komplement $K \in \mathcal{K}$, daß $K^H = K^M$.*

Nach Satz 4.1 reicht eine M -Bahnzerlegung der Menge \mathcal{K} der Komplemente zu M in H zur Bestimmung der Konjugiertenklassen von Komplementen aus.

Einerseits operiert M per Konjugation auf \mathcal{K} . Andererseits operiert M auch auf der Menge $\overline{\mathcal{K}}$ der Komplemente zu M/N in H/N und die Zuordnung $\alpha = (K \in \mathcal{K} \mapsto KN/N) : \mathcal{K} \rightarrow \overline{\mathcal{K}}$ ist mit den Operationen von M auf beiden Mengen verträglich. Wir können also das Homomorphieprinzip aus Lemma 1.17 anwenden. Dies lautet angepaßt auf obige Situation

Lemma 4.2 *Es sei H eine Gruppe mit zwei Normalteiler M und N , so daß $M > N$ gilt. Weiter seien $\mathcal{K} = \{K < H \mid KM = H, K \cap M = \{1\}\}$ und $\overline{\mathcal{K}} = \{\overline{K} < H/N \mid \overline{K} \cdot (M/N) = H/N, \overline{K} \cap (M/N) = N/N\}$ gegeben. $m \in M$ operiert per Konjugation auf \mathcal{K} und per Konjugation mit mN auf $\overline{\mathcal{K}}$. Dann gilt für die Abbildung $\alpha = (K \in \mathcal{K} \mapsto KN/N) : \mathcal{K} \rightarrow \overline{\mathcal{K}}$, daß $(K^m)^\alpha = (K^\alpha)^{mN}$ für alle $K \in \mathcal{K}$ und $m \in M$.*

Es sei Γ die Menge der Repräsentanten für die M -Bahnen von \mathcal{K} und Λ die Menge der Repräsentanten für die M -Bahnen von $\overline{\mathcal{K}}$. Dann gilt

$$\Gamma = \bigcup_{\overline{K} \in \Lambda} \gamma(\overline{K}),$$

wobei $\gamma(\overline{K})$ ein Repräsentantensystem für die S -Bahnen von $\{K \in \mathcal{K} \mid K^\alpha = \overline{K}\}$ des vollständigen Urbildes S von $C_{M/N}(\overline{K})$ in M ist.

Beweis: Die Abbildung α ist wohldefiniert. Denn sei K ein Komplement zu M , dann folgt sofort $(KN/N)(M/N) = (KNM/N) = H/N$ und $(KN/N) \cap M/N = (KN \cap M)/N$. Einerseits enthält $KN \cap M$ den Normalteiler N , andererseits folgt für ein beliebiges $x \in KN \cap M$, daß sich $x = kn$ zerlegen läßt in einen K -Anteil k und einen N -Anteil n , weiter folgt damit aber $k = x/n \in M \cap K = \{1\}$. Also ist KN/N ein Komplement zu M/N in H/N .

Alles was noch zu beweisen ist, daß der Stabilisator von \overline{K} für ein $\overline{K} \in \overline{\mathcal{K}}$ dem vollständigen Urbild von $C_{M/N}(\overline{K})$ in H entspricht. Es sei also ein g aus dem Stabilisator gegeben. Dann gilt $\overline{K}^{gN} = \overline{K}$ und somit liegt gN im Normalisator $N_{H/N}(\overline{K})$. Andererseits ist der Stabilisator eine Untergruppe von M und somit liegt gN im Schnitt $N_{H/N}(\overline{K}) \cap M/N$. Nach Lemma 3.9 entspricht dieser Schnitt dem Zentralisator $C_{M/N}(\overline{K})$. Also liegt g in seinem vollständigen Urbild. $\ddagger\ddagger$

Lemma 4.2 erlaubt nun rekursiv ein Repräsentantensystem für die Konjugiertenklassen von Komplementen zu M zu berechnen. Wir können also per Induktion annehmen, daß für einen Normalteiler N von H mit $M > N$

1. ein Repräsentantensystem $\{U_1/N, \dots, U_k/N\}$ der Konjugiertenklassen von Komplementen zu M/N in H/N sowie
2. für jeden Repräsentanten U_i/N der Zentralisator $C_{M/N}(U_i/N)$

bekannt sei.

Es sei $U = U_i$ und S das vollständige Urbild von $C_{M/N}(U_i/N)$ in H für ein festes $i \in \{1, \dots, k\}$. Nach Lemma 4.2 müssen wir nun das Urbild von U/N in \mathcal{K} bestimmen und dann seine S -Bahnzerlegung. Dies geschieht in den folgenden Abschnitten.

4.2 Herunterziehen von Komplementen

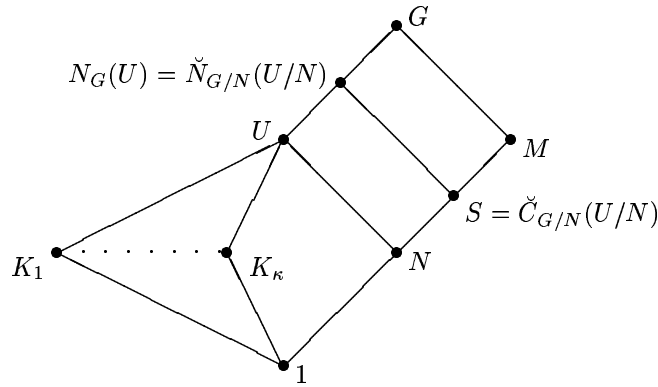
Wir wollen uns nun auf den Fall einer endlichen, polyzyklischen Gruppe G und dem Normalteiler $N = N_r$ einschränken. Da N nun ein elementar abelscher Normalteiler ist, können wir die Ergebnisse aus Kapitel 2 verwenden. Allgemein gilt aber

Lemma 4.3 *Es H eine beliebige Gruppe, $M > L$ zwei Normalteiler von H und \overline{K}/L ein Komplement von M/L . Falls ein Komplement K zu M in H existiert mit $KL/L = \overline{K}/L$, so ist K auch ein Komplement zu L in \overline{K} . Umgekehrt ist jedes Komplement K zu L in \overline{K} ein Komplement zu M in H .*

Beweis: Es sei K ein Komplement zu M in H mit $KL/L = \overline{K}/L$. Dann folgt sofort $K \cap L \leq K \cap M = \{1\}$. Weiter folgt nach Voraussetzung $\overline{K} = KL$, also ist K ein Komplement zu L in \overline{K} .

Es sei jetzt K ein Komplement zu L in \overline{K} . Einerseits gilt dann $K \cap M \leq K \cap (\overline{K} \cap M) = K \cap L = \{1\}$. Andererseits ist $K \cdot M = K \cdot L \cdot M = \overline{K} \cdot M = H$. Somit ist K ein Komplement zu M in H . ‡‡

Mit $H \leftarrow G$, $\overline{K} \leftarrow U$ und $L \leftarrow N = N_r$ entspricht das Urbild von $(U/N)^{\alpha^{-1}}$ nach Lemma 4.3 also der Menge aller Komplemente zu N in U . Diese lassen sich mit Algorithmus 2.4 als 1-Kozykel berechnen beziehungsweise es läßt sich entscheiden, ob keine Komplemente zu N in U existieren. Insgesamt liegt also folgende Situation vor. Dabei bezeichne $\check{C}_{G/N}(U/N)$ und $\check{N}_{G/N}(U/N)$ das vollständige Urbild in G von $C_{G/N}(U/N)$ beziehungsweise $N_{G/N}(U/N)$.



4.3 S -Bahnen

Sollte U keine zerfallende Erweiterung von U/N mit N sein, so ist die Menge der Komplemente leer und somit können wir das nächste U_i betrachten. Nach Satz 2.10 beziehungsweise mit Algorithmus 2.4 läßt sich dies durch Lösen eines linearen Gleichungssystems entscheiden. Wir wollen also bis zum Ende des Abschnitts annehmen, daß Komplemente zu N in U existieren.

S operiert dann nach Lemma 3.2 und 3.7 auf der Menge Θ der Komplemente zu N in U per Konjugation. Andererseits operiert auch N auf Θ per Konjugation. Da N ein Normalteiler von S ist, bildet die N -Bahnzerlegung von Θ eine Blockzerlegung für die Operation von S . Die N -Bahnen von Θ sind nach Lemma 2.8 die Konjugiertenklassen von Komplementen zu N in U , und damit nach Kapitel 2 durch die 1-Kohomologiegruppe $H^1(U/N, N)$ bestimmt. S beziehungsweise S/N operiert damit in natürlicher Weise auf der 1-Kohomologiegruppe $H^1(U/N, N)$ und die S -Bahnzerlegung von Θ unter Operation von S läßt sich aus der S -Bahnzerlegung von $H^1(U/N, N)$ gewinnen.

Insgesamt ergibt sich somit nach Lemma 4.2 folgender Algorithmus zur Bestimmung der Konjugiertenklassen von Komplemente zu einem gegebenen Normalteiler. Man beachte, daß sich der Zentralisator $C_M(K)$ eines Repräsentanten K einer Konjugiertenklasse nach dem Homomorphieprinzip analog Algorithmus 3.2 aus dem Blockstabilisator von $\gamma_K \in H^1(U/N, N)$ und dem Zentralisator $C_N(K)$ zusammensetzen läßt. Der Zentralisator $C_N(K)$ ergibt sich jedoch schon mit Lemma 2.14 bei der Berechnung der 1-Koränder.

Algorithmus 4.1. (Konjugiertenklassen von Komplementen)

- Eingabe* :
- Eine endliche, polyzyklische Gruppe G .
 - Eine Normalreihe $E = [M = N_i, \dots, N_{r+1} = \{1\}]$ mit elementar abelschen Faktoren.
 - Das System Rel_s der Potenz-Kommutator Relationen für G/M .

Ausgabe : Eine Menge von Paaren $(K, C = C_M(K))$, so daß die Menge der K ein Repräsentantensystem für die Konjugiertenklassen von Komplementen zu M in G bilden oder `false`, falls keine Komplemente existieren.

```

Complementclasses := function( G, E, Rel_s )
  if G = E[1] then
    return ( {1}, G );
  fi;
  # Bestimme zuerst Komplemente in der Faktorgruppe G/Nr.
  Nr := E[ Length(E) - 1 ];
  FE := [ E[1]/Nr, ..., E[Length(E) - 1]/Nr ];
  FacReps := Complementclasses(G/Nr, FE, Rel_s);
  if FacReps = false then
    return false ;
  fi;
  # Ziehe Komplemente herunter und bestimme S-Bahnen.
  Reps := [];
  for j from 1 to Length(FacReps) do
    U := Vollständiges Urbild von FacReps[i].K;
    S := Vollständiges Urbild von FacReps[i].C;
    tmp := OneCocycles(U, Nr, Rel_s);
    if tmp ≠ false then
      H1 := Liste aller Elemente der Einskohomologiegruppe tmp.Z1/tmp.B1;
      CN := tmp.CN;
      while H1 ≠ [] do
        h1 := H1[1];
        K := Kh1; # Ein zum 1-Kozykel h1 gehörendes Komplement.
        Orbit := Orbit von h1 unter der Operation von S/N;
        BlockStab := Stabilisator von h1 unter der Operation von S/N;
        Stab := []; # Berechne den Zentralisator für K
        for l from 1 to Length(BlockStab) do
          b := Ein Repräsentant von BlockStab[l] in G;
          n := KonjugierendesElement(U, Nr, K, Kb);
          Stab[i] := b * n-1;
        od;
      od;
    fi;
  od;

```

```

                Zentralisator := CGS(< Stab, CN >);
                Reps := Reps + (K, Zentralisator);
                H1 := H1 \ Orbit;
            od;
        fi;
    od;
if Reps = [] then
    return false;
else
    return Reps;
fi;
end.

```

4.4 Die affine Operation

Da der Orbitalgorithmus aus Algorithmus 4.1 in einer inneren Schleife des Algorithmus liegt, ist eine Beschleunigung wünschenswert. Wir wollen also nun den Orbitalgorithmus genauer untersuchen. Es wird sich in diesem Abschnitt zeigen, daß sich ein zeitaufwendiger “collection”-Prozeß und nicht-kommutativer Gaußalgorithmus zur Berechnung der Operation eines Elements aus S/N auf der 1-Kohomologiegruppe vermeiden lassen.

Es sei S und U wie im vorhergehenden Abschnitt. Es soll jetzt U zerfallen, ein Komplement K zu N in U sei schon bekannt und durch ein kanonisches Erzeugendensystem $\{k_1, \dots, k_l\}$ gegeben. N habe die Basis $\mathcal{N} = \{g_{i_r}, \dots, g_t\}$ und die Dimension $n = t - i_r + 1$. Zur Vereinfachung der Schreibweise sei

$$\begin{aligned} \smile & : \text{Verband}(G/N) \rightarrow \text{Verband}(G) \\ U/N \leq G/N & \mapsto U \leq G \end{aligned}$$

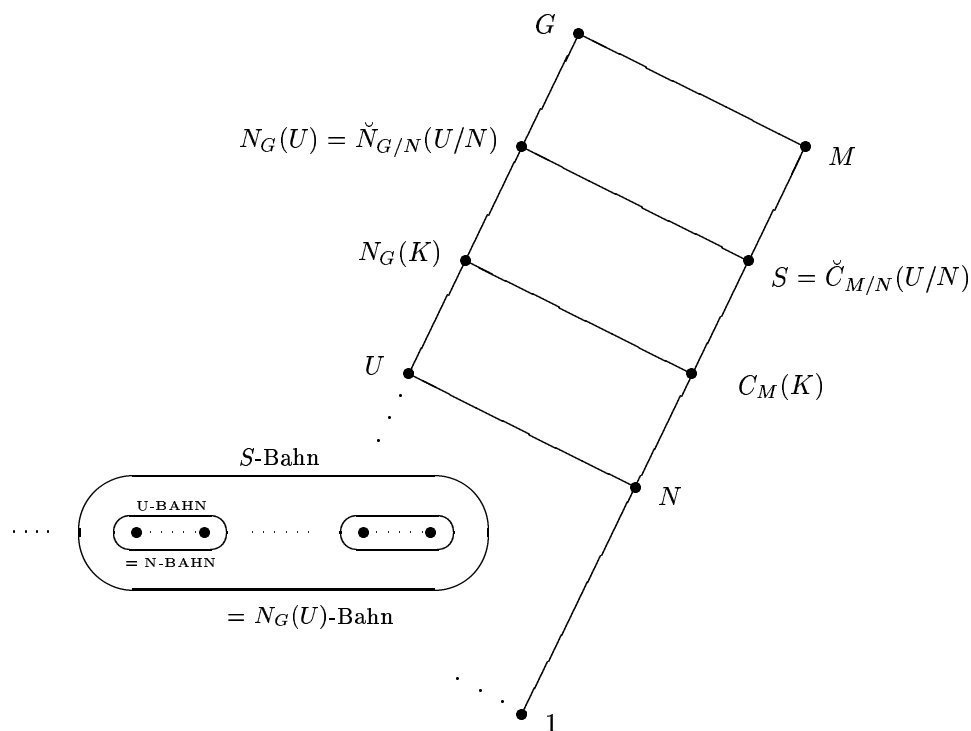
die Abbildung, welche einer Untergruppe U/N von G/N ihr vollständiges Urbild U in G zuordnet. Weiter sei $\mathcal{B} = \{\beta_1, \dots, \beta_b\}$ eine Basis der Gruppe $B^1(K, N)$ der 1-Koränder und $\{\zeta_1, \dots, \zeta_z\}$ sei eine Teilmenge von $Z^1(K, N)$ so, daß $\mathcal{Z} = \{\zeta_1, \dots, \zeta_z, \beta_1, \dots, \beta_b\}$ eine Basis der Gruppe $Z^1(K, N)$ der 1-Kozyklen ist. Dann ist $\mathcal{H} = \{\zeta_1 + B^1(K, N), \dots, \zeta_z + B^1(K, N)\}$ eine Basis der 1-Kohomologiegruppe $H^1(K, N)$.

Die Operation von Elementen aus S auf der Gruppe der 1-Kozykel $Z^1(K, N)$ ist durch

$$\gamma^s := \gamma_{K_\gamma^s}$$

für $\gamma \in Z^1(K, N)$ gegeben, wobei $\gamma_{K_\gamma^s}$ wie in Kapitel 2 definiert ist.

Für die auftretenden Untergruppen liegt also jetzt folgende Situation vor.



Die Operation von S auf dem Normalteiler N kann als eine lineare Abbildung aufgefaßt werden. Dies überträgt sich teilweise auf die Operation auf den 1-Kozykeln.

Satz 4.4 *Es seien die Voraussetzungen und Bezeichnungen wie oben. Es sei ein $s \in S$ und $\gamma \in Z^1(K, N)$ gegeben. Dann gilt*

$$\gamma^s = (k \in K \mapsto [k, s]\gamma(k)^s).$$

Beweis: Es sei ein beliebiges $\gamma \in Z^1(K, N)$ gegeben. Dann ist das zu γ gehörende Komplement K_γ gegeben durch $\{k\gamma(k) \mid k \in K\}$. Für $k \in K$ gilt nun

$$\begin{aligned} (k\gamma(k))^s &= k^s \gamma(k)^s \\ &= k \underbrace{[k, s]\gamma(k)^s}_{\in N}, \end{aligned}$$

denn s zentralisiert Elemente aus K modulo N nach Voraussetzung. Damit ist der 1-Kozykel, der zum zu K_γ unter s konjugierten Komplement K_γ^s gehört, durch $(k \in K \mapsto [k, s]\gamma(k)^s)$ gegeben. ‡‡

Damit ist die Operation affin, denn

Satz 4.5 *Die Voraussetzungen seien wie in Satz 4.4. Die Operation, welche von Elementen aus S auf dem Vektorraum $Z^1(K, N)$ induziert wird, ist affin.*

Beweis: Es sei ein $s \in S$ gegeben. Dann ist

$$\begin{aligned} \varphi_s &= (\gamma \mapsto (k \in K \mapsto \gamma(k)^s)) \\ &: C^1(K, N) \longrightarrow C^1(K, N) \end{aligned}$$

eine lineare Abbildung des Vektorraums $C^1(K, N)$ auf sich. Es sei nun ein beliebiges $\gamma \in Z^1(K, N)$ gegeben. Nach Satz 4.4 reicht es zu zeigen, daß die beiden Abbildungen $(k \in K \mapsto [k, s])$ und $\varphi_s(\gamma)$ wieder 1-Kozykel sind.

Für den trivialen 1-Kozykel $\underline{0}$ gilt aber nach Satz 4.4 $\underline{0}^s = (k \in K \mapsto [k, s])$ und damit ist diese Abbildung wieder ein 1-Kozykel. Somit gilt aber $\gamma - \underline{0}^s = \varphi_s(\gamma) \in Z^1(K, N)$.
‡‡

Die Operation von S auf den 1-Kozykeln überträgt sich auf die Operation von S/N auf der 1-Kohomologiegruppe $H^1(K, N)$.

Korollar 4.6 *Die Voraussetzungen seien wie in Satz 4.4. Die Operation, welche von Elementen aus S/N auf dem Vektorraum $H^1(K, N)$ induziert wird, ist affin. Für ein $sN \in S/N$ ist die lineare Operation durch*

$$\varphi_{sN} = (\gamma + B^1(K, N) \mapsto \varphi_s(\gamma) + B^1(K, N)) : H^1(K, N) \rightarrow H^1(K, N)$$

und die Verschiebung durch

$$\xi_{sN} = (k \in K \mapsto [k, s]) + B^1(K, N)$$

gegeben.

Beweis: Es sei ein $sN \in S/N$ und $\gamma + B^1(K, N) \in H^1(K, N)$ gegeben. Dann liegt γ in einem Block von $Z^1(K, N)$ unter der Operation von S auf $Z^1(K, N)$. Da aber nun für ein beliebiges $sn \in sN$

$$\begin{aligned} \gamma^{sn} &= (k \in K \mapsto [k, sn]\gamma(k)^{sn}) \\ &= (k \in K \mapsto [k, n][k, s]^n(\gamma(k)^s)^n) \\ &= (k \in K \mapsto [k, s]\gamma(k)^s[k, n]) \\ &= \gamma^s + (k \in K \mapsto [k, n]) \\ &\in \gamma^s + B^1(K, N), \end{aligned}$$

also gilt

$$(\gamma + B^1(K, N))^{sN} = \gamma^s + B^1(K, N),$$

und damit ist die Operation ebenfalls affin. $\ddagger\ddagger$

Zum Rechnen ist jedoch eine affine Operation im allgemeinen zu umständlich. Man kann jedoch bekanntlich jede affine Operation auf einem n -dimensionalen Vektorraum mit Basis \mathcal{N} auffassen als eine lineare Operation auf einem $n + 1$ dimensionalen Vektorraum. Es sei $\mathcal{V} = \mathcal{V}_{1 \times n+1}(\mathbf{Z}_{q_r})$, dann ist die der affinen Operation von sN auf $H^1(K, N)$ entsprechende lineare Operation λ_{sN} von sN auf \mathcal{V} gegeben durch

$$\lambda_{sN} = \begin{pmatrix} m(\mathcal{N}, \varphi_{sN}, \mathcal{N}) & 0 \\ m(\mathcal{N}, \xi_{sN}) & 1 \end{pmatrix}.$$

Das heißt, die injektive Abbildung

$$\varrho = \gamma + B^1(K, N) \mapsto (m(\mathcal{H}, \gamma + B^1(K, N)), 1) : H^1(K, N) \longrightarrow \mathcal{V}$$

ist mit den Operationen von S/N auf $H^1(K, N)$ und \mathcal{V} vertauschbar und es ist somit für beliebige $\gamma + B^1(K, N) \in H^1(K, N)$ und $sN \in S/N$

$$\varrho((\gamma + B^1(K, N))^{sN}) = \varrho(\gamma + B^1(K, N)) \cdot \lambda_{sN}.$$

Wir können also die Operationen in den Orbitalgorithmen bei der S -Bahnzerlegung von $H^1(K, N)$ durch Matrixmultiplikation beschreiben.

4.5 Der zentrale Fall

Je näher die Dimension des Vektorraums $B^1(K, N)$ bei der Dimension des Vektorraums $Z^1(K, N)$ liegt, desto kleiner ist die Menge $H^1(K, N)$, von der eine S -Bahnzerlegung durchgeführt werden soll. Gilt sogar $Z^1(K, N) = B^1(K, N)$, so ist die 1-Kohomologiegruppe $H^1(K, N)$ trivial und damit auch die Operation von S auf $H^1(K, N)$.

Falls jedoch N ein zentraler Normalteiler von G ist, so ist der Faktorraum $H^1(K, N)$ der Gruppe $Z^1(K, N)$ der 1-Kozykeln maximal, da die Gruppe $B^1(K, N)$ der 1-Koränder trivial ist, denn N operiert nicht auf der Menge der Komplemente. In diesem Fall läßt sich jedoch auf einen Orbitalgorithmus völlig verzichten. Wir werden hierzu ein Lemma aus [MN89] verallgemeinern, welches bei dem Herunterziehen der Konjugiertenklassen von Elementen bei einem zentralen Normalteiler benutzt wurde.

Offensichtlich ist die Gruppe $B^1(K, N)$ auch dann schon trivial, wenn N zentral unter K ist, dies reicht jedoch im folgenden nicht aus. Es wird sich jedoch zeigen, daß N nur von S und K zentralisiert werden braucht, wir wollen daher dies in diesem Abschnitt voraussetzen. Die sonstigen Voraussetzungen seien wie im vorhergehenden Abschnitt.

Lemma 4.7 *Die Voraussetzungen seien wie oben. Es bezeichne $\underline{0} = (k \in K \mapsto 1) \in Z^1(K, N)$ den trivialen Kozykel. Dann ist der Orbit $\underline{0}^S$ von $\underline{0}$ unter der Operation von S ein Teilraum von $Z^1(K, N)$.*

Beweis: Es seien $\gamma_1, \gamma_2 \in \underline{0}^S$ beliebig. Dann existieren nach Satz 4.4 $s_1, s_2 \in S$, so daß

$$\gamma_i = (k \in K \mapsto [k, s_i])$$

für $i \in \{1, 2\}$. Für die Summe von γ_1 und γ_2 folgt dann per Definition

$$\gamma_1 + \gamma_2 = (k \in K \mapsto [k, s_1] \cdot [k, s_2]).$$

Da $[k, s_1] \in N$ für alle $k \in K$, werden diese Elemente von s_2 nach Voraussetzung zentralisiert. Es folgt somit für beliebige $k \in K$,

$$\begin{aligned} [k, s_1 s_2] &= [k, s_2] \cdot [k, s_1]^{s_2} \\ &= \underbrace{[k, s_2] \cdot [k, s_1]}_{\in N, \text{abelsch!}} \\ &= [k, s_1] \cdot [k, s_2]. \end{aligned}$$

Damit gilt aber

$$\begin{aligned} \gamma_1 + \gamma_2 &= (k \in K \mapsto [k, s_1] \cdot [k, s_2]) \\ &= (k \in K \mapsto [k, s_1 s_2]) \\ &= \underline{0}^{s_1 s_2} \in \underline{0}^S. \end{aligned}$$

Analoges gilt für die Skalarmultiplikation, also ist $\underline{0}^S$ ein Teilraum von $Z^1(K, N)$. $\ddagger\ddagger$

Es gilt aber sogar

Lemma 4.8 *Die Voraussetzungen und Bezeichnungen seien wie in Lemma 4.7. Es sei ein $\gamma \in Z^1(K, N)$ gegeben. Dann gilt*

$$\gamma^S = \gamma + \underline{0}^S.$$

Beweis: Es sei ein $s \in S$ beliebig. Dann gilt mit Satz 4.4

$$\begin{aligned} \gamma^s &= (k \in K \mapsto \gamma(k))^s \\ &= (k \in K \mapsto [k, s] \underbrace{\gamma(k)^s}_{\in N}) \\ &= (k \in K \mapsto [k, s] \gamma(k)) \\ &= \gamma + \underline{0}^s. \end{aligned}$$

Dies zeigt aber in beiden Richtungen gelesen die Behauptung. $\ddagger\ddagger$

Damit zeigt sich

Satz 4.9 Die Voraussetzungen und Bezeichnungen seien wie in Lemma 4.7 beschrieben. Dann ist jedes Repräsentantensystem für die S -Bahnen von $Z^1(K, N)$ —und damit für $H^1(K, N) \cong Z^1(K, N)$ —auch ein Repräsentantensystem für die Restklassen von $Z^1(K, N)$ nach dem Teilraum $\underline{0}^S$ und umgekehrt.

Das heißt, anstelle einer S -Bahnzerlegung durch Orbitalgorithmen, benötigen wir nur eine Basis für den Faktorraum $Z^1(K, N)/\underline{0}^S$. Dies liefert dann jedoch im allgemeinen eine recht große Anzahl von Konjugiertenklassen von Komplementen.

Für den Fall daß $\underline{0}^S = \{\underline{0}\}$ gilt, das heißt, S operiert nicht und zentralisiert damit K , folgt $C_M(K) = S$. Für den Fall daß S doch operiert, läßt sich der Zentralisator jedoch auch ohne explizite Durchführung des Stabilisatoralgorithmus berechnen.

Satz 4.10 Die Voraussetzungen und Bezeichnungen seien wie in Lemma 4.7 beschrieben. $S \geq N = \langle g_{i_r}, \dots, g_t \rangle$ habe das kanonische Erzeugendensystem $(s_1, \dots, s_m, g_{i_r}, \dots, g_t)$. Dann erzeugt $\mathcal{E} = (E_m = \underline{0}^{s_m}, \dots, E_1 = \underline{0}^{s_1})$ den Teilraum $\underline{0}^S$. Wenn aus \mathcal{E} der Reihe nach—also mit E_m beginnend und bei E_1 endend—eine Basis $\mathcal{C} = (C_r = E_{i_r}, \dots, C_1 = E_{i_1})$ mit $m \geq i_r \geq \dots \geq i_1 \geq 1$ ausgewählt wird, so existieren für jedes E_k mit $k \notin \{i_1, \dots, i_r\}$ Zahlen $e_i^{(k)}$ in \mathbf{Z}_{q_r} mit

$$E_k = C_1 * e_1^{(k)} + \dots + C_r * e_r^{(k)}.$$

Mit dieser Setzung hat der Zentralisator $C_M(K)$ das nach Normierung kanonische Erzeugendensystem

$$\{s_k * \left(s_{i_1}^{e_1^{(k)}} \cdots s_{i_r}^{e_r^{(k)}} \right)^{-1} \mid k \in \{1, \dots, m\} \setminus \{i_1, \dots, i_r\}\} \cup \{g_{i_r}, \dots, g_t\}.$$

Beweis: Wir beweisen den Satz durch Induktion über die Schleife des Orbitalgorithmus. Es gilt $\underline{0}^N = \{\underline{0}\}$, denn N wird von K zentralisiert. Es sei also $C = N$. Dies zeigt den Induktionsanfang. Die Behauptung gelte also für $i + 1 \in \{r, \dots, 2\}$. Dann gilt $\underline{0}^{s_i} \in \underline{0}^{\langle s_m, \dots, s_{i+1} \rangle}$ genau dann, wenn $E_i \in \langle E_m, \dots, E_{i+1} \rangle$. Dies ist jedoch genau dann der Fall, wenn E_i nicht zu der Basis \mathcal{C} gehört, also gilt dann $i \notin \{i_1, \dots, i_r\}$. Dann folgt aber per Konstruktion

$$\underline{0}_i^s = \underline{0} \left(s_{i_1}^{e_1^{(i)}} \cdots s_{i_r}^{e_r^{(i)}} \right)$$

und wir müssen

$$s_i * \left(s_{i_1}^{e_1^{(i)}} \cdots s_{i_r}^{e_r^{(i)}} \right)^{-1}$$

zu \mathcal{C} hinzufügen. Im anderen Fall, das heißt, $i \in \{i_1, \dots, i_r\}$, bleibt \mathcal{C} unverändert. Dies zeigt aber schon die Behauptung. $\ddagger\ddagger$

Es gilt sogar

Lemma 4.11 *Die Voraussetzungen und Bezeichnungen seien wie in Lemma 4.7 beschrieben. Es sei ein beliebiges $\gamma \in Z^1(K, N)$ gegeben. Dann gilt $C_S(K) = C_S(K_\gamma)$.*

Beweis: Es sei ein beliebiges $s \in C_S(K)$ gegeben. Dann zentralisiert s per Voraussetzung die Elemente aus K und N , und damit insbesondere $\gamma(k)$ und k für beliebige $k \in K$. Es sei ein $k' \in K_\gamma$ beliebig, dann existiert nach Satz 2.10 ein $k \in K$ mit $k' = k\gamma(k)$. Es folgt

$$\begin{aligned} (k')^s &= (k\gamma(k))^s \\ &= k^s \gamma(k)^s \\ &= k\gamma(k) \\ &= k'. \end{aligned}$$

Also gilt $s \in C_S(K_\gamma)$. Umgekehrt folgt entsprechend $C_S(K) \geq C_S(K_\gamma)$. ‡‡

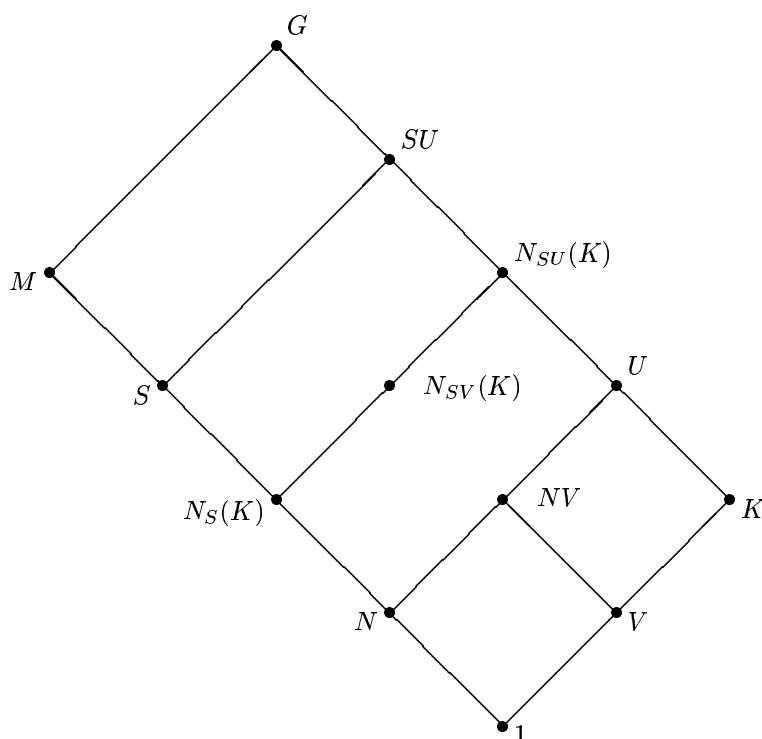
Wir können also bei einem Normalteiler N , welcher von S und K zentralisiert wird, die S -Bahnzerlegung und die Berechnung der Zentralisatoren auf das Lösen linearer Gleichungssysteme reduzieren.

4.6 Verfeinerung der Rekursion

Sollte das Herunterziehen von Komplementen in einem Schritt zu schwierig sein, so läßt sich das Problem eventuell wie folgt reduzieren. Die hier angesprochenen Verfahren gehen auf Vorschläge von C. Wright zurück—siehe [Wri88] und [Wri90].

Es sei also ein Komplement U/N in G/N zu M/N sowie der Normalisator S/N von U/N in M/N bekannt. Statt die Komplemente zu N in U zu bestimmen, kann man versuchen, eine SU -normale Untergruppe W von N zu bestimmen und dann das Homomorphieprinzip anzuwenden. Man beachte, daß hierbei die Folge $(N, W, 1)$ im allgemeinen nicht mehr von der Kompositionsreihe verfeinert wird. Dies hat jedoch eine aufwendigere Berechnung der Produkte und Schnitte zur Folge, für den Algorithmus selbst spielt diese Tatsache keine Rolle. Ein weiterer Nachteil ist die Tatsache, daß, obwohl U/W viele Konjugiertenklassen von Komplemente besitzen kann, nur wenige von diesen auch zerfallen zu brauchen—siehe [Wri88].

Eine andere Möglichkeit der Reduktion ist gegeben, falls eine SU -normale Untergruppe V im Durchschnitt aller Komplemente a priori bekannt ist. In diesem Fall liegt folgende Situation vor.



Denn es gilt

Lemma 4.12 *Die Voraussetzungen und Bezeichnungen seien wie oben. Dann entspricht jedem Komplement K zu N in U eindeutig ein Komplement K/V zu NV/V in U/V und umgekehrt. Der Normalisator $N_S(K)$ ist dann der Schnitt von S mit dem vollständigen Urbild von $N_{SV/V}(K/V)$.*

Beweis: Da jedes K nach Voraussetzung V enthält, ist die Zuordnung der Komplemente klar.

Mit Lemma 3.3 folgt $N_{SV/V}(K/V) = N_{SV}(K)/V$ und damit $N_{SV}(K) \cap S = N_S(K)$. $\ddagger\ddagger$

Da die Anzahl der Relationen quadratisch von der Kompositionslänge von U/N abhängt, lohnt sich dieses Verfahren für solche V , für welche die Kompositionslänge U/NV klein wird. Es ist jedoch zu beachten, daß für U/NV in jedem Schritt eine neue Potenz-Kommutator Präsentation berechnet werden muß. Zum anderen muß für jeden Repräsentanten U eine

Untergruppe V berechnet werden. Diesen beiden Nachteilen steht der Vorteil eines kleineren Gleichungssystems gegenüber.

Nach [Gam90] läßt sich ein solches V bestimmen durch

Lemma 4.13 *Die Voraussetzungen seien wie oben. N sei ein \mathbf{Z}_p -Vektorraum. Es sei $C = C_U(N)$ bekannt. Dann liegt $V = C^p[C, C]$ in jedem Komplement K und ist SU -normal.*

Beweis: Da sowohl N als auch U normal in SU sind, folgt $C \triangleleft SU$ und damit ist auch V normal in SU . Es sei ein beliebiges Komplement K gegeben. Dann ist $C \cap K \triangleleft K$, da C normal in U ist, und $C \cap K \triangleleft N$, da $C \cap K \leq C_U(N)$. Damit ist aber $C \cap K$ normal in $KN = U$. Da nun $C/C \cap K$ ein zu N isomorpher \mathbf{Z}_p -Vektorraum ist, folgt $C \cap K \geq V$. Da K beliebig war, folgt die Behauptung. $\ddagger\ddagger$

Man beachte, daß nur zu Beginn des Algorithmus $C_G(N)$ berechnet zu werden braucht, denn $C_U(N)$ läßt sich für jedes U durch Schnittbildung von U mit dem Normalteiler $C_G(N)$ berechnen—also insbesondere ist zur Schnittberechnung nur ein erweiterter Gaußalgorithmus notwendig.

4.7 Normale Komplemente

Da manchmal selbst ein Repräsentantensystem für die Konjugiertenklassen von Komplementen zu einem Normalteiler zu aufwendig zu berechnen ist, wäre es wünschenswert, sich auf normale Komplemente zu beschränken.

Eine etwas allgemeinere Situation liegt vor, wenn man in einer Gruppe eine Normalreihe kennt, und wissen möchten, ob sich Faktoren dieser Reihe nach unten schieben lassen.

Insgesamt lassen sich diese beiden Fälle wie folgt zusammenfassen. Es sei also ab jetzt eine endliche polyzyklische Gruppe G mit einem Normalteiler H gegeben. Dieser Normalteiler enthalte eine G -normale Untergruppe M . Gesucht ist nun die Menge der G -normalen Komplemente zu M in H .

Es sei weiter ein G -normale Untergruppe N von M gegeben. Dann gilt

Lemma 4.14 *Die Voraussetzungen seien wie oben angegeben. Dann liefert jedes G -normale Komplement K zu M in H ein G/N -normales Komplement KN/N zu M/N in H/N .*

Beweis: Nach Lemma 4.2 ist KN/N ein Komplement zu M/N in H/N . Da sowohl K als auch N Normalteiler von G sind, ist auch ihr Produkt normal in G . Dies überträgt sich auf die Faktorgruppe. $\ddagger\ddagger$

Lemma 4.14 erlaubt es nun, rekursiv die G -normalen Komplemente zu M in H zu berechnen. Wir können also per Induktion annehmen, daß für eine G -normale Untergruppe N von M die Menge der G/N -normalen Komplemente $\{U_1/N, \dots, U_k/N\}$ zu M/N in H/N

bekannt sei. Dann müssen wir für jedes U_i die Menge der G -normalen Komplemente zu N in U_i berechnen.

Es sei also jetzt $U = U_i$ für ein festes $i \in \{1, \dots, k\}$ und N ein elementar abelscher Normalteiler von G . Es gilt

Satz 4.15 *Es seien die Voraussetzung wie in Lemma 4.14 und N sei zusätzlich ein elementar abelscher Normalteiler der Ordnung p^m . Der Normalteiler M habe das kanonische Erzeugendensystem (m_1, \dots, m_m) . Weiter sei U/N ein G/N -normales Komplement zu M/N in H/N , gegeben durch ein kanonisches Erzeugendensystem (u_1N, \dots, u_lN) und mit den Potenz-Kommutatorrelatoren $r_1, \dots, r_{l(l+1)/2}$. Es sei (g_1H, \dots, g_tH) ein kanonisches Erzeugendensystem für die Faktorgruppe G/H .*

Dann wird jedes G -normale Komplement zu N in U durch eine Funktion $f : \{u_1, \dots, u_l\} \rightarrow N$ beschrieben, welche folgendes inhomogene Gleichungssystem erfüllt

$$(I) \quad r_i(u_1f(u_1), \dots, u_lf(u_l)) = 1 \text{ für alle } i \in \{1, \dots, l(l+1)/2\},$$

$$(II) \quad [m_i, u_jf(u_j)] = 1 \text{ für alle } i \in \{1, \dots, m\} \text{ und } j \in \{1, \dots, l\},$$

$$(III) \quad (u_if(u_i))^{g_j} = \omega_{ij}(u_1f(u_1), \dots, u_lf(u_l)) \text{ für alle } i \in \{1, \dots, l\} \text{ und } j \in \{1, \dots, t\},$$

dabei sind die ω_{ij} Worte, so daß $(u_iN)^{g_jN} = \omega_{ij}(u_1N, \dots, u_lN)$ gilt.

Die Worte ω_{ij} lassen sich berechnen, da U/N eine endliche, polyzyklische Gruppe ist. Sie existieren, da U/N ein Normalteiler von G/N ist.

Beweis: Nach Satz 2.10 beschreibt jede Funktion f , welche das Gleichungssystem (I) löst, ein Komplement und umgekehrt. Nach Satz 2.12 ist diese Zuordnung eindeutig.

Wir zeigen zuerst, daß die Gleichungen (II) und (III) linear sind. Es gilt für alle $i \in \{1, \dots, m\}$ und $j \in \{1, \dots, l\}$, daß

$$\begin{aligned} [m_i, u_j] &= \underbrace{(u_j^{-1})^{m_i} \cdot u_j}_{\in U} \\ &= m_i^{-1} \cdot \underbrace{(m_i)^{u_j}}_{\in M} \\ &\in U \cap M = N, \end{aligned}$$

denn sowohl U als auch M sind Normalteiler von G . Andererseits operiert G linear auf den Elementen $f(u_j) \in N$, also liefert dies analog Lemma 2.11 ein inhomogenes lineares Gleichungssystem.

Per Konstruktion gilt für alle $i \in \{1, \dots, l\}$ und $j \in \{1, \dots, t\}$, daß

$$u_i^{g_j} / \omega_{ij}(u_1, \dots, u_t) \in N,$$

so daß sich auch hier ein inhomogenes Gleichungssystem ergibt.

Es sei jetzt ein beliebiges Komplement K zu N in U gegeben, f eine dazu gehörige Funktion, welche Gleichungssystem (I) löst. Dann ist $K = \langle u_1 f(u_1), \dots, u_l f(u_l) \rangle$ genau dann normal in G , wenn es von allen Erzeugern von G normalisiert wird. Die Gruppe G wird nach Voraussetzung von $\{g_1, \dots, g_t\}$ und H erzeugt. Da K ein Komplement zu M in H ist, wird H von $\{u_1 f(u_1), \dots, u_l f(u_l), m_1, \dots, m_m\}$ erzeugt.

Nun ist K , da es ein Komplement zu M in H ist, genau dann H -normal, wenn es von M zentralisiert wird, das heißt, die Erzeuger von K erfüllen das Gleichungssystem (II). Ein g_j für beliebige $j \in \{1, \dots, t\}$ normalisiert K genau dann, wenn $(u_i f(u_i))^{g_j} \in K$ für alle $i = 1, \dots, l$ gilt. Dies gilt aber genau dann, wenn Gleichung (III) für dieses j und alle i erfüllt ist. Denn gilt (III) für ein i und j , so folgt sofort $(u_i f(u_i))^{g_j} \in K$. Gilt dagegen $(u_i f(u_i))^{g_j} \in K$, so folgt

$$(u_i f(u_i))^{g_j} / \omega_{ij}(u_1 f(u_1), \dots, u_l f(u_l)) \in K \cap N = \{1\}.$$

Also gilt dann (III).

Dies zeigt insgesamt die Behauptung. ‡‡

Literatur

Das grundlegende Verfahren der rekursiven Berechnung der Komplemente entlang einer Normalreihe mit elementar abelschen Faktoren ist eine Erweiterung des Konjugiertenklassenalgorithmus aus [MN89].

Ein Algorithmus zur Berechnung von Komplementen in p -Gruppen ist in [LNS84] beschrieben.

Kapitel 5

Kernfunktionen für den PQ

In diesem Kapitel werden die Kernfunktionen und Veränderungen an der internen Darstellung der endlichen, polyzyklischen Gruppen in GAP beschrieben, welche für einen PQ (siehe [Nie91]) in GAP 3.1 notwendig wurden.

Das Handlekonzept sowie die Müllabfuhr (“Garbage Collection”) ist ausführlich beschrieben in [Sch87]. Im folgendem werden die Begriffe “Handle” und “Typ” als bekannt vorausgesetzt.

5.1 Endliche, polyzyklische Gruppen in GAP

Abweichend von der Darstellung in GAP 2.4 (siehe [Bis89]) werden endliche, polyzyklische Gruppen in GAP 3.1 durch Listen dargestellt, welche jedoch nicht den Typ `T_LIST` sondern den Typ `T_AGGRP` tragen. Gruppen in dieser Darstellung in GAP heißen Ag-Gruppen. Der Zugriff auf die Informationen einer Ag-Gruppe G sollte jedoch nicht durch Angabe eines Index erfolgen, sondern unter Benutzung eines der folgenden Macros, so daß sich Erweiterungen ohne Veränderung der bestehenden Funktionen einbauen lassen.

`HD_NUMBER_OF_GENS`

Dieses Macro liefert den Handle einer Zahl vom Typ `T_INT`, welche die Anzahl der Ag-Erzeuger der Gruppe G angibt.

`HD_GENERATORS`

Dieses Macro liefert den Handle einer Liste vom Typ `T_LIST`, welche die Erzeuger der Gruppe G enthält. Diese ließen sich bei Bedarf zwar leicht konstruieren, jedoch hat sich im PQ gezeigt, daß dadurch zuviel Müll produziert wird.

`HD_IDENTITY`

Dieses Macro liefert den Handle des Einselementes der Gruppe G vom Typ `T_AGWORD`.

HD_POWERS

Dieses Macro liefert den Handle einer Liste vom Typ `T_LIST`, welche die rechten Seiten der Potenzrelationen enthält. Triviale Einträge sind hier gebunden und zeigen auf das Einselement der Gruppe G .

HD_COMMUTATORS

Dieses Macro liefert den Handle einer Liste vom Typ `T_LIST`, welche die rechten Seiten der Kommutatorrelationen enthält. Triviale Einträge zeigen hierbei auf das Einselement der Gruppe G . Die rechten Seiten sind linear angeordnet, dies kann im PQ zu Platzproblemen führen, da von vielen der Einträgen bekannt ist, daß sie trivial sind. Unter Umständen wäre eine Abspeicherung als Liste von Listen angebracht, welche jedoch den Kollektor verlangsamen würde.

HD_INDICES

Dieses Macro liefert den Handle eines C-Feldes vom Typ `long`, in welchem die relativen Ordnungen der Erzeuger von G abgespeichert sind.

HD_COLLECTOR

Dieses Macro liefert den Handle einer Zahl vom Typ `T_INT`, welche angibt, welcher Kollektor für G verwendet wird. Hierbei steht `SINGLE_COLLECTOR` für einen Single-Kollektor, `TRIPLE_COLLECTOR` für einen Triple-Kollektor, `QUADR_COLLECTOR` für einen Quadruple-Kollektor, `COMBI2_COLLECTOR` für einen kombinatorischen Kollektor in 2-Gruppen, `COMBI_COLLECTOR` für einen kombinatorischen Kollektor in p -Gruppen (für diese Kollektoren siehe [Bis89]) sowie `LEE_COLLECTOR` für einen kombinatorischen Kollektor in p -Gruppen basierend auf einer Idee von M. Vaughan-Lee und implementiert von Alice Niemeyer. Alle Kollektoren sind Kollektoren von links.

HD_WORDS

Dieses Macro liefert den Handle einer Liste vom Typ `T_LIST`, welche die abstrakten Erzeuger vom Typ `T_AGEN` enthält, die zur Konstruktionen der Gruppe G benutzt wurden. Im Zusammenhang mit Ag-Wörtern wird dieser Eintrag nur zum Ausdrucken verwendet, in den PQ Funktionen spielt er jedoch eine wichtige Rolle, da er dort zur Konversion zwischen Wörtern und Pc-Wörtern verwandt wird.

HD_SAVE_EXPONENTS

Dieses Macro liefert den Handle eines Exponentenvektors vom Typ `T_AGEXP`, welcher während des Kollektens dazu benutzt wird, den ursprünglichen Exponentenvektor zu erhalten, falls der Kollektor wegen zu kleinen Stacks abgebrochen werden muß. Im PQ hat sich jedoch gezeigt, daß auf Grund der Länge der dort auftretenden Exponentenvektoren die Zeit nicht zu vernachlässigen ist, welche zum Kopieren des ursprünglichen Exponentenvektors nötig ist. Daher wird, statt den Exponentenvektor zu kopieren, das ursprüngliche Ag-Wort mitübergeben.

HD_COLLECT_EXPONENTS, HD_COLLECT_EXPONENTS2

Diese beiden Macros liefern Exponentenvektoren, welche für den Kollektor verwendet werden können. Dadurch werden in einem typischen PQ Lauf ungefähr 8 Mbyte Müll eingespart.

Die Macros HD_STACKS, HD_CSERIES, HD_CWEIGHTS, HD_AVEC, HD_CONJUGATES, HD_TRIPLES, HD_QUADRUPLES und HD_TUPLE_BOUND liefern die kollektorabhängigen Informationen und sind im wesentlichen in [Bis89] beschrieben.

5.2 Kommutator- und Konjugationsalgorithmus

Wie in [Bis89] beschrieben, wird in GAP 2.4 der Kommutator zweier Ag-Worte w_1 und w_2 dadurch berechnet, daß zuerst $w_{12} = w_1 * w_2$ und $w_{21} = w_2 * w_1$ ausgerechnet und danach die Gleichung $w_{21} * x = w_{12}$ nach x aufgelöst wird.

Der Algorithmus zum Lösen einer Gleichung $a * x = b$ läßt sich nun basierend auf einer Idee von O'Brian und Newman verallgemeinern auf eine Gleichung vom Typ $a * b * x = c * d$. Wählt man $a = d = w_2$ und $b = c = w_1$, so erhält man als Lösung x den Kommutator von w_1 und w_2 , für $a = 1$, $b = d = w_2$ und $c = w_1$, so erhält man $w_1^{w_2}$.

Im schlechtesten Fall verhält sich der Algorithmus im wesentlichen wie der zur Lösung von $w_{12} * x = w_{21}$ benötigte, jedoch wird im allgemeinen Fall unter Umständen weniger kollektet, da gemeinsame Präfixe von vornherein gekürzt werden können.

Algorithmus 5.1. (Gleichungssystem lösen)

Eingabe : • Eine endliche, polyzyklische Gruppe G mit CGS (g_1, \dots, g_t) , es sei dabei o_i die relative Ordnung von g_i .

• Elemente a, b, c und d von G in Normalform bezüglich (g_1, \dots, g_t) .

Ausgabe : Ein Element x von G in Normalform bezüglich (g_1, \dots, g_t) , so daß x die Lösung der Gleichung $a * b * x = c * d$ ist.

```
AgSolution := function( G, a, b, c, d )
```

```
  Suche gemeinsames Präfix  $w$  in  $a$  und  $c$ ;
```

```
   $a := w^{-1} * a$ ;
```

```
   $c := w^{-1} * c$ ;
```

```
   $f := c * d$ ;
```

```
   $x := 1$ ;
```

```
  for  $i \in [1..t]$  do
```

```
    Es sei  $a = g_i^{e_a} * w_a$  für  $0 \leq e_a < o_i$  und  $e_a$  maximal,  $w_a$  in Normalform;
```

```
    Es sei  $b = g_i^{e_b} * w_b$  für  $0 \leq e_b < o_i$  und  $e_b$  maximal,  $w_b$  in Normalform;
```

```
    Es sei  $f = g_i^{e_f} * w_f$  für  $0 \leq e_f < o_i$  und  $e_f$  maximal,  $w_f$  in Normalform;
```

```
     $e := e_f - e_a - e_b \bmod o_i$ ;
```

```
     $x := x * g_i^e$ ;
```

```

    b := b * g_i^e;
    Es sei b = g_i^{e_b} * w_b für 0 ≤ e_b < o_i und e_b maximal, w_b in Normalform;
    b := w_b;
    a := a * g_i^{e_b};
    Es sei a = g_i^{e_a} * w_a für 0 ≤ e_a < o_i und e_a maximal, w_a in Normalform;
    a := w_a;
    f := w_f;
od;
return x;
end.

```

Folgende Tabelle gibt einen Überblick über das Zeitverhalten des Algorithmus. Die Gruppen sind in 7 beschrieben. Es wurden 50 zufällige Wörter berechnet und alle möglichen Kommutatoren gebildet. "Solution 1" bezeichnet das alte Verfahren, "Solution 2" das neue, die Zeiten sind in Millisekunden gemessen auf einer SparcStation 2.

Gruppe	Solution 1	Solution 2
sla1	35,800	34,883
sla2	160,583	160,067
sla3	111,400	109,383

Insgesamt ergibt sich also bei zufällig gewählten Wörtern keine wesentliche Verbesserung, falls jedoch die Wörter w_1 und w_2 lange gemeinsame Präfixe haben, führt dies zu einer deutlichen Verbesserung, wie folgende Tabelle zeigt. Es wurden 200 zufällige Wörter berechnet und diese mit einem zufälligen Element der letzten nichttrivialen Gruppe der elementar abelschen Reihe multipliziert. Dann wurden die Kommutatoren zwischen diesen Wörtern gebildet.

Gruppe	Solution 1	Solution 2
sla1	1,983	1,800
sla2	8,750	2,083
sla3	7,033	3,535

5.3 Ag-Gruppen und Präsentationen

Endliche, polyzyklische Gruppen werden in GAP, wie oben erwähnt, durch Ag-Gruppen vom Typ T_AGGRP dargestellt. Die Elemente einer solchen Gruppe G in ihrer kanonischen Darstellung heißen Ag-Worte. Insbesondere gehört jedes Ag-Wort zu einer bestimmten Ag-Gruppe. Zwei Ag-Worte w_1 und w_2 einer Ag-Gruppe G können durch die üblichen Infixoperatoren $*$, $/$ und \wedge verknüpft werden. Dabei braucht die Gruppe G nicht mitangegeben werden, die Ag-Worte tragen die zur Verknüpfung notwendigen Informationen mit sich.

Einem Wort w ist somit eine feste Bedeutung als Element von G zugeordnet, diese Zuordnung ist in allen Algorithmen für Ag-Gruppen notwendig, da sonst die Verknüpfung nicht mehr durch $*$ erfolgen könnte. Dies würde desweiteren eine Verwendung von generischen, darstellungsunabhängigen Algorithmen unmöglich machen.

Folglich darf sich die Gruppe G selbst sowie ihre Präsentation nicht ändern, denn eine Veränderung der Präsentationen hätte zur Folge, daß Ag-Wörter, welche an ganz anderen Stellen eines Programms noch gespeichert sind, andere Elemente von G bezeichnen würden.

In einem PQ-Algorithmus, welcher laufend eine Präsentation verändern muß, entweder um neue Erzeuger hinzuzufügen, redundante zu entfernen oder inkonsistente Präsentationen zu korrigieren, sind Ag-Gruppen ungeeignet, obwohl die vorhandenen Kernkollektoren verwendet werden könnten.

Funktionen für den PQ müssen sich von Konzept einer festen Ag-Gruppe lösen und dafür den neu geschaffenen Typ einer Pc-Präsentation verwenden. Diese Pc-Präsentationen beschreiben nicht eine feste Gruppe, sondern eine veränderbare Präsentation. Eine Präsentation P besteht aus Erzeugern, welche sich wie Elemente einer freien Gruppe verhalten, also vollkommen unabhängig von P mit $*$ frei multipliziert werden, und einer Sammlung von Pc-Relationen in diesen Erzeugern.

Um Wörter w_1 und w_2 bezüglich einer gegebenen Präsentation P zu verknüpfen, muß man die entsprechenden Funktionen `ProductPcp`, `QuotientPcp`, `LeftQuotientPcp`, `CommPcp` und `ConjugatePcp` verwenden. Alle diese Funktionen erwarten als erstes Argument eine Pc-Präsentation P und danach Wörter w_1 und w_2 in den Erzeugern von P . Sie liefern das Ergebnis als Wort in Normalform bezüglich P . Die Funktion `NormalWordPcp` liefert die Normalform eines Wortes w bezüglich P .

Die Verknüpfung durch Funktionen hat den Vorteil, daß nun die Wörter keinen Bezug mehr zu einer festen Präsentation haben; sollte also eine Pc-Präsentation geändert werden, beeinflußt dies nicht mehr die Bedeutung eines eventuell an anderer Stelle vorhanden Wortes, denn ein Wort w enthält seine Bedeutung als Element der durch P repräsentierten Gruppe G erst durch Angabe von P in den entsprechenden Funktionen.

5.4 Pc-Präsentationen

Würde man Pc-Präsentationen so implementieren, wie im letzten Abschnitt beschrieben, so müßte bei jeder Verknüpfung überprüft werden, ob die beteiligten Wörter w_1 und w_2 auch Wörter in den Erzeugern einer gegebenen Präsentation P sind und ob sie sich schon in Normalform befinden. Dies würde zu einem beachtlichen Overhead führen.

Daher wurden in GAP 3.1 Pc-Präsentationen und ihre Elemente so implementiert, daß sie sich auf GAP Ebene so verhalten, wie im vorhergehenden Abschnitt beschrieben, ihre interne Darstellung jedoch Ag-Gruppen und Ag-Wörtern gleicht.

Im folgenden bezeichne "Wörter" eine Folge von abstrakten Erzeugern ohne jegliche Zusatzinformation, "Ag-Wörter" Elemente einer Ag-Gruppe wie in [Bis89] beschrieben, "Pc-Wörter" Elemente einer Pc-Präsentation und "S-Wörter" freie Wörter wie unten

beschrieben. “Abstrakte Wörter” sind entweder “Wörter”, “S-Wörter” oder “Pc-Wörter”, sie sind auf GAP Ebene nicht von einander zu unterscheiden.

Wörter vom Typ `T_WORD` sind als Folge von Handles vom Typ `T_AGEN` realisiert. Diese abstrakten Erzeuger von Typ `T_AGEN` werden durch die Funktion `Word` angelegt, sie gelangen jedoch nie direkt an die GAP Ebene. S-Wörter vom Typ `T_SWORD` sind als Folge von Paaren Generatornummer und Exponent realisiert. Ihr erster Eintrag zeigt auf eine Liste von abstrakten Erzeugern, die Generatornummer ist als Index in dieser Liste zu verstehen. Wann immer in einer Verknüpfung ein S-Wort w_1 auf ein Wort w_2 trifft, wird eine Kopie w_1' von w_1 in ein Wort verwandelt, und die Verknüpfung wird mit w_1' und w_2 durchgeführt. Sollten zwei S-Wörter mit unterschiedlicher abstrakter Erzeugerliste aufeinander treffen, werden ebenfalls Kopien in Wörter verwandelt und die Verknüpfung mit diesen durchgeführt.

Pc-Wörter sind ebenso wie S-Wörter vom Typ `T_SWORD`, jedoch ist ihr erster Eintrag eine Ag-Gruppe G von Typ `T_AGGRP`. Werden sie verknüpft, wird jedoch nur der durch `HD_WORDS` gelieferte Eintrag von G verwandt. Von einem Pc-Wort w ist jedoch bekannt, daß es sich in Normalform bezüglich der Ag-Gruppe G befindet. Pc-Präsentationen enthalten nur einen Handle, welcher auf eine Ag-Gruppe zeigt.

Wird nun `ProductPcp` mit einer Pc-Präsentationen P mit Ag-Gruppe G und zwei abstrakten Wörtern w_1 und w_2 aufgerufen, so wird, falls w_1 oder w_2 nicht bereits Pc-Wörter mit Ag-Gruppe G sind, versucht Kopien von w_1 und w_2 in Pc-Wörter mit Ag-Gruppe G zu konvertieren. Danach wird der Ag-Gruppenkollektor zur Multiplikation verwandt. Das Ergebnis ist ein Pc-Wort mit Ag-Gruppe G .

In einem PQ werden die Wörter w_1 und w_2 in fast allen Fällen Pc-Wörter mit der richtigen Ag-Gruppe sein, da sie im Kontext einer Pc-Präsentation P entstanden sind. Hier entsteht also beim Rechner nur ein sehr geringer Overhead.

Sollen nun zu einer Präsentation P mit Erzeugern g_1, \dots, g_t weitere Erzeuger hinzugefügt werden, so kann einfach die Liste der Erzeuger von P erweitert werden. Dies ändert vorhandene Pc-Wörter nicht.

Sollen jedoch Erzeuger einer Präsentation P gelöscht werden, so würde dies, da die Pc-Wörter als Paare Generatornummer und Exponent abgespeichert sind, vorhandene Wörter verändern. Daher wird zum Löschen von Erzeugern die durch P gegebene Ag-Gruppe G kopiert und in dieser Kopie G' die Erzeuger gelöscht, danach wird G' in P eingetragen.

5.5 GAP Funktionen

Folgende Funktionen stehen für zur Konstruktion und Manipulation von Pc-Präsentation in GAP bereit, sie sind ausführlich in [Gap91] beschrieben.

`Pcp(str, n, p, collector)`

`Pcp` legt eine Präsentationen einer elementar abelschen p Gruppe der Ordnung p^n an.

`GeneratorsPcp(P)`

`GeneratorsPcp` liefert die Erzeuger von P als Liste von Pc-Wörtern.

`ExtendCentralPcp(P, L, p)`

`ExtendCentralPcp` erweitert die Präsentation P zentral. In L stehen die Namen der neuen Erzeuger, sie sind von der Ordnung p .

`ShrinkPcp(P, L)`

`ShrinkPcp` löscht die in L gegebenen Erzeuger aus P .

`DefinePowersPcp(P, i, x)`

`DefinePowersPcp` definiert die rechte Seite der i .ten Potenzrelation in P als x .

`DefineCommPcp(P, i, j, x)`

`DefineCommPcp` definiert die rechte Seite der Kommutatorrelation des i .ten und j .ten Erzeugers als x .

`CentralWeights(P)`

`CentralWeights` liefert die Zentralgewichte der Erzeuger von P .

`DefineCentralWeights(P, W)`

`DefineCentralWeights` definiert die Zentralgewichte der Erzeuger von P als W .

`PowerPcp(P, w, n)`

`PowerPcp` berechnet die n .te Potenz von w bezüglich P .

`SumPcp(P, w1, w2)`

`SumPcp` berechnet die Summe der Exponentenvektoren von w_1 und w_2 bezüglich P ohne einen Kollektoraufruf.

`DifferencePcp(P, w1, w2)`

`DifferencePcp` berechnet die Differenz der Exponentenvektoren von w_1 und w_2 bezüglich P ohne einen Kollektoraufruf. Mit dieser Funktionen lassen sich in einem PQ die Quotienten zweier Wörter berechnen, welche sich nur um ein Element aus dem Zentrum unterscheiden, ohne daß der Kollektor verwendet werden muß.

Die oben schon erwähnten Funktionen `CommPcp`, `ConjugatePcp` und `LeftQuotient` verknüpfen gegebene abstrakten Wörter entsprechend einer gegebenen Pc-Präsentation.

5.6 Ausblick

Obige Funktionen ermöglichen es, einen effizienten PQ in GAP zu schreiben (siehe [Nie91]). Für einen SQ müßten jedoch noch einige Erweiterungen vorgenommen werden. So unterstützen die Funktionen bisher nur die kombinatorischen Kollektoren in p -Gruppen. Für einen SQ bräuchte man jedoch einen Single-Kollektor. Desweiteren fehlen Funktionen zum Kopieren einer gegebenen Präsentation, damit man gemachte Änderungen leicht wieder verwerfen kann.

Um einen NQ in GAP zu ermöglichen, bräuchte man noch einen zusätzlichen Typ von abstrakten Worten, nämlich solche mit beliebig langen Exponenten. Desweiteren fehlt ein Kollektor für Pc-Präsentationen, in denen einige Erzeuger unendliche Ordnung haben können. Für einen solchen Kollektor kann jedoch die GAP Langzahlarithmetik verwendet werden.

Literatur

Eine genau Beschreibung der hier vorgestellten Funktionen befindet sich im Gap Handbuch (siehe [Gap91]). Die Interna von GAP sind im allgemeinen in [Sch87] beschrieben. Die Ag-Gruppen der Version 2.4 werden ausführlich in [Bis89] vorgestellt. Über den PQ-Algorithmus sowie seine Implementation in GAP wird in [Nie91] berichtet. [Weg89] beschreibt die SQ Version 2.4.

Kapitel 6

Funktionen der GAP-Bibliothek

In diesem Kapitel werden die Funktionen in GAP für die 1-Kohomologiegruppe sowie für die Konjugiertenklassen von Komplementen beschrieben. Es werden jedoch nur die internen Funktionen sowie ihre Parameterverbunde vorgestellt, eine Beschreibung der Benutzerfunktionen befindet sich zum Beispiel in [Gap91]. Dies gilt ebenso für die Ag-Gruppenfunktionen, welche nicht unmittelbar mit Komplementen oder der 1-Kohomologiegruppe zu tun haben. Einen gewissen Überblick liefert für diese Funktionen jedoch das Kapitel 7. Die Kernfunktionen, welche für einen PQ notwendig wurden, sind schon in Kapitel 5 beschrieben.

6.1 Die Einskohomologiegruppe

Alle Funktionen im Zusammenhang mit den Gruppen der 1-Koränder und 1-Kozyklen erwarten als Parameter einen Verbund, welcher die wesentlichen Informationen zur Berechnung der entsprechenden Gruppen enthält. Dieser Verbund wird von den verschiedenen Funktionen um die berechneten Informationen erweitert. Er hat im allgemeinen folgendes Format.

```

R := rec(
    group      := G,
    module     := N,
    generators  := [g1, ...],
    identity   := 1,
    relators   := [r1, ...],
    matrices   := [G1, ...],
    powerMatrices := [[G1, G1^2, ...], ...],
    sumMatrices := [[1, 1 + G1, ...], ...],
    identityMatrix := 1,
    maximalPowers := [p1, ...],
    smallGeneratingSet := [i1, ...],
    generatorsInSmall := [w1, ...],
    bigMatrices := [[C11, ...], ...],
    bigVectors := [n1, ...],
    pPrimeSet := [i'1, ...],
    complement := K,
    centralizer := C,
    oneCocycles := Z1,
    oneCobounds := B1,
    moduleMap := function( g ) ... end,
    vectorMap := function( v ) ... end,
    cocycleToComplement := function( gamma ) ... end,
    cocycleToList := function( gamma ) ... end,
    complementToCocycle := function( G ) ... end,
    listToCocycle := function( L ) ... end,
    normalIn := H,
    normalGenerators := [s1, ...],
    normalMatrices := [S1, ...],
    operations := rec( ... )
)

```

Die Einträge $R.\text{group}$, $R.\text{module}$, $R.\text{pPrimeSet}$, $R.\text{normalIn}$, $R.\text{smallGeneratingSet}$ und $R.\text{generatorsInSmall}$ sind von Benutzer zu setzen und werden von den Funktionen nicht verändert. $R.\text{smallGeneratingSet}$, $R.\text{generatorsInSmall}$ und $R.\text{pPrimeSet}$ sind optional, wobei jedoch nur entweder $R.\text{smallGeneratingSet}$ und $R.\text{generatorsInSmall}$ oder $R.\text{pPrimeSet}$ gesetzt werden dürfen.

Die Einträge $R.\text{generators}$ und $R.\text{relations}$ können auch von Benutzer gesetzt werden. Anderfalls tragen $R.\text{operations.AddGenerators}$ und $R.\text{operations.AddRelations}$ im Falle einer Ag-Gruppe $R.\text{group}$ die kanonischen Erzeuger beziehungsweise eine Potenz-Konjugierten Präsentation in $R.\text{generators}$ und $R.\text{relators}$ ein. Im Falle einer Permutationsgruppe $R.\text{group}$ trägt $R.\text{operations.AddGenerators}$ für die Erzeuger $R.\text{generators}$ die Erzeuger von $R.\text{group}$ als Repräsentanten für die Erzeuger der Faktorgruppe ein. In

diesem Fall muß der Verbund von `R.group` über weitere Einträge verfügen, siehe die Bemerkung bei `R.group`.

Auch `R.maximalPowers` darf von Benutzer gesetzt werden. Wenn der Eintrag nicht gegeben ist, wird im Falle einer Ag-Gruppe `R.group` die relative Ordnung des entsprechenden Erzeugers oder im Falle einer Permutationsgruppe `R.group` 1 verwendet.

Die Bezeichnungen entsprechen im folgenden Kapitel 2. Es sei G eine Gruppe, für die entweder eine Präsentation bekannt ist oder im Falle einer Ag-Gruppe eine kanonische Präsentation berechnet werden kann, es sei weiter N ein elementar abelscher p -Normalteiler. Die einzelnen Komponenten des Kohomologieverbundes R bedeuten dabei das Folgende.

`bigMatrices`

Ein zweidimensionales Feld, so daß der Eintrag (i, j) die Matrix C_{ij} aus der Bemerkung zu Satz 2.17 liefert für solche i , die nicht im `smallGenerationSet` liegen, und j aus dem `smallGeneratingSet`.

`bigVectors`

Eine Liste von Vektoren, welche die Elemente n_i aus der Bemerkung zu Satz 2.17 beschreiben für solche i , die nicht im `smallGenerationSet` liegen.

`centralizer`

Eine Darstellung des Zentralisators des Komplements aus `R.complement` in der Untergruppe `R.module` in GAP als Verbund.

`cocycleToComplement`

Eine Funktion, welche einen Vektor aus dem Vektorraum `R.oneCocycles` erwartet, und das zu diesem Kozykel gehörende Komplement als Verbund zurückliefert. Da der Kozykel nur in Verbindung mit einem Komplement Sinn macht, muß der Eintrag `R.complement` schon bekannt sein.

`cocycleToList`

Eine Funktion, welche einen Vektor $(\gamma(g_1), \dots)$ aus `R.oneCocycles` erwartet, und $[\gamma(g_1), \dots]$ als Teilfolge von N zurückliefert. Falls `R.smallGeneratingSet` gegeben ist, so wird die Folge nur an den g_i ausgewertet, für die i in `R.smallGeneratingSet` liegt.

`complement`

Ein Darstellung eines Komplements zu N in G in GAP als Verbund.

`complementToCocycle`

Die Umkehrfunktion zu `R.cocycleToComplement`. Dieser Eintrag ist nur im Falle einer Ag-Gruppe `R.group` vorhanden.

`generators`

Eine Liste der Erzeuger von G/N beziehungsweise Repräsentanten für diese. Die Liste

muß so angepaßt sein, daß, falls R .smallGeneratingSet oder R .pPrimeSet gegeben sind, diese die entsprechenden Erzeuger indizieren. Alle Angaben für G/N beziehen sich auf diese Erzeuger.

generatorsInSmall

Worte w_i , so daß $g_i = w_i(g_{i_1}, \dots)$ gilt. Ein Wort

$$r = \prod_{j=1}^t y_{i_j}^{e_j},$$

wird dabei als Verbund

```
rec(      generators := [i_1, i_2, ...],
         powers      := [e_1, e_2, ...],
         usedGenerators := {i_1, i_2, ...}
      )
```

dargestellt.

group

Eine Darstellung der Gruppe G in GAP als Verbund. Dies kann entweder eine Ag-Gruppe oder eine Permutationsgruppe sein.

Sollte R .group eine Permutationsgruppe sein und R .relations nicht gegeben sein, so muß das Folgende für R .group und R .module gelten. Der Verbund von R .group muß über die weiteren Einträge R .abstractGenerators und R .relators verfügen. Der Eintrag R .group.relators muß eine Präsentation liefern, so daß R .group eine Permutationdarstellung der durch diese Präsentation beschriebenen Gruppe ergibt, wenn der i -te abstrakte Erzeuger in R .group.abstractGenerators dem i -ten Permutationserzeuger in R .group.generators entspricht. Weiterhin muß auch R .module über einen Eintrag R .normalAbstractGenerators verfügen, so daß die Worte in R .module.normalAbstractGenerators ein Normalteilererzeugendensystem im Sinne der Präsentation von R .group beschreiben.

identity

Die Eins der Gruppe R .group.

identityMatrix

Die Einheitsmatrix.

listToCocycle

Die Umkehrfunktion zu R .cocycleToList.

matrices

Eine Liste von Matrizen $[G_1, \dots]$, welche die Operation von G/N auf dem Vektorraum N beschreiben.

maximalPowers

Eine Liste $[p_1, \dots]$ von Zahlen $p_i \geq 1$, welche die Obergrenze für die Summen- und Potenzmatrizen in `R.powerMatrices` und `R.sumMatrices` angeben. Es werden dabei maximal Potenzen bis $G_i^{p_i}$ und Summen bis $\sum_{k=0}^{p_i-1} G_i^k$ berechnet.

module

Eine Darstellung des elementar abelschen Normalteiler von G in GAP als Verbund. Man beachte die Bemerkung zu `R.group`.

moduleMap

Eine Funktion, welche zu einem Element aus N den entsprechenden Vektor des zu N isomorphen Vektorraumes liefert.

normalGenerators

Eine Liste von Erzeugern für `R.normalIn`.

normalIn

Falls dieser Eintrag gesetzt ist, muß `R.group` eine Gruppe sein, welche von `R.normalIn` normalisiert wird. Es werden nur solche Kozyklen gesucht, die zu Komplementen gehören, welche normal in `R.modul` und `R.normalIn` sind. Dieser Eintrag darf nur im Falle einer Ag-Gruppe `R.group` vorhanden sein.

normalMatrices

Eine Liste von Matrizen, welche die Operation von `R.normalIn` auf dem Vektorraum N beschreibt.

oneCoboundaries

Der Vektorraum $B^1(G, N)$. Siehe auch `R.smallGeneratingSet`.

oneCocycles

Der Vektorraum $Z^1(G, N)$. Siehe auch `R.smallGeneratingSet`.

operations

Ein Verbund, welcher die folgenden Hilfsfunktionen enthält, in GAP 3.1 wird dieser Eintrag für Ag-Gruppen gesetzt, falls nicht vorhanden.

AddBigMatrices

Setzt `R.bigVectors` und `R.bigMatrices`.

AddCentralizer

Setzt `R.centralizer` bei gegebenen Zentralisator.

AddComplement

Trägt gegebenes Komplement in `R.complement` ein.

AddGenerators

Setzt `R.generators`.

AddMatrices

Setzt $R.identityMatrix$, $R.moduleMap$, $R.matrices$, $R.normalMatrices$ und $R.vectorMap$

AddRelations

Setzt $R.relators$.

AddSumMatrices

Setzt $R.sumMatrices$ und $R.powerMatrices$.

AddToFunctions

Setzt alle "to" Funktionen.

NormalRelations

Liefert die Relatoren für normale Kozykel.

powerMatrices

Ein zweidimensionales Feld, so daß der Eintrag (i, j) die Matrix G_i^j liefert. Man beachte die Bemerkung bei $R.maximalPowers$.

pPrimeSet

Eine Liste von Indices $[i_1, \dots]$, so daß $\{g_{i_1}, \dots\}$ eine p' -Untergruppe von G/N erzeugt. Wenn dieser Eintrag gegeben ist, darf $R.smallGeneratingSet$ nicht gesetzt sein. Sollte $R.group$ eine Permutationgruppe sein, so darf dieser Eintrag nicht gesetzt werden.

relations

Eine Liste von Relationen, welche eine endliche Präsentation für G/N beschreiben. Dies wird im Falle einer Ag-Gruppe $R.group$ die Potenz-Konjugierten Präsentation sein, sollte jedoch eine andere bekannt sein, so kann sie vor Beginn der Berechnung hier eingetragen werden, sie darf danach jedoch nicht mehr verändert werden. Im Falle einer Permutationsgruppe $R.group$ muß, falls der Eintrag nicht gegeben ist, zumindestens für die ganze Gruppe G eine Präsentation und für den Modul N ein Normalteilererzeugendensystem bekannt sein. Man beachte hierzu die Bemerkung bei $R.group$. Ein Relator

$$r = \prod_{j=1}^t y_{i_j}^{e_j},$$

wird dabei als Verbund

```
rec(
    generators := [i1, i2, ...],
    powers    := [e1, e2, ...],
    usedGenerators := {i1, i2, ...}
)
```

dargestellt.

smallGeneratingSet

Ein kleines Erzeugendensystem entsprechend Satz 2.17 als Liste von Indices $\{i_1, \dots\}$, so daß $\{g_{i_1}, \dots\}$ ein solches Erzeugendensystem bildet. In diesem Fall werden die 1-Kozyklen aus $R.\text{oneCocycles}$ und $R.\text{oneCobounds}$ auch nur an diesen g_{i_j} ausgewertet angegeben. Wenn dieser Eintrag gegeben ist, darf $R.\text{pPrimeSet}$ nicht gesetzt sein.

sumMatrices

Ein zweidimensionales Feld, so daß der Eintrag $(i, j + 1)$ die Matrix $\sum_{k=0}^j G_i^k$ liefert. Man beachte die Bemerkung bei $R.\text{maximalPowers}$

vectorMap

Die Umkehrfunktion zu moduleMap .

Für einen Verbund R mit $R.\text{group} := G$ und $R.\text{module} = N$ berechnet die Funktion OneCocycles0C den Vektorraum der 1-Kozyklen beziehungsweise `false`, falls die Erweiterung von G/N mit N nicht zerfällt. Die Funktion OneCobounds0C berechnet den Vektorraum der 1-Koränder selbst dann, wenn die Erweiterung nicht zerfällt. In diesem Fall wird auch der Eintrag $R.\text{centralizer}$ gesetzt, welcher im nicht zerfallenden Fall der Zentralisator von G in N entspricht. Die Funktion ConjugatingWord0C liefert für einen Verbund R sowie zwei Kozykel, deren Differenz in $B^1(G, N)$ liegt, ein konjugierendes Element auf N .

6.2 Konjugiertenklassen von Komplementen

Die meisten Funktionen zur Berechnung von Konjugiertenklassen von Komplementen erwarten als einen Parameter einen Verbund R , welcher die Berechnung steuert. Er hat im allgemeinen folgendes Format.

```

rec(
    generators := [g1, ...],
    relations  := [r1, ...],
    primes     := [p1, ...],
    pPrimeSets := [[i'1, ...], ...],
    smallGeneratingSet := [i1, ...],
    generatorsInSmall := [w1, ...],
    normalSubgroup := function(< S >, < K >, < M >),
    useCentral    := true/false,
    useCentralSK  := true/false,
    normalComplements := true/false,
    elementaryAbelianSeries := [G = E1, ...]
)

```

Dabei sind alle Einträge optional, jedoch dürfen entweder *R.primes* und *R.pPrimeSets* oder *R.smallGeneratingSet* und *R.generatorsInSmall* gesetzt werden. Die Einträge *R.useCentral* und *R.useCentralSK* werden, so sie nicht vorhanden sind, mit *false* initialisiert. Die Bezeichnungen entsprechen im folgenden Kapitel 4. Es sei *G* eine endliche, polyzyklische Gruppe, in der Komplemente nach einem Normalteiler *M* gesucht werden. Es sei *N* ein elementar abelscher Normateiler in *M*, *K* ein Komplement in *G/N* und *S/N* der Normalisator von *K/N* in *M/N*. Die einzelnen Komponenten bedeuten dabei folgendes.

elementaryAbelianSeries

Soll eine andere elementar abelsche Reihe verwendet werden, als die, welche man durch 'ElementaryAbelianSeries' erhält, so muß diese hier eingetragen werden.

generators

Ein Liste der Erzeuger von *K/N*. Diese Liste ist so angepaßt, daß, falls *R.pPrimeSets* oder *R.smallGeneratingSet* gegeben sind, diese die entsprechenden Erzeuger indizieren.

generatorsInSmall

Worte *w_i*, welche die Erzeuger *R.generators* in dem kleinen Erzeugendensystem *R.smallGeneratingSet* ausdrücken entsprechend Abschnitt 6.1.

normalComplements

Falls *R.normalComplements* wahr ist, werden nur normale Komplemente gesucht.

normalSubgroup

Eine Funktion, welche einen Normalteiler von *< S, K >* liefert, der im Schnitt aller Komplemente zu *N* in *K* liegt. Ist dieser Normalteiler *L* nicht-trivial, werden die Einträge *R.pPrimeSets*, *R.smallGeneratingSet*, *R.relations* und *R.generators* ignoriert und die Komplemente zuerst in *K/L* berechnet.

pPrimeSets

Eine Liste von Typ *R.pPrimeSet* aus Abschnitt 6.1.

primes

Eine Liste von Primzahlen p_i , für die eine p_i Untergruppe von K/N bekannt ist. In diesem Fall liefert der i .te Eintrag von $R.pPrimeSets$ eine solche Untergruppe.

relations

Eine Relationensystem für **generators** entsprechend Abschnitt 6.1.

smallGeneratingSet

Ein kleines Erzeugendensystem entsprechend Abschnitt 6.1. Falls dieses gegeben ist, muß auch $R.generatorsInSmall$ bekannt sein.

useCentral Falls N ein zentraler Normalteiler in G ist, werden die Komplemente durch lineare Methoden entsprechend Kapitel 4 berechnet.

useCentralSK Falls N ein zentraler Normalteiler in $\langle S, K \rangle$ ist, werden die Komplemente durch lineare Methoden entsprechend Kapitel 4 berechnet.

Die Funktion **ComplementsC0** erwartet als Parameter einen solchen Verbund R sowie eine Ag-Gruppe G und eine Zahl n , die angibt, zu welcher Untergruppe der elementar abelschen Reihe die Komplemente berechnet werden sollen.

Kapitel 7

Zeiten

In den folgenden Abschnitten sind Zeitvergleiche zwischen den Systemen GAP, SOGOS und CAYLEY aufgelistet, soweit die entsprechenden Funktionen zur Verfügung standen. SOGOS lag dabei in der Version 5.0 auf einem MASSCOMP 5500 PEP vor, GAP stand auf dem MASSCOMP und einer DecStation 5000/120 in der Version 3.1 zur Verfügung, CAYLEY lief in der Version 3.6-416 auf dem MASSCOMP. In den Tabellen bezeichnen dabei die Abkürzungen G1 bis G6 die folgenden Gruppen.

G1 Die Gruppe $Z_5wr_{31}(Z_{31} \rtimes Z_5)$ der Ordnung $5^{32} \cdot 31$.

G2 Die Gruppe $(S_4wr_4S_4)wr_4S_4$ der Ordnung $2^{63} \cdot 3^{21}$.

G3 Die Gruppe $Aut(F) \cdot F^* \cdot F^+$ für den Körper $F = GF(3^{10})$ der Ordnung $2^4 \cdot 3^{10} \cdot 5 \cdot 11^2 \cdot 61$.

G4 Die Boreluntergruppe $B(2, 8) \leq Gl(2, 8)$ der Ordnung $2^{18} \cdot 7^4$.

G5 Die Gruppe $Aut(F) \cdot F^* \cdot F^+$ für den Körper $F = GF(2^{11})$ der Ordnung $2^{11} \cdot 11 \cdot 23 \cdot 89$.

G6 Die Gruppe $H \cdot 7^{2+1} \cdot 13^{14+1}$, wobei H eine nicht zerfallende Doppelüberdeckung der S_4 ist und 7^{2+1} und 13^{14+1} extraspezielle Gruppe von Primezahlexponent sind. Die Gruppe $G6$ hat die Ordnung $2^4 \cdot 3 \cdot 7^3 \cdot 13^{15}$.

Desweiteren bezeichnet $C(n)$ die n .te Kompositionsuntergruppe von G und $H(n)$ eine Hall- n -Untergruppe einer Gruppe. Eine Zahl bei der Angabe einer Untergruppe bezeichnet eine zufällige Untergruppe dieser Größe.

Alle Zeitangaben sind, soweit nicht anders angegeben, in Millisekunden gemessen auf dem MASSCOMP.

7.1 Kanonische Erzeugendesysteme

Sowohl in SOGOS als auch in GAP wird ein nicht-kommutativer Gaußalgorithmus zur Berechnung eines kanonischen Erzeugendensystems für eine durch ein Erzeugendensystem gegebene Gruppe verwendet. In GAP wird bei der Kommutatoren- und Potenzbildung auf die möglichen Tiefen geachtet.

Gruppe	GAP single	GAP quadruple	Cayley	Sogos
G6 ¹	>3000	68,633	>3000	733,700
G6 ²	>3000	60,733	148,476	638,650
G4 ³	19,433	19,133	42,783	13,000

¹ 10 zufällig bestimmte Untergruppen der Gruppe G6 mit einem Erzeugendensystem mit 1 bis 4 Erzeugern. Die Ordnungen der Untergruppen sind dabei $2 \cdot 3 \cdot 7 \cdot 13$, $2^2 \cdot 3 \cdot 7^3 \cdot 13^{15}$, $2^2 \cdot 7 \cdot 13^{15}$, $2^3 \cdot 7^3 \cdot 13^{15}$, $3 \cdot 7 \cdot 13$, $2 \cdot 3 \cdot 7 \cdot 13$, $2^2 \cdot 7 \cdot 13$, $2^2 \cdot 3 \cdot 7^3 \cdot 13^{15}$ und $2^3 \cdot 13$.

² Wie ¹, jedoch nur die ersten 5 Untergruppen

³ 10 zufällig bestimmte Untergruppen der Gruppe G4 mit einem Erzeugendensystem mit 1 bis 4 Erzeugern. Die Ordnungen sind dabei $2^{18} \cdot 7^3$, $2^{21} \cdot 7^2$, 7 , $2^{18} \cdot 7^2$, $2^{18} \cdot 7^3$, $2^{18} \cdot 7^3$, $2^2 \cdot 7$, $2^{18} \cdot 7^3$, $2^{17} \cdot 7^3$ und $2 \cdot 7$.

7.2 Normale Hülle

Bei der Berechnung der subnormalen Reihe einer Untergruppe U in einer Gruppe G wird fast ausschließlich nacheinander die normale Hülle gebildet. Dabei macht sich in GAP die im vorhergehenden Abschnitt erwähnte Beachtung der überhaupt möglichen Tiefen bemerkbar.

Gruppe	Untergruppe	Gap	Cayley	Sogos	Länge
G2	$\langle g_1 \rangle$	40,033	647,516	279,580	2
G2	48	317,950	>2000	858,670	5
G3	$\langle g_1 \rangle$	8,716	185,949	14,600	5
G4	$\langle g_2 g_4 g_8 g_{20} \rangle$	2,683	11,433	4,150	2
G5	$\langle g_4 g_5 \rangle$	1,350	2,599	1,200	3
G6	$\langle g_4 \rangle$	15,284	107,783	20,500	3

Für die Gruppen G3 und G5 wurde in GAP ein "quadruple"-Kollektor angelegt.

7.3 Konjugierte Untergruppen

Die konjugierten Untergruppen werden in GAP nicht mehr durch einen nicht-kommutativen Gaußalgorithmus berechnet, sondern nach Lemma 1.14. Dies macht sich insbesondere bei großen Orbits durch einen deutlichen Zeitvorteil bemerkbar.

Im folgenden wurden für eine Untergruppe U einer Gruppe G alle zu U konjugierten Untergruppen in G berechnet.

Gruppe	Untergruppe	Gap	Sogos	Orbit
G2	$C(10)$	38,100	68,200	3
G3	$H(3 \cdot 5 \cdot 11)$	18,184	52,430	61
G5	$H(11 \cdot 23 \cdot 89)$	49,950	319,180	2048
G6	$H(2 \cdot 13)$	2497,516	> 4000,000	147

Für die Gruppen G3 und G5 wurde in GAP ein “quadruple”-Kollektor angelegt. In CAYLEY lassen sich leider nicht ohne weiteres nur die Konjugierten einer Untergruppe berechnen.

7.4 Schnitte von Untergruppen

GAP verwendet zur Schnittbildung den Algorithmus von Glasby und Slattery (siehe [?]), während SOGOS Glasbys Algorithmus (siehe [GS92]) benutzt. In der zur Verfügung stehenden CAYLEY Version ist jedoch noch keiner der beiden enthalten. Unter Umständen, das heißt, falls eine der Untergruppen subnormal ist, läßt sich der Schnitt jedoch auch ohne die Gefahr eines zu großen Orbits direkt ausrechnen.

Gruppe	U	V	$U \cap V$	GAP quadruple	Cayley	Sogos
G6	$H(2 \cdot 3 \cdot 7)$	U^{99g10}	21	20,170	311,982	> 2000
G5	$H(11 \cdot 23 \cdot 89)$	$U^{95 \dots 910}$	11	7,134	29,148	291,480
G4 ¹	$H(7)$	$U^{95 \dots 910}$	49	5,117	— ⁴	— ⁵
G4 ²	$H(7)$	$U^{95 \dots 910}$	49	3,150	— ⁴	1,980
G4 ²	$H(7)$	U^{95g7}	343	3,167	— ⁴	0,770
G4 ²	$H(7)$	U^{95g9}	49	3,433	— ⁴	2,530
G4 ²	$H(7)$	$U^{98g20g21}$	49	3,217	— ⁴	3,380
G4 ²	$H(7)$	U^{910g11}	49	3,166	— ⁴	3,630
G4 ²	$H(7)$	U^{96g20}	49	3,300	— ⁴	4,090
G4 ²	$H(7)$	$U^{95g10g11}$	7	3,433	— ⁴	27,220
G4 ²	$H(7)$	$U^{98g10g11g7}$	7	3,316	— ⁴	31,470
G4 ^{2,3}	siehe unten	siehe unten	s.u.	6,683	— ⁴	4,500

¹ elementar abelsche Reihe in GAP: $7, 7, 7, 7, 2^3, 2^3, 2^3, 2^3, 2^3, 2^3$

² elementar abelsche Reihe in GAP: $7^4, 2^9, 2^9$

³ 5 Schnitte mit je zwei zufälligen Untergruppen von G4. Die Ordnungen der Schnitte sind $2^{18} \cdot 7^2, 2, 1, 2^{17} \cdot 7$ und 2.

⁴ Orbit zu groß für direkte Methode.

⁵ SOGOS wählt automatisch die elementar abelsche Reihe $7^4, 2^9, 2^9$.

Der Schnittalgorithmus von Glasby und Slattery approximiert die gesuchte Untergruppe durch Wahl von bestimmten Urbildern und konjugierten Untergruppen im Gegensatz zu dem Normalisatoralgorithmus oder dem Zentralisatoralgorithmus, welche immer vollständige Urbilder wählen können. Die Orbitalgorithmen sind im Schnittalgorithmus linear möglich, so daß es hier sinnvoller ist, nicht in die Faktorgruppen überzugehen. Dies kann zwar zu leicht längeren Laufzeiten führen, zahlt sich aber bei größeren Gruppen doch deutlich aus.

Glasbys Algorithmus verwendet einen gewöhnlichen Orbitalgorithmus anstelle eines affinen für den Fall, daß ein Orbitalgorithmus durchgeführt werden muß. Falls ein exzentrischer Hauptfaktor gemieden wird oder ein beliebiger Hauptfaktor gedeckt wird, kann auch hier auf einen Orbitalgorithmus verzichtet werden. Da jedoch der Orbitalgorithmus nicht affin durchgeführt werden kann, führt dies bei längeren Orbits zu deutlich längeren Laufzeiten.

7.5 Kommutatoruntergruppen

Gap verwendet Glasbys Algorithmus zur Berechnung der Kommutatoruntergruppe, so daß auf die Bildung der normalen Hülle unter Umständen verzichtet werden kann. Im folgenden sind die Zeiten zur Berechnung der Kommutatorreihe aufgelistet, welche in GAP und SOGOS jeweils durch iteriertes Berechnen der Kommutatoruntergruppe berechnen werden.

Gruppe	Gap	Cayley	Sogos	Länge
G2	67,383	666,316	602,900	10
G3	1,467	44,283	2,220	3
G5	0,933	11,300	0,950	4
G6	5,900	127,766	25,200	9

In der folgenden Tabelle sind die Zeiten zur Berechnung der unteren Zentralreihe einer Halluntergruppe angegeben.

Gruppe	Untergruppe	Gap	Cayley	Sogos
G2	H(2)	185,467	244,699	1324,500
G4	H(2)	1,012	1,500	1,700
G6	H(2)	0,250	0,200	1,100
G6	H(7*13)	2,266	16,133	9,930

Hier kann zwar Glasbys Algorithmus nicht verwendet werden, jedoch macht sich der Algorithmus zur Berechnung der normale Hülle bemerkbar.

7.6 Halluntergruppen

Alle drei Programme verwenden Glasbys Algorithmus zur Berechnung der Halluntergruppen. Jedoch wird in GAP für große Primzahlen anstelle Glasbys spezieller Lösung ein lineares Gleichungssystem gelöst.

Gruppe	Hall	GAP	Cayley	Sogos
G6	2*3*7	2,050	4,065	3,180
G5	11*23*89	0,517	1,100	0,980
G4 ¹	7	2,400	— ⁴	— ⁴
G4 ²	7	0,950	1,816	0,470
G6 ³	2*7	4,003	>5000	— ⁵

Für die Gruppen G6 wurde in GAP ein “quadruple”-Kollektor angelegt.

¹ elementar abelsche Reihe in GAP: 7, 7, 7, 7, 2³, 2³, 2³, 2³, 2³, 2³

² elementar abelsche Reihe in GAP: 7⁴, 2⁹, 2⁹

³ Hall-{2, 7}-Untegruppe von 5 zufällig bestimmten Untergruppen.

⁴ SOGOS and CAYLEY wählen automatisch die elementar abelsche Reihe 7⁴, 2⁹, 2⁹.

⁵ SOGOS erlaubt nicht die Bildung von Hallgruppen von Untergruppen.

In folgender Tabelle sind noch einige Vergleiche bezüglich linearer Methoden beziehungsweise spezieller Lösung aus Abschnitt 3.5 aufgeführt, es wurden konjugierende Elemente für zwei Halluntergruppen gesucht und mithilfe der beiden Lösungsansätze berechnet. Alle Zeiten sind in Millisekunden gemessen auf einer DecStation 5000/120.

Gruppe	Hall	Methode	Gap Single	Gap Quadruple
G5 × G5	11*23*89	speziell	5,300	0,782
		linear	1,074	0,972
G3 × G3	5*11*61	speziell	133,241	5,441
		linear	22,897	3,515

Im allgemeinen lohnt es sich also bei größeren Primzahlen, zu linearen Methoden überzugehen, insbesondere wenn nur ein “single”-Kollektor installiert ist,

7.7 Normalisator

Da SOGOS den Normalisator mittels eines gewöhnlichen Orbitalgorithmus ausrechnet, ist im folgenden nur ein Vergleich mit CAYLEY angegeben. Leider stand die neuste CAYLEY Version nicht auf dem MASSCOMP zur Verfügung. Die angegebenen Werte für CAYLEY sind in Millisekunden und wurden auf einer APOLLO 10000 ermittelt. Die Wert für GAP sind ebenfalls in Millisekunden allerdings gemessen auf einer DecStation 5000/120. Beide Rechner sind ungefähr gleich schnell.

Es wurde folgende Serie von Gruppen benutzt. Es seien p und q Primzahlen, $n \in \mathbb{N}$ und $p|n|q-1$. Weiter seien Z_p, Z_q und $Z_n = \langle a \rangle$ die zyklischen Gruppen der Ordnung p ,

q beziehungsweise n , $Z_q \rtimes Z_n$ sei das nicht-abelsche, semidirekte Produkt, dargestellt als Permutationsgruppe vom Grad q and $G = Z_p \text{wr}_p(Z_q \rtimes Z_n)$.

Im folgenden sind für einige Werte p , q , n und Untergruppen $U < \langle a \rangle$ die maximalen Ordnungen der auftretenden Gruppen der Einkoränder, die Ordnung des Normalisators sowie die Laufzeiten angegeben.

p	q	n	$ G $	U	$N_G(U)$	$ B^1 $	Gap	Cayley
2	31	30	$2^{32} \cdot 3 \cdot 5 \cdot 31$	$\langle a^{15} \rangle$	$2^{17} \cdot 3 \cdot 5$	2^{15}	1,660	103,000
3	13	12	$3^{14} \cdot 2^2 \cdot 13$	$\langle a^4 \rangle$	$2^2 \cdot 3^6$	3^8	0,535	10,000
5	11	10	$5^{12} \cdot 2 \cdot 11$	$\langle a^2 \rangle$	5^4	5^8	0,359	> 400,000

Literaturverzeichnis

- [Bis89] Thomas Bishops. *Collectoren im Programmsystem GAP*. Diplomarbeit, Juni 1989.
- [CNW90] F. Celler, J. Neubüser, and C.R.B. Wright. *Some remarks on the computation of complements and normalizers in soluble groups*. In G.M.Piacentini Cattaneo and E.Strickland, editors, *Topics in computational algebra*, pages 57–76, 1990.
- [Gam90] Alessandra Gambini. *A polynomial algorithm for computing Praefrattini subgroups*. In *Grouptheory, Proc. 2nd Int. Conf., Bressanon/Italy 1989*, pages 73–76. *Suppl. Rend. Circ. Mat. Palermo, II. Ser.* 23, 1990.
- [Gap91] *Gap Manual*, 1991.
- [Gla87] Stephen P. Glasby. *Computational approaches to the theory of finite soluble groups*. PhD thesis, University of Sydney, 1987.
- [GS92] Stephen P. Glasby and Michael C. Slattery. *Computing Intersections and Normalizers in Soluble Groups*. To appear in *Journal of Symbolic Computation*, 1992.
- [Hup67] B. Huppert. *Endliche Gruppen I*. Springer, Berlin, 1967.
- [KT88] W.M. Kantor and D.E. Taylor. *Polynomial-time version of Sylow's theorems*. *Journal of Symbolic Computation*, 9:1–17, 1988.
- [LNS84] R. Laue, J. Neubüser, and U. Schoenwaelder. *Algorithms for finite soluble groups and the SOGOS system*. In M.D.Atkinson, editor, *Proceedings on Computational Group Theory, Durham 1982*, pages 105–135. LMS Symposium, Academic Press, 1984.
- [MN89] M. Mecky and J. Neubüser. *Some remarks on the computation of conjugacy classes of soluble groups*. *Bulletin of the Australian Mathematical Society*, 40(2):281–292, October 1989.

- [Nie91] Alice Niemeyer. Private communications, 1991.
- [Sch87] Martin Schönert. *Konzeption und Implementation des Programmiersystems GAP für die algorithmische Gruppentheorie*. Diplomarbeit, März 1987.
- [Weg89] Alex Wegner. *Implementation eines Algorithmus zur Berechnung endlicher auflösbarer Faktorgruppen einer endlich präsentierten Gruppe im gruppentheoretischen Programmsystem GAP*. Diplomarbeit, Februar 1989.
- [Wri88] C.R.B. Wright. *Recursive Algorithms for Computing Complements in Finite Groups*. 1988.
- [Wri90] C.R.B. Wright. Private communications, 1990.