Algorithms for Polycyclic Groups

Habilitationsschrift

Eingereicht beim
Fachbereich für Mathematik und Informatik
Universität Gesamthochschule Kassel

von
Bettina Eick
aus Bremervörde

Kassel, im November 2000

Preface

A group is called *polycyclic* if there exists a *polycyclic series* through the group; that is, a subnormal series of finite length with cyclic factors. Each polycyclic group has a normal series of finite length whose factors are finitely generated abelian groups. This can be considered as a finiteness condition on polycyclic groups which makes them accessible for algorithmic purposes and it is the aim of this book to develop a variety of algorithmic methods for polycyclic groups.

The structural investigation of polycyclic groups has been initiated 1938 by Hirsch in [29, 30, 31, 32, 33] and their central position in infinite group theory has long been recognized since. Polycyclic groups form a large class of groups; in particular, each finitely generated nilpotent group and each supersolvable group is polycyclic. Vice versa, each polycyclic group is finitely generated and solvable. In fact, polycyclic groups can be characterized as those solvable groups in which each subgroup is finitely generated.

The algorithmic decidability of group theoretic questions in polycyclic groups has been considered by Baumslag, Cannonito, Robinson & Segal in [3] and by Segal in [69]. They proved that a large variety of interesting problems is decidable. The algorithms introduced for this purpose are algorithms in the classical sense and have not been invented for computer implementations. But clearly they can be considered as the fundamental initial step towards an algorithmic theory for polycyclic groups.

Algorithms for polycyclic groups which are suitable for implementations and practical for interesting computations are the main issue here. The central goal for each of the considered problems is to reduce it either to applications of well-known methods from linear algebra or number theory or otherwise to solve it by methods which have proved to be practical by implementations. A number of the algorithms described here have been implemented in the computer algebra system GAP [74] and a part of these programs is available in the 'Polycyclic' share package [23]. These implementations have been used to exploit the practicality of their underlying methods.

This book is intended to give an overview on algorithms for polycyclic groups with a particular interest in infinite polycyclic groups. It includes methods which have been known for some while and it also presents a variety of new developments in this area. In the following we give a brief overview on the 11 chapters of this book.

1-3 We introduce constructive polycyclic sequences which form the fundamental basis of our algorithms for polycyclic groups. We show that they are closely related to polycyclic presentations and we exploit their applications in handling subgroups, factor groups and homomorphisms of polycyclic groups. Most of the topics in these chapters are well-known in principle and another approach to them can be found in [71].

4-5 We describe algorithms to compute constructive polycyclic sequences for polycyclic permutation groups and polycyclic matrix groups over finite fields or the rational numbers. Permutation groups and matrix groups over finite fields are finite and thus comparatively easy to handle. Our methods for these finite groups are based on an algorithm of Sims [70]. Rational matrix groups can also be infinite and we need a more detailed investigation in this case. Our approaches to rational matrix groups are based on algorithms by Dixon [16] and Ostheimer [59].

6 Cohomology groups can be used to construct complements and extensions. We outline methods to compute cohomology groups for polycyclic groups which generalize the methods for finite polycyclic groups by Celler, Neubüser & Wright [9] and by Plesken and Brückner in [62, 7]. Then we exploit the cohomology groups for infinite polycyclic groups. In particular, we present an algorithm for determining an almost complement in an infinite polycyclic group.

7 The determination of orbits and stabilizers is one of the most fundamental problems in algorithmic group theory. Methods to determine finite orbits are well-known, but the development of algorithms to solve orbit stabilizer problems for infinite polycyclic groups acting as groups of automorphisms on finitely generated abelian groups can be considered as one of the major steps towards an algorithmic theory of infinite polycyclic groups. We present methods for this purpose in this chapter. A part of these methods has been developed in joint work with Ostheimer [25].

8-9 We present a variety of algorithms designed to investigate the structure of polycyclic groups given by constructive polycyclic sequences. Chapter 8 contains algorithms for more general questions on subgroups of polycyclic groups such as the determination of centralizers, normalizers and intersections. In Chapter 9 we consider more specific problems. In particular, we introduce a variety of algorithms to exhibit a number of group theoretic structure theorems for infinite polycyclic groups.

10 The algorithms in this book are often described as 'practical' or 'effective'; that is, they are suitable for implementations and they are expected to have interesting applications. In this chapter we describe some examples of such applications and we use these examples to include a report on the performance of some of our methods.

11 This book provides a basis for the investigation of polycyclic groups by computational methods. However, for a number of interesting questions on polycyclic groups there are no practical algorithms known yet and some questions are not even known to be decidable in infinite polycyclic groups. We close this book with a collection of open problems of this type.

The algorithms presented here all apply to polycyclic groups in general, but the main focus in this book is towards infinite polycyclic groups. In fact, most of the new developments described in this book are algorithms for infinite polycyclic groups. The computational theory of finite polycyclic groups has been developed over the last 30 years and a large number of practical algorithms for such groups are known. Implementations of them are available in the computer algebra systems GAP [74] and MAGMA [5]. In the course of the book we include references to finite group methods corresponding to our algorithms.
The introduced algorithms make use of a variety of group theoretic properties of polycyclic groups. A number of them are recalled briefly when they are used. For more background on the structure of polycyclic groups we recommend the introduction in [67], pages 147ff, and the book by Segal [68].

Acknowledgments

Many thanks are due to Gretchen Ostheimer who introduced me to the interesting field of polycyclic integral matrix groups and her work on algorithms for these groups. Further, I thank Werner Nickel who took part in implementing a basic machinery for polycyclic groups in the 'Polycyclic' share package of GAP. The GAP system has provided a useful framework for the implementation of algorithms for polycyclic groups and I acknowledge the support of the GAP team in this project. Finally, I thank Hans Ulrich Besche, Alexander Hulpke, Gunter Malle and Eamonn O'Brien for reading and commenting this work.

Contents

Notation

$G, H, \ldots$	groups, rings, modules, etc.
$\mathcal{G}, \mathcal{H}, \ldots$	sequences
RG	group ring of a group G over a ring R
$R(G)$	matrix algebra of $G \leq GL(d, R)$ over a ring R
$\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	sets of integers, rational, real and complex numbers
$H \cong G$	H is isomorphic to G
$H \leq G$ $(H < G)$	H is a (proper) subgroup of G
$H \trianglelefteq G$ $(H \lhd G)$	H is a (proper) normal subgroup of G
G^n	$\langle g^n \mid g \in G \rangle \leq G$
$\|G\|$	order of a group G
$[G : H]$	index of the subgroup H in G
$C_G(H)$, $N_G(H)$	centralizer and normalizer of H in G
$Stab_G(x)$	stabilizer of x in G
H^G	normal closure of H in G
$H \times G$	direct product of H and G
$H \otimes G$	tensor product of H and G
G'	derived subgroup of G
$G^{(i)}$	terms of the derived series of G
G_i	terms of a polycyclic series or an abelian normal series of G
$\lambda_i(G)$, $\gamma_i(G)$	terms of the lower and upper central series of G
$\mathcal{F}_i(G)$	terms of the upper nilpotent series of G
$Z(G)$	centre of G
$FC(G)$	FC-centre of G
$Fit(G)$	Fitting subgroup of G
$\Phi(G)$, $\Phi_p(G)$	Frattini and p-Frattini subgroup of G
$Aut(G)$	automorphism group of G
$T(G)$, $TN(G)$	torsion and normal torsion subgroup of G
$exp(G)$	exponent of the group G
$hl(G)$	Hirsch length of the group G
$K_p(G)$, $I_p(G)$	p-congruence subgroup and p-modular image of G
$H^i(G, M)$, $Z^i(G, M)$, $B^i(G, M)$	cohomology groups of a group G with a module M
S_n	symmetric group on n symbols
$GL(n, \mathbb{R})$, $U(n, \mathbb{R})$	general and unitriangular linear group
W^*	dual lattice to W
$Rad_G(V)$	radical of G-module V

$g, h, \ldots$	elements of a set
$\alpha, \beta, \ldots$	functions
$im(\alpha)$, $ker(\alpha)$	image and kernel of α
$x\alpha$ or x^α	image of x under α
x^y	$y^{-1}xy$
$[x, y]$	$x^{-1}y^{-1}xy$
$\exp_{\mathcal{G}}(g)$	exponent vector of g wrt $\mathcal{G}$
$\mathrm{dep}_{\mathcal{G}}(g)$	depth of g wrt $\mathcal{G}$
$\mathrm{lead}_{\mathcal{G}}(g)$	leading exponent of g wrt $\mathcal{G}$
$\mathrm{relord}_{\mathcal{G}}(g)$	relative order of g wrt $\mathcal{G}$
$\mathrm{relind}_{\mathcal{G}}(g)$	relative index of g wrt $\mathcal{G}$

Chapter 1

Introduction to polycyclic sequences

A group is polycyclic if it has a polycyclic series; that is, a subnormal series of finite length with cyclic factors. The algorithmic theory for polycyclic groups developed here uses this polycyclic structure as the fundamental tool to compute with polycyclic groups. To access this polycyclic structure in a computationally useful form we introduce polycyclic sequences. These are generating sets for polycyclic groups which are adapted to a polycyclic series of their underlying group. In this chapter we develop the basic properties of these sequences and thus provide the framework for the algorithms in later chapters.

References: Polycyclic sequences have been known under various names in the past. For example, Laue, Neubüser & Schoenwaelder [38] called them 'AG-systems' and used them for algorithms in finite polycyclic groups. Another often used name is 'polycyclic generating sequences' which, for example, had also been chosen by Sims in [71]. Here we use the term 'polycyclic sequences' as an abbreviation.

1.1 Polycyclic sequences

Let G be a group and consider a sequence $\mathcal{G} = (g_1, \ldots, g_n)$ of its elements. Each such sequence determines a series of subgroups in G defined by $G_{n+1} = 1$ and $G_i = \langle g_i, G_{i+1} \rangle$ for $1 \leq i \leq n$. If the series of subgroups determined by the sequence $\mathcal{G}$ is subnormal, then it is a polycyclic series for G_1. If $\mathcal{G}$ generates G and determines a polycyclic series, then we call $\mathcal{G}$ a *polycyclic sequence* for G.
Hence a polycyclic sequence is a generating set which exhibits the polycyclic structure of its underlying group. The following lemma notes that polycyclic sequences and polycyclic groups are connected to each other.

1.1. Lemma: *A group is polycyclic if and only if it has a polycyclic sequence.*

Proof. Suppose that G is a polycyclic group and let $G = G_1 \trianglerighteq \ldots \trianglerighteq G_n \trianglerighteq G_{n+1} = 1$ be a polycyclic series for G. Choose elements $g_i \in G$ with $G_i = \langle g_i, G_{i+1} \rangle$. Then the resulting sequence $(g_1, \ldots, g_n)$ is a polycyclic sequence for G. The converse is obvious. □

In addition to the above lemma it might be useful to observe that a polycyclic group may have several different polycyclic series and each polycyclic series may have several different polycyclic sequences determining it. Hence polycyclic sequences are by no means unique.
Let $\mathcal{G}$ be a polycyclic sequence of G which determines the series $G_1 \geq \ldots \geq G_{n+1}$. The corresponding indices $r_i = [G_i : G_{i+1}]$ are called the *relative orders* of $\mathcal{G}$. Thus each index r_i is either a positive integer or $r_i = \infty$. We denote $I = \{i \in \{1, \ldots, n\} \mid r_i < \infty\}$ as the *finite index set* for $\mathcal{G}$. The relative orders and their finite index set exhibit some properties of G as we note in the following remark.

1.2. *Remark:* Let G be a polycyclic group with polycyclic sequence $\mathcal{G}$. Let $(r_1, \ldots, r_n)$ be the relative orders of $\mathcal{G}$ and I its finite index set.

- The order of G can be read off from $\mathcal{G}$ as $|G| = r_1 \cdots r_n$. In particular, G is infinite if and only if at least one of the relative orders is infinite; that is, if $|I| < n$.
- The *Hirsch length* of a polycyclic group G is defined as the number of infinite factors in a polycyclic series of G. It is straightforward to observe that this is an invariant of the group G independent of the chosen series. We can read off the Hirsch length $hl(G)$ of G from $\mathcal{G}$ as $hl(G) = n - |I|$.
- If there exists a trivial entry $r_i = 1$ in the sequence of relative orders, then we can remove the corresponding element g_i from $\mathcal{G}$ without loss of information.
- Let $\mathcal{G} = (g_1, \ldots, g_n)$ and denote $\mathcal{G}_i = (g_i, \ldots, g_n)$. For inductive purposes it is useful to observe that $\mathcal{G}_i$ is a polycyclic sequence for $G_i = \langle \mathcal{G}_i \rangle$.

1.2 Normal forms and exponent vectors

In this section we introduce a fundamental property of polycyclic sequences: they allow to define a normal form for elements of their underlying group. This feature will form the basis for the algorithms developed later.

1.3. Lemma: *Let $\mathcal{G} = (g_1, \ldots, g_n)$ be a polycyclic sequence of G with relative orders $r_1, \ldots, r_n$ and finite index set I. Then each element $g \in G$ has a unique* normal form $g = g_1^{e_1} \cdots g_n^{e_n}$ *with $e_1, \ldots, e_n \in \mathbb{Z}$ and $0 \leq e_i < r_i$ for $i \in I$.*

Proof. Let $g \in G$. Since G_1/G_2 is generated by $g_1 G_2$, we obtain that $gG_2 = (g_1 G_2)^{e_1} = g_1^{e_1} G_2$ for some $e_1 \in \mathbb{Z}$. If $r_1 = \infty$, then e_1 is uniquely defined. If r_1 is finite, then restricting the choice of e_1 to the set $\{0, \ldots, r_1 - 1\}$ yields a unique exponent e_1. Thus we obtain a unique expression $g = g_1^{e_1} \cdot h$ for some element $h \in G_2$. Now we apply this approach recursively to h using the polycyclic sequence $\mathcal{G}_2$ of G_2. □

Let G be a polycyclic group with polycyclic sequence $\mathcal{G}$ and let $g \in G$. If $g = g_1^{e_1} \cdots g_n^{e_n}$ is the normal form of g with respect to $\mathcal{G}$, then we call the sequence $(e_1, \ldots, e_n)$ the *exponent vector* of g with respect to $\mathcal{G}$. Thus the determination of an exponent vector is equivalent to the computation of a normal form. Many of the algorithms for polycyclic groups introduced in this book are based on polycyclic sequences which allow the algorithmic determination of exponent vectors.

If G is a finite polycyclic group and if we can compare elements in G, then we can determine exponent vectors: for example, we could use Lemma 1.3 to list all normal forms of elements in G and then we could compute the exponent vector of a given element in G by looking up the element in the list. This might not be an practical approach, but it shows that it is possible to construct exponent vectors. However, if G is infinite, then the algorithmic determination of exponent vectors may not be obvious.
We call $\mathcal{G}$ a *constructive polycyclic sequence* if it admits the algorithmic construction of exponent vectors. Additionally, we usually assume that there exists an *effective* method to compute exponent vectors with respect to $\mathcal{G}$, since this will be one of the fundamental tools for later algorithms. The method used to determine exponent vectors with respect to $\mathcal{G}$ will depend on the representation in which G is given; for example, we will use different methods for permutation groups, matrix groups or finitely presented groups G. More information on this topic appears in Section 1.3 below.

A constructive polycyclic sequence $\mathcal{G}$ allows us to access more detailed information on an element g which we summarize together with the used notation in the following list:

- *Exponent vector:* $\exp_{\mathcal{G}}(g) = (e_1, \ldots, e_n)$.
- *Depth:* $\operatorname{dep}_{\mathcal{G}}(g) = d$ for the smallest d with $e_d \neq 0$ and $\operatorname{dep}_{\mathcal{G}}(1) = n + 1$.
- *Leading exponent:* $\operatorname{lead}_{\mathcal{G}}(g) = e_d$ where $\operatorname{dep}_{\mathcal{G}}(g) = d$.
- *Relative order:* $\operatorname{relord}_{\mathcal{G}}(g) = [\langle g, G_{d+1}\rangle : G_{d+1}]$ where $\operatorname{dep}_{\mathcal{G}}(g) = d$.
- *Relative index:* $\operatorname{relind}_{\mathcal{G}}(g) = [G_d : \langle g, G_{d+1}\rangle]$ where $\operatorname{dep}_{\mathcal{G}}(g) = d$.

The depth and the leading exponent can obviously be read off from the exponent vector. In the following lemma we note that also the relative order and the relative index can be determined from the exponent vector using elementary integer operations.

1.4. Lemma: *Let $\mathcal{G}$ be a polycyclic sequence for G with relative orders $(r_1, \ldots, r_n)$ and consider an element $g \in G$ with $\operatorname{dep}_{\mathcal{G}}(g) = d$.*

- *If $r_d = \infty$, then $\operatorname{relind}_{\mathcal{G}}(g) = |\operatorname{lead}_{\mathcal{G}}(g)|$ and $\operatorname{relord}_{\mathcal{G}}(g) = \infty$.*
- *If $r_d < \infty$, then $\operatorname{relind}_{\mathcal{G}}(g) = |\gcd(r_d, \operatorname{lead}_{\mathcal{G}}(g))|$ and $\operatorname{relord}_{\mathcal{G}}(g) = r_d/\operatorname{relind}_{\mathcal{G}}(g)$.*
- *$r_d = \operatorname{relind}_{\mathcal{G}}(g) \cdot \operatorname{relord}_{\mathcal{G}}(g)$.*

Proof. We consider the cyclic factor G_d/G_{d+1} of the series determined by $\mathcal{G}$. Since $\operatorname{dep}_{\mathcal{G}}(g) = d$, we obtain that gG_{d+1} is a non-trivial element in this factor. Further, if $r_d = \infty$, then $G_d/G_{d+1} \cong \mathbb{Z}$ and otherwise $G_d/G_{d+1} \cong \mathbb{Z}/r_d\mathbb{Z}$. In both cases we obtain an isomorphism into an additive group and the leading exponent of g yields the explicit image of gG_{d+1} under this isomorphism. Therefore, the lemma follows by elementary integer arithmetic. □

1.3 On the determination of constructive polycyclic sequences

There are various natural ways in which a polycyclic group might be given. Probably the best-known representation for polycyclic groups is by a polycyclic presentation which we recall in Chapter 2. In fact, polycyclic presentations form an important class of examples for representations of polycyclic groups which admit an effective determination of constructive polycyclic sequences. Vice versa, we will observe in Section 2.1 that each constructive polycyclic sequence determines a polycyclic presentation of the underlying group.

If a polycyclic group is given by an arbitrary finite presentation, then we can determine a constructive polycyclic sequence for this group by computing a polycyclic presentation. There are methods available for this purpose and we comment on them in Section 2.4.

There are (at least) two types of matrix representations which are of interest in the study of polycyclic groups: polycyclic matrix groups over the rational numbers and polycyclic matrix groups over finite fields. We consider both types of representations in Chapters 4 and 5 and describe methods to determine constructive polycyclic sequences in either of these two cases. Apart from being interesting in their own rights, these methods have important applications in the algorithmic theory of polycyclic groups which we exploit in later applications.

An effective method to determine a constructive polycyclic sequence for a polycyclic permutation group has been introduced by Sims in [70] and we recall it in Chapter 4. This method is based on the determination of a base and a strong generating set which is adapted to a polycyclic series of the underlying group. In particular, exponent vectors relative to such a polycyclic sequence are computed applying the ideas of a base and a strong generating set.

Further references and comments

Polycyclic sequences have been considered in various places. In many cases they have been used as synonym for the abstract generators of a polycyclic presentation. Here we have a slightly different point of view, as polycyclic sequences might arise from an arbitrary representation of the underlying group for our purposes. However, this implies that we introduce the special case of constructive polycyclic sequences. Most of the polycyclic sequences that we consider are constructive; in particular, the abstract generators of a polycyclic presentation form a constructive polycyclic sequence.

Chapter 2

Polycyclic presentations

Each polycyclic group is finitely presented. In fact, as we recall below, each polycyclic group has a presentation which exhibits its polycyclic structure: a polycyclic presentation. Polycyclic presentations allow effective computations with the groups they define. In particular, the word problem is effectively solvable in group defined by a polycyclic presentation. Further, polycyclic presentations give rise to a constructive polycyclic sequence for the groups they define. Hence they form an important class of examples for the groups considered in this book.

References: Polycyclic presentations are a well-known concept in computational group theory; in particular, many algorithms for finite groups with polycyclic presentations had been developed. Another introduction to polycyclic presentations is contained in Sims [71], Chapter 9.

2.1 Definition and relation to polycyclic sequences

A *polycyclic presentation* is a certain type of finite presentation and thus it has a set of abstract generators and relations of a certain form. In contrast to ordinary finite presentations, the generators of a polycyclic presentation are sorted into a sequence and, additionally, a polycyclic presentation has as a third parameter a list of exponents attached to it. More precisely, we define a polycyclic presentation $\mathcal{P}$ by:

- a sequence of abstract generators $\mathcal{G} = (g_1, \ldots, g_n)$.
- a sequence of *exponents* $x = (x_1, \ldots, x_n)$ with $x_i \in \mathbb{N}$ or $x_i = \infty$.
- a list of relations of the form

$$\begin{aligned} g_i^{x_i} &= w_i(g_{i+1}, \ldots, g_n) && \text{if } x_i \text{ is finite,} \\ g_j^{-1}\, g_i\, g_j &= s_{ij}(g_{j+1}, \ldots, g_n) && \text{for } 1 \le j < i \le n \text{ and} \\ g_j\, g_i\, g_j^{-1} &= t_{ij}(g_{j+1}, \ldots, g_n) && \text{for } 1 \le j < i \le n \text{ if } x_j = \infty, \end{aligned}$$

  where the right hand sides in these relations are words in the given generators.

The relations in this presentation imply that the sequence $\mathcal{G}$ of abstract generators determines a polycyclic series in the group defined by $\mathcal{P}$. The finite exponents in the sequence x yield that the corresponding factors in this polycyclic series are finite of order at most x_i. However, they do not determine the

indices of the polycyclic series precisely; that is, a factor in the polycyclic series can be finite even if the corresponding exponent is infinite or a factor in the polycyclic series can have smaller order than its corresponding exponent. In fact, the relation between the sequence of exponents and the orders of the factors of the polycyclic series determined by $\mathcal{P}$ plays a fundamental role in computing with polycyclic presentations and we investigate this correspondence in more detail below.

2.1. Lemma: *Let G be a group.*

a) If G is defined by a polycyclic presentation $\mathcal{P}$, then the abstract generators $\mathcal{G}$ of $\mathcal{P}$ form a polycyclic sequence of G. Further, $r_i \mid x_i$ for the relative orders r_i of $\mathcal{G}$ and the exponents x_i of $\mathcal{P}$.

b) Suppose G is given by a polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$ with relative orders $(r_1, \ldots, r_n)$ and finite index set I. Consider the free group F on n generators $f_1, \ldots, f_n$ and the natural epimorphism $\alpha : F \to G : f_i \mapsto g_i$. Then we obtain defining relations for G in F by

$$\begin{aligned} f_i^{r_i} &= f_{i+1}^{e(i,i+1)} \cdots f_n^{e(i,n)} \quad \text{for } i \in I, \\ f_j^{-1}\, f_i\, f_j &= f_{j+1}^{f(i,j,j+1)} \cdots f_n^{f(i,j,n)} \quad \text{for } 1 \le j < i \le n \text{ and} \\ f_j\, f_i\, f_j^{-1} &= f_{j+1}^{l(i,j,j+1)} \cdots f_n^{l(i,j,n)} \quad \text{for } 1 \le j < i \le n \quad \text{with } j \notin I \end{aligned}$$

where each word w on a left hand side defines the exponent vector of its corresponding right hand side via $\exp_{\mathcal{G}}(w^\alpha)$.

Proof. a) Follows directly from the defining relations of $\mathcal{P}$.
b) Obviously, the described relations R on the generators $f_1, \ldots, f_n$ yield a polycyclic presentation and thus they define a polycyclic group $H = F/R$. By Lemma 1.3 we obtain that the generators of G fulfill the defining relations R of H. Hence denoting $h_i = f_i R$ we obtain that $\psi : H \to G : h_i \mapsto g_i$ defines an epimorphism. It remains to show that ψ is injective. By our setup, H is a polycyclic group with polycyclic sequence $(h_1, \ldots, h_n)$. We suppose by induction that the restriction of ψ to $H_2 = \langle h_2, \ldots, h_n \rangle$ is injective and thus $H_2 \cong G_2 = \langle g_2, \ldots, g_n \rangle$. Since $H_2 \lhd H$ by the conjugation relations on f_1, we can consider the induced homomorphism $H/H_2 \to G/G_2$. This is an epimorphism of isomorphic cyclic groups by the power relation on f_1. Hence it is an isomorphism. Therefore also the original map ψ is injective. □

Thus Lemma 2.1 shows that a polycyclic sequence of a polycyclic group G determines a polycyclic presentation for G. Further, if we determine a polycyclic presentation of G by a polycyclic sequence, then the exponents of the presentation coincide with the relative orders of the polycyclic sequence. Moreover, using Lemmas 2.1 and 1.3 we obtain the following corollary.

2.2. Corollary: *A group is polycyclic if and only if it has a polycyclic presentation.*

2.2 Collected words and collection

By virtue of Lemma 2.1 we observe that the abstract generators of a polycyclic presentation form a polycyclic sequence for the group defined by the presentation, but the exponents of the presentation give only upper bounds for the relative orders of this polycyclic sequence. Hence we cannot use Lemma 1.3 to determine normal forms in a group given by a polycyclic presentation. In the following lemma we observe that we can use these upper bounds to write each element in the group in an 'almost normal form' with respect to the abstract generators of the presentation. Its proof follows from Lemmas 1.3 and 2.1.

2.3. Lemma: *Let G be defined by a polycyclic presentation $\mathcal{P}$. Then each element $g \in G$ can be expressed as a* collected word *$g = g_1^{e_1} \cdots g_n^{e_n}$ with $e_1, \ldots, e_n \in \mathbb{Z}$ and $0 \le e_i < x_i$ if x_i is finite.*

In particular, we observe that the normal form of an element with respect to the defining generators of $\mathcal{P}$ is a collected word. However, while a normal form defines an element uniquely, collected words need not be unique and an element in a polycyclically presented group may be represented by different collected words.

The *collection algorithm* can be used to determine a collected word for an arbitrary element g in a group G defined by a polycyclic presentation $\mathcal{P}$. The element g is given as a word in the defining generators of $\mathcal{P}$. There are three basic transformations on basic subwords h of g which correspond to the three types of relations for $\mathcal{P}$:

- If $h = g_j^f$ with $x_j < \infty$ and $f \notin \{0, \ldots, x_j - 1\}$, then we determine $f = qx_j + e$ with $q, e \in \mathbb{Z}$ and $0 \leq e < x_j$ and we substitute $h = g_j^e \cdot w_j^q$.
- If $h = g_i^f \cdot g_j$ for $j < i$, then we substitute $h = g_j \cdot s_{ij}^f$.
- If $h = g_i^f \cdot g_j^{-1}$ for $j < i$ with $x_j = \infty$, then we substitute $h = g_j^{-1} \cdot t_{ij}^f$.

All three basic transformations apply relations of $\mathcal{P}$ and thus they do not change the element defined by g. Further, each of them transforms an uncollected basic subword h to a subword in which the generator with smallest index g_j is in its proper position; that is, $h = g_j^e \cdot k$ where $e \in \mathbb{Z}$ and $e \in \{0, \ldots, x_j - 1\}$ if x_j is finite and k is a word in $g_{j+1}, \ldots, g_n$. The collection algorithm consists of an iterated application of these three basic transformations to g until no uncollected basic subword remains in the word representing the same element as the word g. A word with no uncollected basic subwords is collected.

```
CollectWord(P, g)
    while g is not collected do
        choose an uncollected basic subword h in g
        apply the relevant basic transformation to h
    end while
```

2.4. Lemma: *Let $\mathcal{P}$ be a polycyclic presentation and g a word in its defining generators. Then the algorithm call 'CollectWord($\mathcal{P}, g$)' terminates.*

Proof. Suppose by induction that the collection algorithm terminates on each word in the generators $g_2, \ldots, g_n$. Let $g = k_1 g_1^{e_1} k_2 \cdots g_1^{e_l} k_{l+1}$ where all k_i are words in $g_2, \ldots, g_n$. Since the algorithm terminates on the words $k_1, \ldots, k_{l+1}$, it will eventually consider an uncollected basic subword h involving g_1. The basic transformations modify h to a word in which g_1 is in its proper position. The transformations either reset the exponent of g_1 if this is appropriate or they do not change the exponent of g_1. In all cases they only introduce new words in $g_2, \ldots, g_n$. Hence after finitely many steps we obtain $g = g_1^{e_1 + \ldots + e_l \bmod x_1} \cdot k$ and k is a word in $g_2, \ldots, g_n$. At this stage all uncollected basic subwords are contained in k and by our induction hypothesis the algorithm terminates. □

A critical factor for the collection algorithm is the strategy which is used to choose the next uncollected subword to transform. There are various strategies known for this purpose. In the following list we recall three of the well-known collection strategies:

- Collection from the left: We always transform the left-most uncollected subword of g.
- Collection to the left: Let i be the smallest index such that $g_i^{\pm 1}$ occurs in g. Then we move all occurrences of $g_i^{\pm 1}$ to the left of g using the basic transformations. Now $g = g_i^e k$ and k is a word in $g_{i+1}, \ldots, g_n$. We apply this approach recursively to k.
- Collection from the right: We always transform the right-most uncollected subword of g.

Collection to the left has been the original collection strategy introduced by Hall in 1934 for free nilpotent groups. It has properties which are useful in theoretical arguments. But extensive experiments by Leedham-Green & Soicher [40] lead to the observation that collection from the left is usually superior in its effectiveness to the other strategies and hence this strategy is mostly used in implementations.

References: A detailed account on collection in finite polycyclic groups considering various strategies is described by Leedham-Green & Soicher [40]. They also introduced a "symbolic collection" method using Hall polynomials in [41]. Further collection methods for finite polycyclic groups have been considered by Brückner [7].

2.3 Consistency of polycyclic presentations

We call a polycyclic presentation $\mathcal{P}$ *consistent* (or *confluent*) if there exists exactly one collected word for each word in the defining generators. Such polycyclic presentations are particularly interesting, since the word problem can be solved effectively in them using the collection algorithm of Section 2.2. The following lemma can be used to check the consistency of a polycyclic presentation.

2.5. Lemma: *Let $\mathcal{P}$ be a polycyclic presentation with generators $\mathcal{G}$. Then the following three conditions are equivalent.*

a) $\mathcal{P}$ is consistent.

b) The relative orders of the polycyclic sequence $\mathcal{G}$ coincide with the exponents of $\mathcal{P}$.

c) The words on the right hand sides and the left hand sides of the following equations yield the same collected word in $\mathcal{P}$ where the subwords in brackets are collected first.

$$\begin{aligned}
g_k(g_j g_i) &= (g_k g_j) g_i && \text{for} && k > j > i \\
(g_j^{x_j}) g_i &= g_j^{x_j - 1}(g_j g_i) && \text{for} && j > i, j \in I \\
g_j(g_i^{x_i}) &= (g_j g_i) g_i^{x_i - 1} && \text{for} && j > i, i \in I \\
(g_i^{x_i}) g_i &= g_i(g_i^{x_i}) && \text{for} && i \in I \\
g_j &= (g_j g_i^{-1}) g_i && \text{for} && j > i, i \notin I
\end{aligned}$$

Proof. The equivalence of a) and b) follows from Lemmas 2.3 and 2.1. The equivalence of a) and c) is essentially proved in [71], page 424. Note that in [71] there are more defining relations used in a polycyclic presentation and thus two more consistency relations are needed to check the compatibility of the additional relations. □

We can now use Lemma 2.5 c) and the collection algorithm to obtain a method for checking the consistency of a polycyclic presentation $\mathcal{P}$ as follows.

```
IsConsistent(P)
    for each equation g = h in Lemma 2.5 c) do
        collect g with CollectWord respecting brackets in g
        collect h with CollectWord respecting brackets in h
        if g ≠ h then return false
    end for
    return true
```

The polycyclic presentations obtained from polycyclic sequences as in Lemma 2.1 are obviously consistent. Since each polycyclic group has a polycyclic sequence, we obtain the following corollary.

2.6. Corollary: *Each polycyclic group has a consistent polycyclic presentation.*

In summary, the defining generators of a consistent polycyclic presentation together with the collection algorithm form a constructive polycyclic sequence for the underlying group. Hence consistent polycyclic presentations form an important class of examples for the groups considered in this book.

2.4 On the determination of polycyclic presentations

If a polycyclic group G and a constructive polycyclic sequence $\mathcal{G}$ for G is given, then a polycyclic presentation for G can be determined using Lemma 2.1. Hence we can construct polycyclic presentations for polycyclic permutation groups and polycyclic matrix groups using the methods of the Sections 4 and 5. Another main source for polycyclic groups are finitely presented groups. A variety of methods to determine quotients with certain properties of a given finitely presented group are known. For example, Newman and O'Brien [53] developed a method to construct finite p-quotients. Then there are two different approaches to compute finite polycyclic quotients: one introduced by Plesken and Brückner [62, 7], and another one described by Niemeyer [55]. Further, Nickel presents an algorithm to compute nilpotent quotients in [54]. Lo [42] introduces a method to determine polycyclic quotients. An alternative approach to the determination of polycyclic quotients is currently developed in joint work with Niemeyer [24].

Further references and comments

More details on polycyclic presentations can be found in [71]. However, we note that in [71] a polycyclic presentation has two more types of defining relations:

$$\begin{aligned} g_j^{-1}\, g_i^{-1}\, g_j &= u_{ij}(g_{j+1}, \ldots, g_n) \quad \text{for } 1 \le j < i \le n \text{ if } x_j = \infty \text{ and} \\ g_j\, g_i^{-1}\, g_j^{-1} &= v_{ij}(g_{j+1}, \ldots, g_n) \quad \text{for } 1 \le j < i \le n \text{ if } x_i, x_j = \infty. \end{aligned}$$

It is straightforward to observe that $u_{ij}^{-1} = s_{ij}$ and $v_{ij}^{-1} = t_{ij}$ in the notation of Section 2.1. Hence we can omit these two additional relations in a group presentation.

Chapter 3

Subgroups, factor groups and homomorphisms

Subgroups and factor groups of a polycyclic group G are polycyclic as well. Thus we would like to exhibit the polycyclic structure of subgroups and factor groups in a form that is compatible with the polycyclic structure of G. More precisely, we suppose that a constructive polycyclic sequence of G is given and we introduce methods to determine constructive polycyclic sequences for subgroups and factor groups of G. These algorithms then lead naturally to methods to compute with homomorphisms of polycyclic groups.

References: Algorithms to compute with subgroups of finite polycyclic groups have been presented by Laue, Neubüser & Schoenwaelder in [38]. The methods described here are generalizations of those algorithms. Another approach to the problems considered in this chapter has been outlined by Sims [71], Chapter 9. Further, algorithms to compute with group homomorphisms are described by Leedham-Green, Praeger & Soicher in [39].

3.1 Subgroups and induced polycyclic sequences

We consider a subgroup U of a polycyclic group G. Let $\mathcal{G}$ and $\mathcal{U}$ be polycyclic sequences of G and U, respectively. Suppose that $\mathcal{G}$ determines the series $G_1, \ldots, G_{n+1}$. Then we call $\mathcal{U}$ an *induced polycyclic sequence* for U with respect to $\mathcal{G}$ if the series determined by $\mathcal{U}$ is obtained from $U \cap G_1, \ldots, U \cap G_{n+1}$ by removing duplicate subgroups. We first give a criterion for induced polycyclic sequences.

3.1. Lemma: *Let G be a polycyclic group with polycyclic sequence $\mathcal{G}$. Consider a sequence $\mathcal{U} = (u_1, \ldots, u_l)$ and let $U = U_1, \ldots, U_{l+1}$ be the series determined by $\mathcal{U}$. Then $\mathcal{U}$ is an induced polycyclic sequence for U with respect to $\mathcal{G}$ if and only if*

a) $u_j^{u_i} \in U_{i+1}$ *for* $1 \leq i < j \leq l$.

b) $u_i^{s_i} \in U_{i+1}$ *if* $s_i = \mathrm{relord}_{\mathcal{G}}(u_i)$ *is finite.*

c) $\mathrm{dep}_{\mathcal{G}}(u_1) < \ldots < \mathrm{dep}_{\mathcal{G}}(u_l)$.

Proof. $\Rightarrow$ If $\mathcal{U}$ is an induced polycyclic sequence, then by definition the series determined by $\mathcal{U}$ is polycyclic and thus a) follows. If $U_i = U \cap G_{d_i}$ and $U \cap G_{d_i+1} < U_i$, then $d_i = \mathrm{dep}_{\mathcal{G}}(u_i)$ and hence the depths of the elements in $\mathcal{U}$ are strictly decreasing which yields c). Finally, $[U_i : U_{i+1}] = [U \cap G_{d_i} : U \cap G_{d_i+1}] = [UG_{d_i+1} \cap G_{d_i} : G_{d_i+1}] = \mathrm{relord}_{\mathcal{G}}(u_i) = s_i$ and hence we obtain b).

$\Leftarrow$ First note that the condition in a) implies $u_j^{u_i^{-1}} \in U_{i+1}$ for $1 \leq i < j \leq l$, since G is polycyclic and thus U_{i+1} cannot be conjugate to a proper subgroup of itself. Hence part a) asserts that the series determined by $\mathcal{U}$ is subnormal. Thus the series is polycyclic and $\mathcal{U}$ is a polycyclic sequence. We use induction to show that $\mathcal{U}$ is induced by $\mathcal{G}$. We assume that $\mathcal{U}_2 = (u_2, \ldots, u_l)$ is an induced polycyclic sequence for U_2 with respect to $\mathcal{G}$. Let $d_i = \mathrm{dep}_{\mathcal{G}}(u_i)$. By construction, $U \leq G_{d_1}$ and thus $U = U \cap G_{d_1}$. By c) we obtain that $U_2 \leq G_{d_2} < G_{d_1}$. Further, if there exists an index e with $U_2 = U \cap G_{d_2} < U \cap G_e < U \cap G_{d_1} = U$, then $u_1 U_2$ has finite order and $u_1^{s_1} \in G_e \setminus G_{d_2}$ which contradicts b). Hence the series induced by $\mathcal{U}$ coincides with $U \cap G_1, \ldots, U \cap G_{n+1}$ after removing duplicate subgroups. □

3.2. *Remark:* Let $\mathcal{U}$ be an induced polycyclic sequence with respect to $\mathcal{G}$. Then the matrix formed by the exponent vectors of the elements in $\mathcal{U}$ with respect to $\mathcal{G}$ is in semi-echelonized form.

The following corollary observes that the relative orders of an induced polycyclic sequence for a subgroup can be read off. Thus we can also determine the order and the Hirsch length of a subgroup. Further, the corollary shows that an induced polycyclic sequence can be used to determine the index of its subgroup.

3.3. Corollary: *Let G be a polycyclic group with polycyclic sequence $\mathcal{G}$ and let $\mathcal{U} = (u_1, \ldots, u_l)$ be an induced polycyclic sequence for U with respect to $\mathcal{G}$.*

a) The relative orders of $\mathcal{U}$ are $(\mathrm{relord}_{\mathcal{G}}(u_1), \ldots, \mathrm{relord}_{\mathcal{G}}(u_l))$.

b) With $D = \{\mathrm{dep}_{\mathcal{G}}(u_i) \mid 1 \leq i \leq l\}$ *we obtain*

$$[G : U] = \prod_{i \in \{1,\ldots,l\}} \mathrm{relind}_{\mathcal{G}}(u_i) \cdot \prod_{j \in \{1,\ldots,n\} \setminus D} \mathrm{relord}_{\mathcal{G}}(g_j).$$

By definition, an induced polycyclic sequence for a subgroup U is a polycyclic sequence and thus, by Lemma 1.3, each element of U has a unique normal form with respect to $\mathcal{U}$. We introduce the algorithm 'ExponentVector' which determines the exponent vector of an element u with respect to $\mathcal{U}$ if $u \in U$. If $u \notin U$, then the algorithm returns fail. Hence this algorithm also yields a membership test in subgroups given by induced polycyclic sequences. In particular, we obtain that $\mathcal{U}$ is constructive if $\mathcal{G}$ is constructive.

```
ExponentVector(G,U,g)
    initialize e as zero vector of length l where l = |U|
    let (r_1,...,r_n) be the relative orders of G
    while g ≠ 1 do
        let d = dep_G(g)
        find j with dep_G(u_j) = d
            (if j does not exist, then return fail)
        let s = lead_G(u_j) and r = lead_G(g)
        solve xs = r or xs ≡ r mod r_d with x ∈ {0,...,r_d − 1} if r_d < ∞
            (if x does not exist, then return fail)
        store e_j = x
        replace g by u_j^{−x} · g
    end while
    return e
```

In many computations with subgroups of polycyclic groups the parent group G is given by a constructive polycyclic sequence $\mathcal{G}$, while the subgroup U is just given by generators. Thus we introduce an algorithm to compute an induced polycyclic sequence for U from a generating set in the following.

The determination of a generator for a subgroup of a cyclic group is equivalent to a gcd computation. We extend this feature to the factors of the polycyclic series $G_1, \ldots, G_{n+1}$ determined by $\mathcal{G}$ as follows. Let $h_1, \ldots, h_r \in G$ with $\mathrm{dep}_{\mathcal{G}}(h_i) = d$. Then $h_i \equiv g_d^{l_i} \bmod G_{d+1}$ for $l_i = \mathrm{lead}_{\mathcal{G}}(h_i)$. We consider $e = \gcd(l_1, \ldots, l_r)$ with its corresponding linear combination $e = l_1x_1 + \ldots + l_rx_r$ and we define $\gcd(h_1, \ldots, h_r) = h_1^{x_1} \cdots h_r^{x_r}$. The following properties of this gcd are used throughout. To shorten notation we denote $k = \gcd(h_1, \ldots, h_r)$ and $U = \langle h_1, \ldots, h_r \rangle$.

- $k \in U$ and $k \equiv g_d^e \bmod G_{d+1}$. Thus kG_{d+1} generates UG_{d+1}/G_{d+1}.
- $hG_{d+1} \in UG_{d+1}/G_{d+1}$ is a generator of UG_{d+1}/G_{d+1} if and only if $\mathrm{relind}_{\mathcal{G}}(h) = \mathrm{relind}_{\mathcal{G}}(k)$.
- If $hG_{d+1} \in UG_{d+1}/G_{d+1}$ is a generator of UG_{d+1}/G_{d+1}, then for each $u \in U$ there exists an integer $q \in \mathbb{Z}$ with $h^{-q} \cdot u \in G_{d+1}$. We denote such a quotient by $\mathrm{quot}(u, h) = h^{-q} \cdot u$.
- If $h \in U$ and hG_{d+1} generates UG_{d+1}/G_{d+1}, then $U = \langle h, \mathrm{quot}(h_1, h), \ldots, \mathrm{quot}(h_r, h) \rangle$.

The algorithm to determine an induced polycyclic sequence for a subgroup U given by generators uses an induction approach. We build up an ascending series of subgroups by adding new generators to the existing subgroup. Then we close the currently considered subgroup using powers and commutators as outlined in Lemma 3.1 to obtain an induced polycyclic sequence.
Thus in the inductive step of the algorithm we have a sequence $\mathcal{S}$ of elements in U such that $\mathcal{S}$ has the same length as $\mathcal{G}$ and position d in $\mathcal{S}$ is either empty or contains an element of depth d with respect to $\mathcal{G}$. The following 'CloseSequence' algorithm modifies such a sequence $\mathcal{S}$ in place to another sequence $\mathcal{S}'$ of this type and it closes $\mathcal{S}$ with an element g. The procedure returns the elements of $\mathcal{S}' \setminus \mathcal{S}$.

```
CloseSequence(G, S, g)
    initialize C as an empty list and let L = {g}
    for d in {1, ..., n} do
        let L_d = {h ∈ L | dep_G(h) = d}
        if L_d ≠ ∅ and S_d is empty then
            determine k = gcd(L)
            insert k into S_d and add k to C
        else if L_d ≠ ∅ and S_d is not empty then
            determine k = gcd(L_d, S_d)
            if relind_G(k) = relind_G(S_d) then
                replace k by S_d
            else
                replace S_d by k and add k to C
                add quot(S_d, k) to L
            end if
        end if
        for h in L with dep_G(h) = d do
            replace h by quot(h, k)
        end for
    end for
    return C
```

The induction process to determine an induced polycyclic sequence for U from a generating set $\mathcal{E}$ is now obtained by the procedure 'InducedPcSequence'. It initializes $\mathcal{S}$ as an empty sequence of length n and then successively shifts all generators of U into $\mathcal{S}$. Additionally, it closes $\mathcal{S}$ at each stage under powers and commutators as outlined in Lemma 3.1.

InducedPcSequence($\mathcal{G}, U$)
  initialize $i = 1$ and $\mathcal{S}$ as an empty sequence of length n
  initialize $\mathcal{E}$ as the generating set of U
  while $i <=$ Length($\mathcal{E}$) do
    let $\mathcal{C}$ = CloseSequence($\mathcal{G}, \mathcal{S}, \mathcal{E}_i$)
    for $g \in \mathcal{C}$ and $h \in \mathcal{S}$ add $[g, h]$ to $\mathcal{E}$
    for $g \in \mathcal{C}$ add $g^{\mathrm{relord}_{\mathcal{G}}(g)}$ to $\mathcal{E}$
    increase $i = i + 1$
  end while
  let $\mathcal{U}$ be the subsequence of $\mathcal{S}$ obtained by removing empty places
  return $\mathcal{U}$

3.4. Lemma: *Let $U \leq G$ and consider a call of 'InducedPcSequence($\mathcal{G}, U$)'.*

a) The algorithm 'InducedPcSequence' terminates.

b) The output $\mathcal{U}$ of the algorithm is an induced polycyclic sequence for U with respect to $\mathcal{G}$.

Proof. a) In each pass through the for loop of 'CloseSequence' the depth of the considered elements is increased and hence 'CloseSequence' terminates. Whenever 'CloseSequence' returns a non-trivial set, then there is a factor G_i/G_{i+1} in the polycyclic series determined by $\mathcal{G}$ such that the modified sequence $\mathcal{S}'$ covers a larger subgroup in this factor than the previous sequence $\mathcal{S}$ did. This can only happen finitely many times, since there are only finitely many factors in the polycyclic series and ascending subgroup series in polycyclic groups are finite. Hence after finitely many iterations of 'CloseSequence' we arrive at the stage where the output of 'CloseSequence' is always the empty set. Thus at this point the set $\mathcal{E}$ is not enlarged further and eventually 'InducedPcSequence' terminates.
b) Clearly, all elements inserted into $\mathcal{S}$ lie in U by construction and thus $\mathcal{U} \subseteq U$. Further, $\mathcal{U}$ generates U, since we successively insert a generating set of U into $\mathcal{S}$. We consider the conditions of Lemma 3.1 to show that $\mathcal{U}$ is an induced polycyclic sequence for U. Condition c) on the depth of elements is satisfied by the construction of $\mathcal{S}$. Conditions a) and b) are fulfilled, since we close $\mathcal{S}$ under taking the required powers and commutators in 'InducedPcSequence'. Thus we obtain that $\mathcal{U}$ is an induced polycyclic sequence by Lemma 3.1. □

3.2 Canonical polycyclic sequences

Induced polycyclic sequences are useful tools in computations with polycyclic groups. However, they are not unique for the subgroup they define. We use an approach similar to the Hermite normal form method to construct unique induced polycyclic sequences.
Let G be a polycyclic group with polycyclic sequence $\mathcal{G}$. We call an element $g \in G$ *normalized* with respect to $\mathcal{G}$ if $\mathrm{lead}_{\mathcal{G}}(g) = \mathrm{relind}_{\mathcal{G}}(g)$. For each g there exists a power g^e which is normalized and e can be determined using integer arithmetic as observed in Lemma 1.4.
Let $g, h \in G$ with $\mathrm{dep}_{\mathcal{G}}(g) < \mathrm{dep}_{\mathcal{G}}(h) = d$. Then g is *reduced* with respect to h, if the d-th entry e_d in $\exp_{\mathcal{G}}(g)$ is reduced; that is, $0 \leq e_d < \mathrm{lead}_{\mathcal{G}}(h)$. If g is not reduced with respect to h, then we can determine $e = \mathrm{lead}_{\mathcal{G}}(h)q + r$ with $0 \leq r < \mathrm{lead}_{\mathcal{G}}(h)$ and obtain that gh^{-q} is reduced with respect to h.
We define that the sequence $\mathcal{U} = (u_1, \ldots, u_l)$ is a *canonical polycyclic sequence*, if $\mathcal{U}$ is an induced polycyclic sequence which consists of normalized elements and each element u_i is reduced with respect to $u_{i+1}, \ldots, u_l$. The following algorithm determines a canonical polycyclic sequence using its definition.

CanonicalPcSequence($\mathcal{G}, U$)
  let $\mathcal{U} = (u_1, \ldots, u_l) =$ InducedPcSequence($\mathcal{G}, U$)
  for i from 1 to l do
    normalize u_i
    reduce $u_1, \ldots, u_{l-1}$ with respect to u_i
  end for
  return $\mathcal{U}$

3.5. Lemma: *A subgroup U of G has a unique canonical polycyclic sequence with respect to $\mathcal{G}$.*

Proof. We consider two canonical polycyclic sequences for U and show that they are equal. First note that they are both induced with respect to $\mathcal{G}$ and hence determine the same polycyclic series for U. We use induction upwards this series to prove equality. Hence we suppose by induction that $\mathcal{U}_2 = (u_2, \ldots, u_l)$ is the unique canonical polycyclic sequence for U_2. Suppose that u and v are elements in U such that $(u, \mathcal{U}_2)$ and $(v, \mathcal{U}_2)$ are canonical polycyclic sequences for U. Then $\mathrm{dep}_{\mathcal{G}}(u) = \mathrm{dep}_{\mathcal{G}}(v)$ and thus, since u and v are normalized, $\mathrm{lead}_{\mathcal{G}}(u) = \mathrm{lead}_{\mathcal{G}}(v)$. Hence $uv^{-1} \in U_2$. But u and v are both reduced with respect to all elements in $\mathcal{U}_2$ and therefore induction on $\mathrm{dep}_{\mathcal{G}}(u) + 1, \ldots, n$ shows that u and v are equal. □

3.6. *Remark:* Let G be a group given by a constructive polycyclic sequence. Let U and H be subgroups of G. Then we can test equality of U and H either by computing canonical polycyclic sequences for both subgroups or we test $U \leq H$ by applying the membership test in H to the generators of U and then we check $[H : U] = 1$ using Corollary 3.3.

3.3 Normal closures of subgroups

We observe in the following algorithm that the normal closure U^H for two subgroups $U, H \leq G$ can be obtained as an application of a method to test equality for subgroups of G.

NormalClosure(H, U)
  let $K = U$
  repeat
    set $L = K$
    compute $K = \langle L^h \mid h \text{ generator of } H \rangle$
  until $K = L$
  return K

Note that in the above method we do not need to consider conjugates under inverses of generators of H, since $K^h \leq K$ implies that $K^h = K$ and thus $K^{h^{-1}} = K$ for subgroups K of a polycyclic group G.
A similar approach can be used to check if a subgroup U of G is normal under the action of H.

IsNormal(H, U)
  for each generator h of H do
    if $u^h \notin U$ for a generator u of U, then return false
  end for
  return true

3.4 Induced polycyclic sequences for factor groups

We consider two subgroups $U, N \leq G$ with $N \trianglelefteq U$. Suppose that $\mathcal{G}$ is a polycyclic sequence for G and $\mathcal{U}$ and $\mathcal{N}$ are induced polycyclic sequences for U and N, respectively. We denote $\mathcal{U} = (u_1, \ldots, u_l)$ and use this to define $\mathcal{U}/N = (u_1N, \ldots, u_lN)$. It is straightforward to observe that this is a polycyclic sequence for U/N and we denote it as the *induced polycyclic sequence* for the subfactor U/N. In the following lemma we consider the relative orders corresponding to $\mathcal{U}/N$.

3.7. Lemma: *Let $\mathcal{U}$ and $\mathcal{N}$ be induced polycyclic sequences with respect to $\mathcal{G}$ for subgroups U and N. Let $\mathcal{U}/N$ be a polycyclic sequence for U/N where $\mathcal{U} = (u_1, \ldots, u_l)$. Then we have the following.*

a) If $U_1, \ldots, U_{l+1}$ is the polycyclic series determined by $\mathcal{U}$, then $U_1N/N, \ldots, U_{l+1}N/N$ is the polycyclic series determined by $\mathcal{U}/N$.

b) The relative orders $(\overline{r}_1, \ldots, \overline{r}_l)$ of $\mathcal{U}/N$ are determined by $\mathcal{U}$ and $\mathcal{N}$ as

$$\overline{r}_i = \begin{cases} \mathrm{relind}_{\mathcal{G}}(n_j)/\mathrm{relind}_{\mathcal{G}}(u_i) & \text{if } \mathrm{dep}_{\mathcal{G}}(n_j) = \mathrm{dep}_{\mathcal{G}}(u_i) \text{ for some } n_j, \text{ and} \\ \mathrm{relord}_{\mathcal{G}}(u_i) & \text{otherwise.} \end{cases}$$

Proof. a) is obvious, since $U \to U/N$ is an epimorphism. b) can now be read off, since $\mathcal{U}$ and $\mathcal{N}$ are both induced with respect to $\mathcal{G}$. □

3.8. *Remark:* Let $\mathcal{U}/N$ be an induced polycyclic sequence for U/N.

- The relative orders $(\overline{r}_1, \ldots, \overline{r}_l)$ corresponding to $\mathcal{U}/N$ might contain entries $\overline{r}_i = 1$. In this case we can discard u_iN from the sequence $\mathcal{U}/N$ without loss of information.
- The relative orders of $\mathcal{U}/N$ yield the index of N in U by Corollary 3.3.

Hence the sequences $\mathcal{U}$ and $\mathcal{N}$ allow us readily to read off the relative orders of $\mathcal{U}/N$. We show that they also allow us the effective determination of exponent vectors with respect to $\mathcal{U}/N$ if the underlying polycyclic sequence $\mathcal{G}$ is constructive. Hence we obtain that $\mathcal{U}/N$ is also constructive.

```
ExponentVector(G,U,N,g)
    let (r_1,...,r_n) be the relative orders of G
    let (r̄_1,...,r̄_l) be the relative orders of U/N
    initialize e as zero vector of length l
    while g ≠ 1 do
        let d = dep_G(g)
        if there exists a j with dep_G(u_j) = d then
            let s = lead_G(u_j) and r = lead_G(g)
            solve xs = r or xs ≡ r mod r̄_j with x ∈ {0,...,r̄_j − 1} if r̄_j is finite
            set e_j = x
            replace g by u_j^{−x} · g
        end if
        if d = dep_G(g) then
            find j with dep_G(n_j) = d
            let s = lead_G(n_j) and r = lead_G(g)
            solve xs = r or xs ≡ r mod r_d if r_d is finite
            replace g by n_j^{−x} · g
        end if
    end while
    return e
```

In summary, if $\mathcal{G}$ is a constructive polycyclic sequence for G and $N \trianglelefteq U \leq G$ are given by induced polycyclic sequences $\mathcal{N}$ and $\mathcal{U}$, then we can derive an induced polycyclic sequence for the factor U/N, compute its relative orders and determine effectively exponent vectors with respect to this polycyclic sequence.

3.5 Homomorphisms

Let $k_1, \ldots, k_r$ be a generating set of a polycyclic group G and consider a set of elements $\overline{k}_1, \ldots, \overline{k}_r$ in a group H. We first consider the question if the map $\varphi : G \to H : k_i \mapsto \overline{k}_i$ extends to a homomorphism from G to H. If this is the case and φ defines a homomorphism, then we are interested in computing with this homomorphism; that is, we want to determine its image and its kernel and we want to construct images and preimages of elements or subgroups in G or H.
In solving these problems we consider three cases: G has a constructive polycyclic sequence or H has a constructive polycyclic sequence or both groups have constructive polycyclic sequences. We solve the above problems depending on the given constructive polycyclic sequences in the Sections 3.5.1, 3.5.2 and 3.5.3.
Further, we consider the special case that the image H is an abelian group. We show in Section 3.5.4 that in this special case we can compute more effectively with group homomorphisms φ than in the general case. In fact, computations with φ can mainly be reduced to linear algebra calculations in this case.

3.5.1 Verifying homomorphisms and computing images

Suppose that G has a constructive polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$. In this section we introduce a *source induced form* relative for φ; that is, an induced form with respect to the polycyclic sequence $\mathcal{G}$ of the source G. We show that this form allows us to verify whether φ is a group homomorphism. Further, if φ is a group homomorphism, then we observe that we can use the source induced form to read off images of elements and subgroups of G.
For this purpose we determine a canonical polycyclic sequence with respect to $\mathcal{G}$ by applying the method of Section 3.2 to the generators $k_1, \ldots, k_r$ of G. Clearly, the resulting canonical polycyclic sequence will be equal to $\mathcal{G}$, since canonical polycyclic sequences are unique. During the process of computing the canonical polycyclic sequence we apply all operations on $k_1, \ldots, k_r$ simultaneously to $\overline{k}_1, \ldots, \overline{k}_r$. This process is also called *shadowing*. Thus we obtain the images of $\mathcal{G}$ under φ:

$$\varphi_S : G \to H : g_i \mapsto \overline{g}_i \text{ for } 1 \leq i \leq n.$$

Now φ is a group homomorphism if φ_S is a group homomorphism. The later can be checked using the polycyclic presentation corresponding to the polycyclic sequence $\mathcal{G}$ as determined in Lemma 2.1: We need to verify that the elements $(\overline{g}_1, \ldots, \overline{g}_n)$ fulfill the relations of this presentation.
Further, since $\mathcal{G}$ is a constructive polycyclic sequence, we can write each element g of G as a word in normal form in $\mathcal{G}$. Hence if φ is a group homomorphism, then we can determine images of elements, subsets or subgroups of G using φ_S.

3.5.2 Computing the kernel and preimages

Suppose that both groups G and H have constructive polycyclic sequences $\mathcal{G} = (g_1, \ldots, g_n)$ and $\mathcal{H} = (h_1, \ldots, h_m)$, respectively, and let φ be a group homomorphism. In this section we introduce the *two-sided induced form* for φ; that is, an induced form to $\mathcal{H}$ and $\mathcal{G}$. We show that this induced form allows us to determine generators for $ker(\varphi)$. Further, we observe that we can construct preimages of elements and subgroups of H using the two-sided induced form for φ.

We consider the direct product $H \times G = \{(h, g) \mid h \in H, g \in G\}$ which has a constructive polycyclic sequence $\mathcal{H} \times \mathcal{G} = ((h_1, 1), \ldots, (h_m, 1), (1, g_1), \ldots, (1, g_n))$. Let $U = \langle (\overline{k}_i, k_i) \mid 1 \leq i \leq r \rangle$ be the diagonal in $H \times G$ corresponding to φ. We use the method of Section 3.1 to determine an induced polycyclic sequence for U with respect to $\mathcal{H} \times \mathcal{G}$. This is a sequence of the form $\mathcal{U} = ((\overline{u}_1, u_1), \ldots, (\overline{u}_s, u_s))$ with $u_i \in G$ and $\overline{u}_i \in H$. We obtain

$$\varphi_I : G \to H : u_i \mapsto \overline{u}_i \text{ for } 1 \leq i \leq s.$$

3.9. Lemma: *Let φ be a group homomorphism with two-sided induced form φ_I. Let t be maximal with $\overline{u}_t \neq 1$. Then we have the following.*

a) $(\overline{u}_1, \ldots, \overline{u}_t)$ is an induced polycyclic sequence for the image of φ with respect to $\mathcal{H}$.

b) $(u_{t+1}, \ldots, u_s)$ is an induced polycyclic sequence for the kernel of φ with respect to $\mathcal{G}$.

Proof. Projecting $\mathcal{U}$ into H yields an induced polycyclic sequence with respect to $\mathcal{H}$ by construction. Thus a) follows and we consider b). By construction, we have $u_i \in ker(\varphi)$ for $t < i \leq s$. Further, the sequence $(u_{t+1}, \ldots, u_s)$ is an induced polycyclic sequence with respect to $\mathcal{G}$. Hence it remains to show that the sequence of elements in b) generates $ker(\varphi)$. Let $k \in ker(\varphi)$. We consider $(1, k) \in U \leq H \times G$ and write this element as a word in normal form with respect to $\mathcal{U}$, say $(1, k) = (\overline{u}_1, u_1)^{e_1} \cdots (\overline{u}_s, u_s)^{e_s}$. Then $\overline{u}_1^{e_1} \cdots \overline{u}_t^{e_t} = 1$. By a) this yields $e_1 = \ldots = e_t = 0$. Hence $k = u_{t+1}^{e_{t+1}} \cdots u_s^{e_s}$ and the sequence considered in part b) generates $ker(\varphi)$. □

Hence the kernel of φ can be read off from φ_I as shown in Lemma 3.9 b). For the determination of a preimage of an element $h \in H$, we write h as a word in the induced polycyclic sequence of the image of φ using the methods of Section 3.1 and then read off the preimage of h from this word and φ_I. Full preimages of subgroups of H can be obtained by constructing a preimage for each generator of the subgroup and append a generating set of $ker(\varphi)$ to obtain a generating set of the full preimage.

3.10. Corollary: *Surjectivity and injectivity of φ can be read off from φ_I by Lemma 3.9.*

3.11. *Remark:* The computation in the direct product $H \times G$ used here is in fact an explicit formulation for the shadowing process described in Section 3.5.1.

3.5.3 Computing the kernel, revisited

In Section 3.5.2 we described an algorithm to compute the kernel of a homomorphism $\varphi : G \to H$ if G and H both have a constructive polycyclic sequence. However, in some later applications of group homomorphisms we only know that the range H has a constructive polycyclic sequence, but no such sequence for G is given. We observe in this section that it is still possible to determine elements in the kernel of φ which generate the kernel as normal subgroup of G.
We use a similar approach as in Section 3.5.1 and apply the method of Section 3.1 to determine an induced polycyclic sequence $(\overline{u}_1, \ldots, \overline{u}_t)$ for $\langle \overline{k}_1, \ldots, \overline{k}_r \rangle$ with respect to $\mathcal{H}$. We shadow the elements $k_1, \ldots, k_r$ through this process and obtain preimages $u_1, \ldots, u_t$ with $\overline{u}_i = u_i^{\varphi}$. There are two ways to determine normal subgroup generators for the kernel of φ in this setting.

First method: We compute the polycyclic presentation corresponding to $(\overline{u}_1, \ldots, \overline{u}_t)$ as described in Lemma 2.1. The relators of this presentation can be considered as words in the generators $\overline{u}_1, \ldots, \overline{u}_t$. We evaluate these words in the preimages $u_1, \ldots, u_t$ and thus obtain normal subgroup generators of the kernel.

Second method: We consider the simultaneous computation of the sequences $(\overline{u}_1, \ldots, \overline{u}_t)$ and $(u_1, \ldots, u_t)$ in more detail. We obtain these two sequences using a modification of the algorithm 'InducedPcSequence' of Section 3.1. If we store all the preimages of the identity elements occurring the application of 'InducedPcSequence' to the generators of the image, then we obtain a normal subgroup generating set of the kernel of φ. This second approach is closely related to the method described in Section 3.5.2.

We summarize our methods to determine the kernel of a group homomorphism as follows. As above, let G and H be polycyclic groups. Let $k_1, \ldots, k_r$ be a generating set for G and suppose that $\varphi : G \to H : k_i \mapsto \overline{k}_i$ is a homomorphism from G to H. Further, we suppose that we have a constructive polycyclic sequence $\mathcal{H}$ for H. Then the following algorithm yields either generators or normal subgroup generators for $ker(\varphi)$.

Kernel$(G, H, \varphi : G \to H : k_i \mapsto \overline{k}_i)$
  let $\mathcal{H}$ be a constructive polycyclic sequence for H
  if G has a constructive polycyclic sequence then
    determine the two-sided induced form φ_I
    read off an induced polycyclic sequence for $ker(\varphi)$ using Lemma 3.9
  else
    compute an induced polycyclic sequence $(\overline{u}_1, \ldots, \overline{u}_t)$ for $im(\varphi)$ with respect to $\mathcal{H}$
    simultaneously obtain preimages $(u_1, \ldots, u_t)$ with $u_i^{\varphi} = \overline{u}_i$
    determine the polycyclic presentation corresponding to $(\overline{u}_1, \ldots, \overline{u}_t)$ by Lemma 2.1
    evaluate its relators in $(u_1, \ldots, u_s)$ and obtain normal subgroup generators for $ker(\varphi)$
  end if
  return $ker(\varphi)$

3.12. *Remark:* If we are able to test equality for subgroups in the kernel K, then we can use the methods of Section 8.6 to obtain a subgroup generating set for K from normal subgroup generators.

3.5.4 Homomorphisms with abelian image

In this section we consider the special case of group homomorphisms with abelian image group. This case has many applications in later algorithms and thus we are interested in effective methods to handle it. The following lemma shows that the kernel of such a homomorphism can be determined using mainly integer arithmetic.

3.13. Lemma: *Let G be a polycyclic group with a polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$. We consider a group homomorphism $\varphi : G \to H : g_i \mapsto \overline{g}_i$ into an additively written abelian group H. The set $rl(\overline{\mathcal{G}}) = \{(e_1, \ldots, e_n) \in \mathbb{Z}^n \mid e_1\overline{g}_1 + \ldots + e_n\overline{g}_n = 0\}$ is a sublattice of $\mathbb{Z}^n$ called the* relation lattice *for $\overline{g}_1, \ldots, \overline{g}_n$. Let $b_1, \ldots, b_l$ be a basis in upper triangular form for $rl(\overline{\mathcal{G}})$ and let $k_i \in G$ with $exp_{\mathcal{G}}(k_i) = b_i$. Then the sequence $\mathcal{K} = (k_1, \ldots, k_l)$ is a polycyclic sequence for $ker(\varphi)$.*

Proof. By construction, the elements k_i are contained in $ker(\varphi)$. Hence $K = \langle \mathcal{K} \rangle \leq ker(\varphi)$. We show that $\mathcal{K}$ generates $ker(\varphi)$ using induction on the length of $\mathcal{G}$. In the inductive step we suppose that all elements $g \in ker(\varphi)$ with $\mathrm{dep}_{\mathcal{G}}(g) > i$ are contained in K. We consider $g \in ker(\varphi)$ with depth i and show that $g \in K$. The exponent vector $e = exp_{\mathcal{G}}(g)$ is a vector of depth i. By construction, $e \in rl(\overline{\mathcal{G}})$. Since the basis $b_1, \ldots, b_l$ is in upper triangular form, there exists an element of depth i in this basis, say b_j, and there exists an integer $a \in \mathbb{N}$ such that $e - ab_j$ has depth greater than i. Now we obtain that $k_j^{-a} \cdot g$ is an element of $ker(\varphi)$ which has depth greater than i. Thus, by induction, we obtain that $k_j^{-a} \cdot g \in K$ and therefore $g \in K$. Thus $K = ker(\varphi)$.

Let $G = G_1 \rhd \ldots \rhd G_n \rhd G_{n+1} = 1$ be the polycyclic series determined by $\mathcal{G}$. Then $K \cap G_i$ is the set of elements in K of depth at least i. Thus the above argument shows that $K_j = K \cap G_i$ where $K = K_1 \rhd \ldots \rhd K_l \rhd K_{l+1} = 1$ is the series determined by the sequence $\mathcal{K}$. Thus $\mathcal{K}$ determines a polycyclic series and therefore $\mathcal{K}$ is a polycyclic sequence. □

3.14. *Remark:*

a) We note that the sequence $\mathcal{K}$ obtained in Lemma 3.13 might contain elements k with $\text{relord}_{\mathcal{G}}(k) = 1$. Hence $\mathcal{K}$ is not necessarily induced with respect to $\mathcal{G}$.

b) Note that the given polycyclic sequence $\mathcal{G}$ for G in Lemma 3.13 need not be constructive.

Lemma 3.13 yields a method to compute a polycyclic sequence for the kernel into an additive abelian group using linear algebra methods. To apply this method we need to be able to compute a relation lattice for a generating set of the abelian group. This is straightforward if the image of the considered homomorphism φ is an explicitly given $\mathbb{Z}$-lattice or a vector space over a finite field.

Further references and comments

An important class of examples for polycyclic groups is given by free abelian and elementary abelian groups of finite rank. These correspond naturally to $\mathbb{Z}$-lattices and vector spaces over finite fields, respectively. In turn, for $\mathbb{Z}$-lattices and vector spaces over finite fields there are effective methods known to compute with sublattices or subspaces: one usually describes them using echelonized matrices and the Gauss elimination method or a Hermite normal form computation can be used to obtain such echelonized forms. The methods to compute induced or canonical polycyclic sequences as introduced in this chapter form natural extensions of these algorithms for $\mathbb{Z}$-lattices and vector spaces over finite fields.

Chapter 4

Polycyclic groups with finite action

Polycyclic groups often arise naturally as permutation or matrix groups. For example, the action of a polycyclic group on an elementary or free abelian normal subgroup yields a matrix representation of this group. In this chapter we introduce methods for polycyclic groups which induce a finite action on an underlying set.
In the first part of this chapter we describe an effective algorithm to determine finite orbits and their corresponding stabilizers under action of a polycyclic group. Our method applies to polycyclic groups which are given by polycyclic sequences and we assume that we know *a priori* that the considered orbit is finite. Our approach uses induction over a polycyclic sequence and exploits normal subgroups to build up orbits and stabilizers. The underlying observations on orbits and stabilizers of normal subgroups are interesting in their own right and they will be used in various places throughout this book.
Then we consider finite polycyclic permutation or matrix groups. If we want to compute with such groups using the algorithms developed in this book, then we need to be able to determine a constructive polycyclic sequence for them. In the second part of this chapter we describe an algorithm for this purpose. This method adapts ideas from permutation group theory and is based on the finite orbit stabilizer method introduced in the first part of the chapter. Additionally, the algorithm can be used to test if a given finite permutation or matrix group is polycyclic.

References: The finite orbit stabilizer algorithm for polycyclic groups in Section 4.2 and its underlying idea in Section 4.1 was introduced by Leedham-Green and is described and used in [38]. Other applications and extensions of this approach are discussed in [22].
The method to determine a constructive polycyclic sequence described in this chapter is based on the permutation group algorithm introduced by Sims [70]. It also uses ideas of an algorithm by Luks [47] to determine a constructive polycyclic sequence for matrix groups over finite fields. Various extensions of Sims method have been described in [75].

4.1 Normal subgroups and blocks

Let G be a group acting by multiplication from the right on an arbitrary set W and let $N \trianglelefteq G$. Let $w \in W$ and consider the orbit and stabilizer of w under the action of N. For $g \in G$ we have $(wN)g = (wg)N$ and, further, we either obtain $wN = (wg)N$ or $wN \cap (wg)N = \emptyset$. Thus wN is a block for the G-orbit of w and G acts on the set of N-orbits in W. We exploit this action to derive the G-orbit and stabilizer for $w \in W$.
Let $Stab_G(wN) = \{g \in G \mid (wN)g = wN\}$ the block stabilizer of wN in G. Since N acts trivially on the set of N-orbits in W, we obtain $N \leq Stab_G(wN) \leq G$. Let T be a transversal for $Stab_G(wN)$ in G

and let $\mathcal{E} \subset G$ such that $Stab_G(wN) = \langle \mathcal{E}, N \rangle$. Note that for $g \in Stab_G(wN)$ there exists an $n \in N$ with $wg = wn$ and thus $\tilde{g} = gn^{-1} \in Stab_G(w)$. Then

$$wG = \bigcup_{t \in T} (wN)t \quad \text{and} \quad Stab_G(w) = \langle \tilde{g} \mid g \in \mathcal{E} \rangle Stab_N(w).$$

If $Stab_G(wN)$ has finite index in G, then we can use these formulas to extend a solution for the orbit stabilizer problem in N to G as follows.

ExtendOrbitStabilizer(G, N, w)
  determine $Stab_G(wN) = \langle \mathcal{E}, N \rangle$
  compute a (finite) transversal T for $Stab_G(wN)$ in G
  construct wG as union of sets $(wN)t$ for $t \in T$
  obtain $Stab_G(w) = \langle \tilde{g} \mid g \in \mathcal{E} \rangle Stab_N(w)$

For this algorithm we need to be able to solve the orbit stabilizer problem for N. Further, in order to use this reduction method we need to find a finite generating set $\mathcal{E}$.

4.1. *Remark:* If N has finite index in G, then $Stab_G(wN)$ has finite index in G and thus the reduction can always be applied in this case.

4.2 Determining finite orbits

Let G be a finite polycyclic group acting by multiplication on the right on a set W and let $\mathcal{G} = (g_1, \ldots, g_n)$ is a polycyclic sequence for G with relative orders $(r_1, \ldots, r_n)$. We apply the method of Section 4.1 to determine the orbit and stabilizer of an element $w \in W$ under action of G; that is, we step up the polycyclic series $G_1, \ldots, G_n$ determined by $\mathcal{G}$ and construct orbits and stabilizers of w under G_i, successively.
In the iterative step from G_{i+1} to G_i we first determine the transversal T. We can choose $T = \{1, g_i, g_i^2, \ldots, g_i^l\}$ where l is the smallest index with $(wG_{i+1})g_i^{l+1} = wG_{i+1}$. Concurrently with the determination of T we obtain the enlarged orbit $wG_i = \bigcup_{j=0}^{l} (wG_{i+1})g_i^j$. Now we choose $\mathcal{E} = \{g_i^{l+1}\}$ if $l + 1 < r_i$ or $\mathcal{E} = \emptyset$ otherwise. Thus we build up $Stab_{G_i}(w)$ from $Stab_{G_{i+1}}(w)$. We summarize this approach in the following.

FiniteOrbitStabilizer$(G, \mathcal{G}, w)$
  initialize $\mathcal{O} = \{w\}$ and let H be the trivial subgroup of G
  (note that $\mathcal{O} = wG_{n+1}$ and $H = Stab_{G_{n+1}}(w)$)
  for i from n to 1 do
    compute $\mathcal{O}_j = \mathcal{O}g_i^j$ for $j \in \{1, 2, \ldots\}$ until $\mathcal{O}_j = \mathcal{O}$
    set $l = j - 1$ and reset $\mathcal{O} = \mathcal{O} \cup \mathcal{O}_1 \cup \ldots \cup \mathcal{O}_l$
    obtain $T = \{1, g_i, \ldots, g_i^l\}$
    read off $\mathcal{E} = \{g_i^{l+1}\}$ if $l + 1 < r_i$ or $\mathcal{E} = \emptyset$
    reset $H = \langle \tilde{g}, H \mid g \in \mathcal{E} \rangle$
    (note that $\mathcal{O} = wG_i$ and $H = Stab_{G_i}(w)$)
  end for
  return the orbit $\mathcal{O} = wG$ and the stabilizer $H = Stab_G(w)$

Note that it is sufficient to know $\mathcal{G}$ and its relative orders, but we need not assume that $\mathcal{G}$ is constructive. Further, our choice for the generators in $\mathcal{E}$ asserts that we obtain an induced polycyclic sequence for the stabilizer with respect to $\mathcal{G}$. The relative orders of the induced polycyclic sequence can also be read off.

References: The method described in this section also applies to polycyclic groups acting linearly on vector spaces over finite fields. A variety of improvements on stabilizer computations under this type of action have been introduced in a joint work with Leedham-Green and O'Brien. We omit details here and refer to [22].

4.3 Determining an abelian normal series

In this section we describe an approach to determine a normal series with abelian factors for a finite polycyclic group G. The approach uses induction and constructs an ascending series of this type. Further, we show that we can use such a series to determine a constructive polycyclic sequence for the underlying group G by extending a constructive polycyclic sequences upwards along the subgroups of such a series. The algorithm used to extend a constructive polycyclic sequence may depend on the representation in which G is given and we discuss approaches for this problem in more detail in Section 4.4.
We want to construct a normal series with abelian factors for the finite group G. By induction we assume that we have determined a normal subgroup N of G and we suppose that we can test membership in N. Our first aim is to determine an *abelian upwards extension* of N in G; that is, a normal subgroup H of G with $N < H$ and H/N abelian. Then, once generators for H are given, we extend the membership test from N to H and thus satisfy the induction hypothesis for the next inductive step.
We approximate an abelian upwards extension of N using a recursive method. At first, we choose an arbitrary element $g \in G \setminus N$; for example, we can take a generator of G which is not contained in N. We consider the normal closure $\langle g, N\rangle^G$. If this normal closure is an abelian upwards extensions of N, then the following algorithm recognizes this and constructs a set of generators $\mathcal{E}$ with $\langle g, N\rangle^G = \langle \mathcal{E}, N\rangle$. If the normal closure is an upwards extension with non-abelian factor, then the method returns a 'witness' for this fact. We analyze the algorithm in more detail below.

AbelianUpwardsExtension(G, N, g)
  initialize $\mathcal{E} = \{g\}$ and $H = \langle \mathcal{E}, N\rangle$
  while there is an unmarked element in $\mathcal{E}$ do
    choose an unmarked element $x \in \mathcal{E}$ and mark it
    let $\mathcal{C} = \{x^y \mid y \text{ generator of } G\}$
    if there is any $c \in \mathcal{C}$ with $[g, c] \notin N$, then return $[g, c]$
    add the elements of $\mathcal{C} \setminus \mathcal{E}$ to $\mathcal{E}$ and reset $H = \langle \mathcal{C}, H\rangle$
  end while
  return H

When applying this algorithm there are two possible outcomes:

Case 1: The algorithm returns a subgroup H. Clearly, H/N is non-trivial, since $N \neq gN \in H/N$. Further, $\mathcal{E}$ is a generating set for H modulo N. Since we closed this generating set under the action of G and we know that N is normal, we obtain that H is normal in G. Note that every element in $\mathcal{E}$ is conjugate to g under G. Two such conjugates commute modulo N if and only if g commutes with each element of $\mathcal{E}$ modulo N. Thus we obtain that H/N is abelian.

Case 2: The algorithm returns an element h. By the setup, $h \in G \setminus N$. Further, $h = [g, c]$ and c is conjugate to g in G. Hence, if g is an element of the i-th term of the derived series $G^{(i)}$ of G, then we observe that $c \in G^{(i)}$ and thus $h \in G^{(i+1)}$.

Thus in the first case we successfully determined an abelian upwards extension of N in G. In the second case we use the witness h for a recursive call of AbelianUpwardsExtension(G, N, h). The above analysis of the second case shows that if $g \in G^{(i)}$, then $h \in G^{(i+1)}$. Thus, since the derived series of the polycyclic group G has finite length, the recursion depth of an recursive application of this approach is finite.
We use the determination of abelian upwards extensions to compute constructive polycyclic sequences as outlined in the following algorithm. The algorithm takes as input a finite group G and an upper bound b for the derived length of G. It returns a constructive polycyclic sequence for G if G is polycyclic of derived length at most b. It returns fail if G is not polycyclic. We can allow $b = \infty$ as valid input if we know *a priori* that the given group G is polycyclic.

```
ConstructivePcSequence(G, b)
    initialize N = {1} ≤ G and c = 0
    initialize E as list of generators of G
    while E ≠ ∅ do
        if c > b then return fail (G is not polycyclic of derived length at most b)
        let h be the an element in E
        let H = AbelianUpwardsExtension(G, N, h)
        if H is a group then
            extend the constructive polycyclic sequence from N to H (Section 4.4)
            reset N = H and c = 0
            remove h from E
        elif H is an element then
            append H to the list E
            increase c = c + 1
        end if
    end while
    return N with its constructive polycyclic sequence
```

4.4 Extending constructive polycyclic sequences

Let $N \leq G$ and suppose that $\mathcal{N}$ is a constructive polycyclic sequence for N and we can test membership in the subgroup N of G. We consider an abelian upwards extension H of N in G which is given by a generating set $\mathcal{E} = \{h_1, \ldots, h_r\} \subseteq H$ such that $H = \langle \mathcal{E}, N \rangle$. We want to extend $\mathcal{N}$ to a constructive polycyclic sequence $\mathcal{H}$ for H and, similarly, we want to extend the membership test in N to a test in H. Both aims can be reached by a similar approach as follows.
First we reduce to the case that $\mathcal{E}$ contains a single element only. For this purpose we consider $H_i = \langle h_i, \ldots, h_r, N \rangle$. Since H/N is abelian, we obtain that $H_{i+1} \lhd H_i$. Further, $H_i = \langle h_i, H_{i+1} \rangle$ and H_i/H_{i+1} is cyclic. Now we use induction upwards the series $N \leq H_r \leq \ldots \leq H_1 = H$ and compute the desired information for each subgroup H_i successively.
Thus we assume that $H = \langle h, N \rangle$ for an element $h \in H$. Clearly, if $h \in N$, then $\mathcal{H} = \mathcal{N}$ is a constructive polycyclic sequence of H and we also obtain a membership test for H from N. Otherwise $\mathcal{H} = (h, \mathcal{N})$ is a polycyclic sequence of H. In the later case it remains to describe an algorithm to compute exponent vectors with respect to $\mathcal{H}$ and to test membership in H.
Each element $g \in H$ has a unique expression as $g = h^e \cdot n$ for $0 \leq e < [H : N]$ and $n \in N$. If we can determine e, then we can also obtain the exponent vector $(e, e_1, \ldots, e_r)$ for g with respect to $\mathcal{H}$, since we can compute the exponent vector $(e_1, \ldots, e_r)$ of n with respect to $\mathcal{N}$. Similarly, we can reduce the membership test of g in H to the determination of the exponent e, if it exists, and a membership test of n in N.

Thus it remains to determine the exponent e for a given element g if $g \in H$. Our method for this purpose may depend on the action of the given parent group G on the underlying set. We describe algorithms for this purpose in the following sections.

4.4.1 Permutation groups

Let G be a finite group acting faithfully on a set of points W. As above, let $N \lhd G$ and let H by a cyclic upwards extension of N in G such that $H = \langle h, N\rangle$ for $h \notin N$. Our aim is to find a method to determine the unique expression $g = h^e \cdot n$ for $0 \leq e < [H : N]$ and $n \in N$ if $g \in H$.
We consider a sequence of points $\mathcal{B} = (w_1, \ldots, w_t) \subseteq W$ and we define the corresponding sequence of stabilizers in N by $N_0 = N$ and $N_i = Stab_{N_{i-1}}(w_i)$. We call $\mathcal{B}$ a *base* for N if $N_t = 1$. We denote $\mathcal{O}_i = w_i N_{i-1}$ as the *basic orbits* corresponding to $\mathcal{B}$.
Since $N \lhd H$ we can use the ideas of the orbit stabilizer algorithms for finite polycyclic groups as described in Section 4.1 to extend the basic orbits and their stabilizers from N to H. In summary, since $H = \langle h, N\rangle$, we obtain that $w_1 H$ can be written as a disjoint union $w_1 H = \mathcal{O}_1 \cup \mathcal{O}_1 h \cup \ldots \cup \mathcal{O}_1 h^{l-1}$ for an integer $l \mid [H : N]$ with $\mathcal{O}_1 h^l = \mathcal{O}_1$. Thus $w_1 h^l = w_1 n$ for an element $n \in N$ and the stabilizer of w_1 in H can be read off as $H_1 = Stab_H(w_1) = \langle h^l n^{-1}, N_1\rangle$. Note that H_1 is a cyclic extension of N_1 and thus we can iterate this procedure with the remaining base points in $\mathcal{B}$.
Hence we can readily determine the basic orbits for $\mathcal{B}$ under H and the corresponding stabilizer sequence $H_1, \ldots, H_t$. By construction, H_t is a cyclic group. If $H_t = 1$, then $\mathcal{B}$ is a base for H as well. Otherwise we choose new base points $w_{t+1}, \ldots, w_{t+d}$ which are moved by the generator of H_t and extend $\mathcal{B}$ by these new base points to obtain a base for H. Here it is critical for the effectiveness of the approach to choose the new base points in such a form that their resulting basic orbits are small.
We can use the obtained base and stabilizer sequence for H to test membership in H by sifting an element through the chain. Further, the base and stabilizer sequence are adapted to the description of H by $\langle h, N\rangle$. In particular, for a given element $g \in H$ we can determine the exponent e with $g = h^e \cdot n$ by bookkeeping and sifting through the stabilizer sequence.

If we have an explicit finite set W on which G acts faithfully, then we obtain an upper bound for the derived length of polycyclic groups G by Theorem 4.2 below. Hence we can then also use the function 'ConstructivePcSequence' to test deterministically if a finite group G is polycyclic.

4.2. Theorem: (Dixon, Newman [52]) *Let G be a polycyclic subgroup of the symmetric group S_n. Then the derived length of G is at most $5 \log_9 n$.*

4.4.2 Matrix groups

Let $G \leq GL(d, K)$ be a finite polycyclic matrix group over a field K. Then G acts on the set of vectors in K^d and hence we can apply the approach of Section 4.4.1 to determine a constructive polycyclic sequence for G. We describe a number of refinements to this method in this section.
First, in the matrix group context there is a variety of other actions of G that we can exploit: for example, the action of G on subgroups of K^d or the action of G on vectors in E^d for an extension field $E > K$ may be useful if they yield base points with short orbits under G.
Suppose that N is a normal abelian subgroup of G such that N is diagonalizable over an extension field E of K. If we can determine such an extension E, then the vectors in E^d have short orbits under the action of N. Further, we can also use the diagonal form of N itself and arithmetic in E to obtain a constructive polycyclic sequence for N and a membership test.

Recall that in the overall outline of the algorithm as described in Section 4.3 we first determine an abelian normal subgroup N of the given group G. We can use this for a general reduction of the algorithm to determine a constructive polycyclic sequence of G as follows. We consider the matrix algebra $K(N)$ generated by the matrices in N and determine an K-basis for this algebra. The group G acts on $K(N)$ by conjugation and hence we obtain a new matrix representation for G over K in dimension $n = dim_K(K(N))$. Corresponding to this new action we obtain a homomorphism $\psi : G \to GL(n, K)$ such that $N \leq ker(\psi)$. We use ψ to break up the determination of a constructive polycyclic sequence for G by determining constructive polycyclic sequences for kernel and image of ψ and concatenating them to a sequence for G. In particular, if N is a diagonalizable group, then $n \leq d$ and thus we can hope to reduce the degree of the considered matrix representation.

4.3. *Remark:* Newman [52] has determined the maximal derived length of the soluble linear groups of degree d. The obtained bounds can be used in a similar form as in Theorem 4.2 to check if a given matrix group is polycyclic. The bounds are explicit, but complex to describe. Thus we refer to [52] for further information at this point.

Chapter 5

Polycyclic matrix groups

Auslander [1] and Swan [73] proved that every polycyclic group has a faithful integral matrix representation. Vice versa, such matrix representations arise naturally in the study of polycyclic groups: If G is given by a constructive polycyclic sequence, then an often used approach to compute information about G is to determine a normal series with elementary or free abelian factors of G and use induction over this series. The natural conjugation action of G on the factors of this series yields matrix representations of G over R where R is either a finite field or the integers.
The determination of a constructive polycyclic sequence for polycyclic subgroups of $GL(d, R)$ where R is either a finite field or the integers is an important issue for our algorithmic theory for polycyclic groups. In Section 4.4.2 we had solved this problem for the finite field case on R. In this chapter we investigate the remaining case $R = \mathbb{Z}$. More generally, we develop methods to determine a constructive polycyclic sequence for a polycyclic subgroup of $GL(d, \mathbb{Q})$.
To exhibit a constructive polycyclic sequence for a rational polycyclic matrix group we exploit the module structure of its natural module. In particular, we describe methods to determine the radical of this module and to split up the radical factor. These methods are of independent interest and we will use them later in algorithms for polycyclic groups.

References: Rational polycyclic matrix groups have been investigated by Wehrfritz in [76] and later by Dixon in [16]. Their results yield a structure theory for rational polycyclic matrix groups which has been fundamental for the algorithmic study of these groups. Many of the methods in this chapter are based on the approach in [16]. A similar approach to compute a constructive polycyclic sequence for polycyclic rational matrix groups has been introduced by Ostheimer in [59].
Various investigations of the natural module for a matrix group over a finite field can be performed using the 'Meataxe' methods. Their development has been initiated by Parker [60] and we refer to [48] or [34] for an outline of these algorithms. In particular, all problems considered here for rational modules can be solved effectively in the finite field context using the Meataxe.

5.1 Structure theory for rational polycyclic matrix groups

Let G be a polycyclic subgroup of $GL(d, \mathbb{Q})$. Then G is finitely generated and thus there exists a set of primes π such that $G \leq GL(d, \mathbb{Q}_\pi)$ where $\mathbb{Q}_\pi$ is the subring of $\mathbb{Q}$ consisting of those rational numbers $\frac{a}{b}$ where b is divisible by primes in π only. For example, we can take π as the set of all those primes which divide a denominator of a matrix entry in a generator or its inverse.

Let p be a prime with $p \notin \pi$ and consider the natural ring homomorphism $\iota : \mathbb{Q}_\pi \to \mathbb{F}_p$. This ring homomorphism extends to the *p-congruence homomorphism* of matrix groups by applying ι to each entry in a matrix:

$$\psi_p : GL(d, \mathbb{Q}_\pi) \to GL(d, p).$$

We call the kernel $K_p(G)$ of the restriction $\psi_p|_G$ of ψ_p to G the *p-congruence subgroup* of G and its image $I_p(G) = G^{\psi_p}$ is the *p-modular image*. We denote the primes p with $p \notin \pi$ as *admissible* primes for G. Clearly, the image $I_p(G)$ is a finite matrix group and thus the methods of Section 4.4.2 can be applied to determine a constructive polycyclic sequence for this image. The kernel $K_p(G)$ may be an infinite group. We can determine normal subgroup generators for $K_p(G)$ as described in Section 3.5.3.
The following theorem shows that the kernel $K_p(G)$ has an algorithmically useful structure if G is a polycyclic group. This theorem yields a structural setup for rational polycyclic matrix groups which is fundamental for our later investigations of such groups.

5.1. Theorem: (Dixon [16], Lemma 9) *Let G be a polycyclic subgroup of $GL(d, \mathbb{Q})$ and p an admissible prime for G. Then the p-congruence subgroup $K_p(G)$ is torsion-free and $K_p(G)'$ is unipotent.*

An immediate consequence of Theorem 5.1 is that the p-congruence subgroup $K_p(G)$ of a polycyclic group G is *triangularizable* over $\mathbb{C}$; that is, $K_p(G)$ is conjugate in $GL(d, \mathbb{C})$ to a group of upper triangular matrices. This is useful in theoretical arguments on $K_p(G)$, but for algorithmic methods we prefer to avoid computations over extension fields of $\mathbb{Q}$.
In the following lemma we recall some basic features of p-congruence subgroups. These features will be used throughout our investigations.

5.2. Lemma: *Let $G \leq GL(d, \mathbb{Q}_\pi)$ and $V = \mathbb{Q}^d$ the natural G-module.*

a) $K_p(G) = G \cap K_p(GL(d, \mathbb{Q}_\pi))$.

b) $K_p(G^g) = (K_p(G))^g$ for $g \in GL(d, \mathbb{Q}_\pi)$ and thus $K_p(G)$ is invariant under base changes.

c) Let W be a G-submodule of V. For the induced actions on W and V/W we obtain that $K_p(G)_W \leq K_p(G_W)$ and $K_p(G)_{V/W} \leq K_p(G_{V/W})$.

5.2 Radicals of rational modules

Let G be a polycyclic subgroup of $GL(d, \mathbb{Q})$ and let $V = \mathbb{Q}^d$ be the natural G-module. The radical $Rad_G(V)$ is defined as the intersection of all maximal G-submodules of V. Our aim in this section is to describe a practical method to determine a basis for $Rad_G(V)$.
For a subgroup $G \leq GL(d, \mathbb{Q})$ we denote with $\mathbb{Q}(G)$ the matrix subalgebra of $M^{dxd}(\mathbb{Q})$ generated by G. This is a finite dimensional G-module where G acts by multiplication from the right. Further, a G-module is called *semisimple* if it is a direct sum of irreducible G-modules. The following lemma gives a summary on the basic module theory used in this section. For more background we refer to [51].

5.3. Lemma: *Let $G \leq GL(d, \mathbb{Q})$ and $V = \mathbb{Q}^d$ the natural G-module.*

a) $\mathbb{Q}(G)$ is an Artinian algebra and V is an Artinian module.

b) $Rad_G(V) < V$ and $V/Rad_G(V)$ is semisimple.

c) $Rad_G(V) = V Rad_G(\mathbb{Q}(G))$.

d) If W is a G-submodule of $Rad_G(V)$, then $Rad_G(V/W) = Rad_G(V)/W$.

5.4. *Remark:* A well-known method that we use throughout is the *spinning algorithm*. This algorithm determines a basis for a finite dimensional module which is given by module generators. It approaches this problem by successively producing new basis elements for the module until a basis is found which generates an invariant subspace. For more details we refer to [47].

5.2.1 Determining the radical for an abelian group

In this section we describe the determination of the radical of a natural module under action of an abelian group. The special case of an abelian acting group will form the key part for the general algorithm for polycyclic groups. The following lemma yields a criterion for the radical of an abelian group algebra. We denote a matrix or a matrix group as *diagonalizable* if it is conjugate in $GL(d,\mathbb{C})$ to a diagonal matrix or an diagonal matrix group.

5.5. Lemma: *Let G be an abelian subgroup of $GL(d,\mathbb{Q})$ and let $\mathcal{B}$ be a basis of $\mathbb{Q}(G)$.*

a) $Rad_G(\mathbb{Q}(G)) = 0$ if and only if each element in $\mathcal{B}$ is diagonalizable.

b) Let $g \in \mathbb{Q}(G)$ with minimal polynomial $f(x) \in \mathbb{Q}[x]$ and let $f(x) = f_1(x)^{e_1} \cdots f_r(x)^{e_r}$ be its factorization into irreducible factors. Then g is diagonalizable if and only if $e_1 = \ldots = e_r = 1$. If g is not diagonalizable, then $f_1(g) \cdots f_r(g)$ is a non-trivial element of $Rad_G(\mathbb{Q}(G))$.

Proof. For a) we note that if each basis element of an abelian algebra is diagonalizable, then the basis elements are also simultaneously diagonalizable by Schur's lemma. Thus each element of $\mathbb{Q}(G)$ is diagonalizable in this case and hence $\mathbb{Q}(G)$ is semisimple. The converse is obvious. For b) we recall that the radical of an abelian Artinian algebra is the set of its nilpotent elements. If there exists an index i with $e_i \neq 1$, then $f_1(g) \cdots f_r(g)$ is a non-trivial nilpotent element. Hence b) follows. □

Let G be an abelian subgroup of $GL(d,\mathbb{Q})$ and $V = \mathbb{Q}^d$ its natural module. Lemma 5.5 can be translated into the following method to determine the radical $Rad_G(V)$.
We start by building up a basis for $\mathbb{Q}(G)$ using the spinning algorithm. Whenever we add an element g to the basis, we check if g is diagonalizable using Lemma 5.5 b). If g is not diagonalizable, then we obtain a non-trivial nilpotent element $h \in \mathbb{Q}(G)$ and the G-submodule Vh spanned satisfies $Vh \leq Rad_G(V)$ by Lemma 5.3 c). In this case we then proceed by recursing to V/Vh. If g is diagonalizable, then we continue the computation. This approach eventually produces a diagonalizable basis $\mathcal{B}$ in which case we are finished by Lemma 5.5 a). Note that the constructed partial basis of $\mathbb{Q}(G)$ always consists of diagonalizable elements and hence has length at most d.

5.2.2 Determining the radical for a polycyclic group

In this section we outline a method to determine $Rad_G(V)$ where G is a polycyclic subgroup of $GL(d,\mathbb{Q})$ and V is its natural module. Our overall approach will be to use the structure theory for such groups G introduced in Section 5.1 to reduce the problem to the case solved in Section 5.2.1.
The following theorem yields an important tool to determine $Rad_G(V)$. It can be considered as an extended version of Maschke's theorem on finite rational matrix groups. We refer to [76], 1.5 and 1.8, for a proof.

5.6. Theorem: *Let $G \leq GL(d,\mathbb{Q})$ and $N \trianglelefteq G$ with $[G:N] < \infty$. Let $V = \mathbb{Q}^n$ be the natural G-module. Then $Rad_G(V) = Rad_N(V)$.*

Let p be an admissible prime for G. Then, by Theorem 5.6, $Rad_G(V) = Rad_{K_p(G)}(V)$ for the p-congruence subgroup $K_p(G)$ of G. However, $K_p(G)$ is triangularizable over $\mathbb{C}$ as noted in Section 5.1 and this simplifies the determination of the corresponding radical significantly as we show in the following lemma.

5.7. Lemma: *Let $G \leq GL(d,\mathbb{Q})$ be triangularizable. Then $V(g-1) \subseteq Rad_G(V)$ for each $g \in G'$.*

Proof. G' consists of unipotent elements and hence, since each $x \in G$ is triangularizable, $x(g-1)$ is nilpotent for $x \in G$ and $g \in G'$. Thus $(g-1) \in Rad_G(\mathbb{Q}(G))$ and we obtain $V(g-1) \subseteq Rad_G(V)$ by Lemma 5.3 c). □

We obtain the following algorithm to determine $Rad_{K_p(G)}(V)$. We consider a generating set $\mathcal{E}$ for $K_p(G)$ and we define W as the $K_p(G)$-submodule of V generated by $\{V([g,h]-1) \mid g,h \in \mathcal{E}\}$. By Lemma 5.7, this yields $W \leq Rad_{K_p(G)}(V)$. Further, $K_p(G)$ acts as abelian group on V/W by construction. Thus we can now determine $Rad_{K_p(G)}(V)/W = Rad_{K_p(G)}(V/W)$ using the algorithm of Section 5.2.1.

5.2.3 A summary of the radical algorithm

We summarize the resulting algorithm to compute the radical $Rad_G(V)$ for a polycyclic rational matrix group $G \leq GL(d,\mathbb{Q})$ acting on its natural module as follows. We first outline a version of the algorithm which assumes that we can determine a generating set for a p-congruence subgroup. We discuss a generalization of this approach below.

```
Radical(G)
  let V = Q^d be the natural G-module
  choose an admissible prime p and determine K_p(G) with generators E
  compute the G-submodule W of V generated by V([g,h]-1) for g,h in E
  induce the matrix action of G to V/W
  start spinning up a basis B for Q(K_p(G)) acting on V/W:
    for each new basis element g of Q(K_p(G)) do
      determine the minimal polynomial f(x) in Q[x] for g
      factorize f(x) over Q
      if f(x) is not square-free then
        determine a non-trivial element h in Rad_G(Q(K_p(G))) using Lemma 5.5
        spin up the G-module U/W of V/W generated by (V/W)h
        reset W = U and and induce the action of G to V/W
      end if
    end for
  return the radical W
```

This approach to compute $Rad_G(V)$ needs to determine a generating set for the subgroup $K_p(G)$ of G. However, as described in Section 5.1 we may have normal subgroup generators for $K_p(G)$ only. To cover this case as well, we can modify the above approach to determine the radical as follows. In spinning up the basis $\mathcal{B}$ we additionally close this basis of $\mathbb{Q}(K_p(G))$ under the conjugation action of G. Whenever we add a new element to the partial basis $\mathcal{B}$, we additionally check if it commutes with all elements in $\mathcal{B}$. If not, then we enlarge the determined subspace W as in Lemma 5.7.

References: An alternative approach to determine the radical of an arbitrary rational algebra has been introduced by Dickson in [14]. This approach relies on the determination of a $\mathbb{Q}$-basis for the algebra $\mathbb{Q}(G)$. Since the dimension of this algebra can be as large as d^2, this approach is often less practical than our method.

5.3 Semisimple rational matrix groups

In this section we consider a polycyclic subgroup G of $GL(d,\mathbb{Q})$ such that $Rad_G(V) = 0$ for the natural G-module $V = \mathbb{Q}^d$. Then V is a semisimple G-module and we denote G as semisimple matrix group. In the following corollary of Theorem 5.1 and Lemma 5.7 we investigate the structure and the action of the p-congruence subgroup $K_p(G)$ of G in this case.

5.8. Corollary: *Let $G \leq GL(d,\mathbb{Q})$ be a polycyclic semisimple matrix group and let p be an admissible prime for G. Then $K_p(G)$ is free abelian and the natural module $V = \mathbb{Q}^d$ is semisimple as $K_p(G)$-module.*

Hence the natural G-module V is a direct sum of irreducible $K_p(G)$-submodules. Our aim in this section is to describe an algorithm which exhibits such a direct splitting of V. This direct splitting into irreducible $K_p(G)$-submodules will be used in later applications for polycyclic rational matrix groups.
By Corollary 5.8, the subgroup $K_p(G)$ is a finitely generated abelian semisimple matrix group. In the following section we describe a method to split the natural module of such a group into a direct sum of irreducibles.

5.3.1 Abelian semisimple rational matrix groups

Let G be a finitely generated abelian subgroup of $GL(d,\mathbb{Q})$ and let $V = \mathbb{Q}^d$ be its natural module. We assume that V is semisimple as G-module and we describe an algorithm to determine a direct factorization of V into irreducible G-submodules. Abelian semisimple matrix groups are diagonalizable. Accordingly, their irreducible submodules over $\mathbb{C}$ all have dimension one. This feature also yields that a splitting for such a group over $\mathbb{Q}$ is not difficult to obtain. As a first step towards such a splitting we show that the matrix algebras for the considered groups are generated by a single element and that such a generator can be determined easily.

5.9. Lemma: *Let G be a finitely generated subgroup of $GL(d,\mathbb{Q})$ such that $\mathbb{Q}(G)$ is an abelian semisimple algebra with $\dim_{\mathbb{Q}} \mathbb{Q}(G) = s$. Then we have the following.*

a) If $\mathbb{Q}(G)$ contains an element c whose minimal polynomial has degree s, then $\mathbb{Q}(G) = \mathbb{Q}(c)$.

b) Almost all linear combinations of generators of G yield an element $c \in \mathbb{Q}(G)$ with $\mathbb{Q}(G) = \mathbb{Q}(c)$.

Proof. a) is well-known and it remains to show b). We consider two elements $x, y \in \mathbb{Q}(G)$. Clearly, $\mathbb{Q}(x + ay) \subseteq \mathbb{Q}(x, y)$ for any scalar $a \in \mathbb{Q}$. We show in the following that for almost all scalars $a \in \mathbb{Q}$ we obtain $\mathbb{Q}(x + ay) = \mathbb{Q}(x, y)$. This yields part b) by induction on a generating set of G.
We have $\mathbb{Q}(x + ay) = \mathbb{Q}(x, y)$ if and only if $x, y \in \mathbb{Q}(x + ay)$; that is, if $x = f(x + ay)$ and $y = g(x + ay)$ for two rational polynomials f and g. Recall that $\mathbb{Q}(G)$ is diagonalizable over $\mathbb{C}$ and consider a diagonal form of this algebra. Let $x_1, \ldots, x_d$ and $y_1, \ldots, y_d$ be the diagonal elements of x and y, respectively. Then $x + ay$ has the diagonal elements $x_i + ay_i$ for $1 \leq i \leq d$. Hence $x = f(x + ay)$ if and only if $x_i = f(x_i + ay_i)$ for $1 \leq i \leq d$. A complex polynomial with these conditions can be determined using interpolation whenever $x_i + ay_i \neq x_j + ay_j$ implies $x_i \neq x_j$. However, there are at most $d(d-1)/2$ elements a which violate this condition. Further, if a satisfies the condition, then $x, y \in \mathbb{C}(x + ay)$ and thus $x, y \in \mathbb{Q}(x + ay)$, since $a \in \mathbb{Q}$. □

By Lemma 5.9 it is straightforward to determine a generator c for the semisimple abelian matrix algebra $\mathbb{Q}(G)$. We first determine the dimension $\dim_{\mathbb{Q}}(\mathbb{Q}(G))$ by spinning up a basis of $\mathbb{Q}(G)$. Note that in our case we have $\dim_{\mathbb{Q}}(\mathbb{Q}(G)) \leq d$ for $G \leq GL(d,\mathbb{Q})$ and thus the determination of a basis for $\mathbb{Q}(G)$ is practical. Then we search randomly for a linear combination c of the generators of G which has a minimal polynomial of degree $\dim_{\mathbb{Q}}(\mathbb{Q}(G))$. In practice, we usually find that most of the generators themselves have such a minimal polynomial.

Further, once a generator c for $\mathbb{Q}(G)$ is given, it remains to determine a c-invariant direct splitting of the underlying natural module into irreducibles. It is well-known from linear algebra that such a splitting can be read off from the minimal polynomial of c. We recall this feature in the following two lemmas.

5.10. Lemma: *Let $c \in GL(d,\mathbb{Q})$ be diagonalizable and let $f(x) \in \mathbb{Q}[x]$ be its minimal polynomial. We consider the factorization $f(x) = f_1(x)\cdots f_r(x)$ into irreducible polynomials. We denote $V = \mathbb{Q}^d$ and define $W_i = ker(f_i(c)) \leq V$.*

a) $V = W_1 \oplus \ldots \oplus W_r$ is a direct $\mathbb{Q}(c)$-invariant factorization of V.

b) Each submodule W_i is a homogeneous $\mathbb{Q}(c)$-module; that is, it is a direct sum of isomorphic irreducible $\mathbb{Q}(c)$-submodules.

Proof. The subspaces W_i are invariant under $\mathbb{Q}(c)$, since they are kernels of elements in the matrix algebra $\mathbb{Q}(c)$. Further, the polynomials $f_1,\ldots,f_r$ are pairwise coprime which yields that the subspaces W_i generate $\mathbb{Q}^d$ as direct sum. Finally, the algebra induced by $\mathbb{Q}(c)$ on W_i is irreducible and thus W_i is homogeneous. □

Using Lemma 5.10 we can split a natural module for a finitely generated abelian semisimple matrix group into homogeneous submodules. It remains to refine the homogeneous components into irreducibles. This is obtained by the following lemma.

5.11. Lemma: *Let G be an abelian subgroup of $GL(d,\mathbb{Q})$ and let $W \leq V$ be a homogeneous submodule of the natural module V. Then each non-trivial vector of W is contained in an irreducible G-submodule of W.*

Proof. Let $W = W_1 \oplus \ldots \oplus W_r$ be a direct factorization of W into isomorphic irreducible G-submodules and let $0 \neq w \in W$. We write $w = w_1 + \ldots + w_r$ with $w_i \in W_i$. If $w \neq 0$, then there exists a component i with $w_i \neq 0$. Let U be the submodule of W generated by w and let $\psi : U \to W_i$ be the projection onto the i-th component of W. Since $w_i \neq 0$, we obtain that ψ is surjective. It remains to show that ψ is injective. Let $u \in ker(\psi)$. By construction, $u = wh$ for an element $h \in \mathbb{Q}(G)$ and thus $u_i = (wh)_i = w_i h = 0$. By Schur's lemma we obtain that $h_{W_i} = 0$. Hence $h = 0$ and $u = wh = 0$ as desired. □

The Lemmas 5.9, 5.10 and 5.11 can be translated into the following algorithm to determine a direct factorization of the natural G-module V for a finitely generated abelian semisimple subgroup of $GL(d,\mathbb{Q})$.

```
DirectSplitting(G)
    assume that G is a finitely generated abelian semisimple subgroup of GL(d,Q)
    let V = Q^d be the natural G-module
    determine a basis B of Q(G) and thus find s = dim_Q(Q(G))
    find an element c ∈ Q(G) with minimal polynomial f(x) of degree s
    factorize f(x) = f_1(x)···f_r(x) over Q
    determine W_i = ker(f_i(c)) < V for 1 ≤ i ≤ r
    for each W_i do
        split W_i into irreducibles W_{i,1} ⊕ ... ⊕ W_{i,r_i} by spinning up vectors
    end for
    return the set of direct factors {W_{i,j} | 1 ≤ i ≤ r, 1 ≤ j ≤ r_i}
```

5.4 Module series and induced actions

In this section we provide a summary for the submodule series which can be obtained by the methods in the above sections. We consider the action of the underlying group on the factors of such series and thus we provide the basis for later algorithms which use induction over a series of this kind. Our investigations in this section are based on a well-known relation between rational subspaces and sublattices which we recall in the following.

5.4.1 Rational spaces and lattices

For two vectors x and w in $\mathbb{Q}^d$ we denote with xw the standard inner product of the two vectors. For any subset W of $\mathbb{Q}^d$ we define $W^* = \{x \in \mathbb{Z}^d \mid xw = 0 \text{ for each } w \in W\}$. The set W^* is a sublattice of $\mathbb{Z}^d$ which is called *dual lattice* to W. If W is a subspace of $\mathbb{Q}^d$ or a sublattice of $\mathbb{Z}^d$ and W is given by a basis, then a basis of W^* can be determined effectively by solving a homogeneous system of linear equations over $\mathbb{Z}$.
We call a sublattice L of $\mathbb{Z}^d$ *pure* if $\mathbb{Z}^d/L$ is torsion-free. For an arbitrary sublattice L in $\mathbb{Z}^d$ we define its *pure hull* $\overline{L}$ as the preimage of the torsion subgroup of $\mathbb{Z}^d/L$; that is, $\overline{L}/L = T(\mathbb{Z}^d/L)$. Clearly, the pure hull of sublattice is a pure sublattice of $\mathbb{Z}^d$. The importance of pure sublattices for our purposes is that there exists a basis of $\mathbb{Z}^d$ through a sublattice if and only if the sublattice is pure.

5.12. Lemma:

a) Let $W \leq \mathbb{Q}^d$. Then $W^{**} \subseteq W$ and $\dim_{\mathbb{Q}} W^{**} \otimes \mathbb{Q} = \dim_{\mathbb{Q}} W$.

b) Let $L \leq \mathbb{Z}^d$. Then $L \leq L^{**}$ and L^{**} is the pure hull of L.

Proof. a) It is straightforward to observe that $W^{**} = W \cap \mathbb{Z}^d$ and, further, the $\mathbb{Q}$-dimensions of W and W^{**} are equal. b) Let v be an element of the pure hull of L. Then $ev \in L$ for some $e \in \mathbb{N}$. Thus for each $w \in L^*$ we have $evw = 0$. Hence $vw = 0$ for each $w \in L^*$ and $v \in L^{**}$. Therefore L^{**} contains the pure hull. On the other hand the $\mathbb{Q}$-dimensions of L and L^{**} are equal and hence L^{**}/L is a torsion group. Thus we obtain that L^{**} is contained in the pure hull of L. □

5.13. Corollary: *Let $W \leq \mathbb{Q}^d$. Then $W^{**} \leq \mathbb{Z}^d$ and there exists a basis $\mathcal{B}$ of $\mathbb{Z}^d$ through W^{**}. Further, $\mathcal{B}$ is an integral basis of $\mathbb{Q}^d$ through W.*

5.4.2 Induced actions on factor modules

Let G be a subgroup of $GL(d, \mathbb{Q}_\pi)$ and V the natural G-module. Let $V = V_1 > \ldots > V_m > V_{m+1} = 0$ be a series of G-submodules in V. We consider the action homomorphism of G corresponding to this series

$$\nu : G \to G_{V_1/V_2} \times \ldots \times G_{V_m/V_{m+1}}.$$

For induction purposes it will be useful to observe that also the kernel and the image of the action homomorphism ν can be written over the same ring $\mathbb{Q}_\pi$ and do not require a larger subring.

5.14. Lemma: *Suppose that $G \leq GL(d, \mathbb{Q}_\pi)$ for a set of primes π. Then $ker(\nu)$ is a unipotent subgroup of $GL(d, \mathbb{Q}_\pi)$ and $G_{V_i/V_{i+1}} \leq GL(d_i, \mathbb{Q}_\pi)$ for $1 \leq i \leq m$.*

Proof. This is obviously true for $ker(\nu)$ and we only consider $im(\nu)$. By Lemma 5.13 we can determine an integral basis $\mathcal{B}$ of $\mathbb{Q}^d$ through the series $V_1, \ldots, V_{m+1}$. Let $g \in GL(d, \mathbb{Z})$ be the base change matrix corresponding to $\mathcal{B}$. Then $G^g \leq GL(d, \mathbb{Q}_\pi)$ and the elements of G^g are in block upper triangular form. Further, the induced actions on V_i/V_{i+1} can be read off from G^g as the blocks on the diagonals. □

5.4.3 Submodule series

Let G be a polycyclic subgroup of $GL(d,\mathbb{Q})$ and $V = \mathbb{Q}^d$. The radical series $V = R_1 > \ldots > R_l > R_{l+1} = 0$ is defined by $R_{i+1} = Rad_G(R_i)$. This series can be determined using an iterated application of the algorithm in Section 5.2.2.

RadicalSeries(G)
  let $V = \mathbb{Q}^d$ be the natural G-module
  initialize $\mathcal{L} = \{V\}$
  while $\dim(V) > 0$ do
    determine $V = Rad_G(V)$ (Section 5.2.2)
    add V to $\mathcal{L}$
    induce the action of G to an action on V
  end while
  return the series $\mathcal{L}$

Let p be an admissible prime and consider the p-congruence subgroup $K_p(G)$. In our later applications of rational matrix groups to algorithms for polycyclic groups we will use module composition series in V under the action of $K_p(G)$; that is, a sequence of $K_p(G)$-submodules $V = V_1 > \ldots > V_m > V_{m+1} = 0$ such that V_i/V_{i+1} is an irreducible module for $\mathbb{Q}(K_p(G))$. We can determine such a series using the algorithms of the above sections as follows.

CompositionSeries(G)
  assume that G is a polycyclic subgroup of $K_p(GL(d,\mathbb{Q}_\pi))$
  compute the radical series $V = R_1 > \ldots > R_l > R_{l+1} = 0$ (using RadicalSeries(G))
  for each i in $\{1,\ldots,l\}$ do
    split R_i/R_{i+1} into a direct sum of irreducibles (Section 5.3.1)
    use this direct splitting to refine the factor R_i/R_{i+1}
  end for
  return the resulting refined series

5.15. *Remark:* Let $G \leq K_p(GL(d,\mathbb{Z}))$ and consider the natural action of G on the free abelian group $V = \mathbb{Z}^d$. Using the algorithm 'CompositionSeries' and Corollary 5.13 we can determine a series of pure subgroups $V = V_1 > \ldots > V_m > V_{m+1} = 0$ such that V_i/V_{i+1} is irreducible as $\mathbb{Q}(G)$-module. Such a series is called *irreducible block flag* for G and it will be used in later applications such as the orbit stabilizer algorithm of Chapter 7.

5.5 Constructive polycyclic sequences in rational matrix groups

Let G be a subgroup of $GL(d,\mathbb{Q})$ given by a generating set. Suppose we know that G is polycyclic and we want to compute with G using the methods introduced in this book. Then we need to determine a constructive polycyclic sequence for G as a first step. In this section we introduce an algorithm for this purpose.
In our approach we use a p-congruence subgroup $K_p(G)$ and the method of Section 5.4.3 to compute the radical series $V = R_1 > \ldots R_l > R_{l+1}$ of G. Then we exploit the corresponding action homomorphism $\nu : G \to G_{R_1/R_2} \times \ldots \times G_{R_l/R_{l+1}}$ as outlined in Section 5.4.2. Define $U = ker(\nu|_{K_p(G)})$ the kernel of ν restricted to $K_p(G)$. We consider the normal series

$$1 \trianglelefteq U \trianglelefteq K_p(G) \trianglelefteq G$$

and we determine a constructive polycyclic sequence for G by combining constructive polycyclic sequences for $G/K_p(G)$, $K_p(G)/U$ and U. The first factor $G/K_p(G)$ embeds into $GL(d,p)$ and thus we can use the method of Section 4.4.2 for this factor. The next factor $K_p(G)/U$ acts faithfully as free abelian semisimple matrix group on the direct sum of factors R_i/R_{i+1} by Corollary 5.8. We describe a method for this case in Section 5.5.1. Finally, U is unipotent and we consider this case in Section 5.5.2. We summarize the resulting algorithm in Section 5.5.3.

We note that we determine $K_p(G)$ and U by computing a constructive polycyclic sequence for the corresponding factor $G/K_p(G)$ or G/U, evaluating the relations of the corresponding polycyclic presentation and enlarging the resulting normal subgroup generating set to a generating set as in Section 3.3. For this purpose we need to have method to determine constructive polycyclic sequences for subgroups of the considered groups and a membership test for subgroups. Both will be obtained by the algorithms described in the following two sections.

5.5.1 Abelian semisimple rational matrix groups

Let $G \leq GL(d,\mathbb{Q})$ be a free abelian group such that $\mathbb{Q}(G)$ is a semisimple algebra and let $\mathcal{E} = \{h_1,\ldots,h_l\}$ be a set of generators for G. We want to determine a constructive polycyclic sequence $\mathcal{G}$ for G and describe a membership test for G. First we note that the generators in $\mathcal{E}$ form a polycyclic sequence for G, since G is abelian. However, the sequence might not be constructive. Hence we have to investigate the situation in more detail.

The relations for the generators in $\mathcal{E}$ correspond to the *relation lattice* $rl(\mathcal{E}) = \{(e_1,\ldots,e_l) \in \mathbb{Z}^l \mid h_1^{e_1}\cdots h_l^{e_l} = 1\}$. If generators for the subgroup $rl(\mathcal{E})$ of $\mathbb{Z}^l$ are given, then we can apply the Smith normal form algorithm to $rl(\mathcal{E})$ and derive a multiplicatively independent set of generators $\mathcal{G}$ for G.

If $\mathcal{G}$ is a multiplicative independent set of generators for G, then $\mathcal{G}$ is clearly also a polycyclic sequence for G. The relative orders for $\mathcal{G}$ are all infinite, since G is torsion-free. Further, if we have an algorithm to determine the relation lattice for an arbitrary generating set of G, then the sequence $\mathcal{G}$ can be made constructive as follows. Suppose that $g \in G$ is given. Then we apply the relation lattice algorithm to the generating set $\{\mathcal{G}, g\}$ of G. Since $g \in G$, this yields a relation of the form $(e_1,\ldots,e_n,1)$ and we can read off $\exp_{\mathcal{G}}(g) = (-e_1,\ldots,-e_n)$. Similarly, we can use this approach to test membership in G.

Hence it remains to describe an algorithm to determine the relation lattice for an arbitrary generating set of G. We consider two approaches for this purpose. The first method is using additive valuations of extension fields of $\mathbb{Q}$. It turns out that this problem can easily be reduced to a well-known number theoretic question and number theory provides an algorithmic solution for this problem. The second algorithm is a group theoretic approach to this problem. It has the advantage that it avoids computations over extension fields of $\mathbb{Q}$ and uses rational arithmetic only.

Additive valuations

Let G be a torsion-free abelian semisimple subgroup of $GL(d,\mathbb{Q})$ generated by $\mathcal{E}$. By Lemma 5.9 we can find a generator c for the algebra $\mathbb{Q}(G) = \mathbb{Q}(c)$. Let f be the minimal polynomial of c and recall that f is square-free by Lemma 5.5. Thus $\mathbb{Q}(c) = \mathbb{Q}[x]/(f)$ is a direct sum of fields.

Let $\lambda : \mathbb{Q}(c) \to \mathbb{R} \cup \{\infty\}$ be an additive valuation for $\mathbb{Q}(c)$. Then the relation lattice $rl(\mathcal{E})$ is contained in the integral kernel of the linear form $x_1\lambda(h_1) + \ldots + x_l\lambda(h_l)$. If we choose λ such that $\lambda(h) = 0$ if and only if $h = 1$, then $rl(\mathcal{E})$ is equal to the kernel of the above linear equation.

It remains to determine a suitable additive valuation of $\mathbb{Q}(c)$. There are several choices possible for this purpose. For example, the extension of the natural logarithm to $\mathbb{Q}(c)$ can be considered or we can extend p-adic valuations of $\mathbb{Q}$ to $\mathbb{Q}(c)$. We omit details on this number theoretic problem here and refer to [63] or [10] for further information.

A finiteness condition and the Dixon bound

Let $G \leq GL(d,\mathbb{Q})$ be a torsion-free abelian semisimple group generated by the set $\mathcal{E} = \{h_1,\ldots,h_l\}$. In this section we observe that the the relation finding problem for $\mathcal{E}$ is a finite problem. For this purpose we first introduce the *Dixon bound* and we show that this bound yields a condition of the desired type. We consider additive valuations of $\mathbb{Q}$. Let λ_p be the normalized p-adic valuation of $\mathbb{Q}$ and let λ be the negative natural logarithm of the absolute value. Let f be the minimal polynomial of c with degree s and coefficients $f_0,\ldots,f_s$. We define

$$b = l(s-1)\cdot \max_{1\leq i\leq s} |\lambda(1+|f_i|)| \quad \text{and} \quad b_p = l(s-1)\cdot \max\{\lambda_p(f_0),\lambda_p(f_s)\}.$$

Further, let $u_i, v_i \in \mathbb{Q}[x]$ such that $u_i(c) = h_i$ and $v_i(c) = h_i^{-1}$. Note that these polynomials have degree at most s and hence we may denote their coefficients by $u_{i,0},\ldots,u_{i,s}$ and $v_{i,0},\ldots,v_{i,s}$. Then we consider

$$\begin{aligned} a &= \sum_{i=1}^{l} \max\{|\lambda(|u_{i,0}|+\cdots+|u_{i,s}|)|, |\lambda(|v_{i,0}|+\cdots+|v_{i,s}|)|\} \\ a_p &= \sum_{i=1}^{l} \max\{\min_{1\leq j\leq s} \lambda_p(u_{i,j}), \min_{1\leq j\leq s} \lambda_p(v_{i,j})\} \end{aligned}$$

There are only finitely many primes p such that b_p or a_p are non-trivial. Now we obtain the Dixon bound B as follows.

$$B = B(\mathcal{E},c) = (C^l+1)/C \quad \text{with}$$

$$C = \max\{\frac{7s^2}{\log s}(b+a), s(b_p+a_p) \mid \text{ for all primes } p\}.$$

Although we have to consider additive valuations as in the section above, we only need valuations of $\mathbb{Q}$ and we only need to find bounds for certain values of these valuations. Hence we can compute the Dixon bound using rational arithmetic only.

5.16. Theorem: (Dixon, [16]) *Let $G = \langle \mathcal{E}\rangle$ an abelian, torsion-free, integral matrix group such that $\mathbb{Q}(G) = \mathbb{Q}(c)$ for an element $c \in \mathbb{Q}(G)$. Let $B = B(\mathcal{E},c)$ be the corresponding Dixon bound. Then $rl(\mathcal{E}) = S^{**}$ with $S = \{(e_1,\ldots,e_l) \in rl(\mathcal{E}) \mid |e_i| \leq B \text{ for } 1\leq i\leq l\}$.*

Since we can easily check if a given vector $e \in \mathbb{Z}^l$ is a relation for $\mathcal{E}$, we obtain by Theorem 5.16 that the problem of finding the relation lattice is a finite problem. However, the Dixon bound is exponential in the number of given generators of G and hence it might not be practical to check all B^l necessary vectors for relations. The method described in the following section yields a reduction to this problem.

Finite approximation of the relation lattice

Let G be a free abelian semisimple subgroup of $GL(d,\mathbb{Q})$ generated by $\mathcal{E} = \{h_1,\ldots,h_l\}$. As outlined in Section 5.1 there exist infinitely many admissible primes p for G. We consider the corresponding congruence homomorphisms $\psi_p : G \to GL(d,p)$. For each prime p the image of ψ_p is a finite abelian subgroup of $GL(d,p)$. Using the method of Section 4.4.2 we can determine the relations of the images of the generators $\mathcal{E}$ for G. These relations can be pulled back to $\mathbb{Z}^l$ and thus we can effectively determine a basis of the *p-adic relation lattice* $rl_p(\mathcal{E}) = \{e \in \mathbb{Z}^l \mid (h_1^{\psi_p})^{e_1}\cdots(h_l^{\psi_p})^{e_l} = 1\}$. For a set of primes π we denote $rl_\pi(\mathcal{E}) = \cap_{p\in\pi} rl_p(\mathcal{E})$. For any finite set of primes π a basis for $rl_\pi(\mathcal{E})$ can be computed from bases of $rl_p(\mathcal{E})$ using a Hermite normal form algorithm.

5.17. Lemma: *Let G be a free abelian integral matrix group with generators $\mathcal{E}$. Then $rl_\Pi(\mathcal{E}) = rl(\mathcal{E})$ where Π is the set of all admissible primes and $rk(rl_\pi(\mathcal{E})) = |\mathcal{E}|$ for any finite set of admissible primes π.*

Proof. Since ψ_p is a homomorphism, $rl(\mathcal{E}) \leq rl_p(\mathcal{E})$ and thus $rl(\mathcal{E}) \leq rl_\Pi(\mathcal{E})$. Conversely, with $l = |\mathcal{E}|$ we have that for each $e \in \mathbb{Z}^l \setminus rl(\mathcal{E})$ there exists an admissible prime p with $e \notin rl_p(\mathcal{E})$. Further, each element $h_i^{\psi_p}$ has finite order. Therefore we obtain that $rl_\pi(\mathcal{E})$ has full rank for a finite set of primes π. □

We use p-adic relation lattices in combination with the LLL-algorithm to determine relations of $\mathcal{E}$. Experimental evidence shows that it is a good heuristic approach to first consider a small set of admissible primes π for G and determine $rl_\pi(\mathcal{E})$. Now we apply the LLL-algorithm to a basis of $rl_\pi(\mathcal{E})$ and thus find a basis consisting of short vectors in $rl_\pi(\mathcal{E})$. We check which of the determined short vectors are relations and, usually, we find a large number of relations. Once some relations are found we can use them to reduce the given generating set of $\mathcal{E}$ to a smaller set using a Smith normal form computation and then we iterate the process. The following outline gives a summary of this approach.

```
ReduceGenerators(E)
    assume that G = ⟨E⟩ is abelian and acts semisimply
    repeat
        choose a small finite set of new admissible primes π for G
        determine rl_π(E)
        compute a LLL basis B of rl_π(E)
        find those vectors C in B which are relations for E
        reduce E by C using a Smith normal form computation
    until C = ∅
    return E.
```

We reduce the generating set $\mathcal{E}$ until we 'guess' that $\mathcal{E}$ is multiplicatively independent. It remains to prove this assumption. For this purpose we determine the Dixon bound B of the remaining small generating set $\mathcal{E}$ and use the relation finding process to show that there are no relations in which all entries are smaller than the Dixon bound as described in Theorem 5.16. For example, the following approach can be used for this purpose.

```
IsMultiplicativelyIndependent(E)
    assume that G = ⟨E⟩ is abelian and acts semisimply
    let c be a generators of Q(G); that is, Q(c) = Q(G)
    determine the Dixon bound B = B(E, c)
    initialize T = Z^l where l is the length of E and p = 1
    repeat
        reset p to the next admissible prime for G larger than p
        compute rl_p(E) and reset T = T ∩ rl_p(E)
        let e be a shortest vector in T
        if the length of e is greater than √l B then
            return true
        elif e is a relation for E then
            return false
        end if
    until false
```

We note that the Dixon bound is exponential in the number of generators. Hence it is important for the practicality of the overall approach to reduce the given generating set first with 'ReduceGenerators' and then apply 'IsMultiplicativelyIndependent' incorporating the Dixon bound to a small generating set.

5.18. *Remark:* If we are content with a probabilistic method to determine relations, then we can just rely on the p-adic relation finding process for sufficiently many primes and omit the verification with the Dixon bound. Similarly, if we just want to find a single relation knowing that it must exists, then we can apply the p-adic relation finding process for increasing sets of primes π until we find the desired relation and we do not need the verification process.

5.5.2 Unipotent rational polycyclic matrix groups

The subgroup $U(d,\mathbb{Z})$ of upper unitriangular matrices in $GL(d,\mathbb{Z})$ is a torsion-free nilpotent group. It is generated by $\{I+E(i,j) \mid 1 \leq i < j \leq d\}$ where $E(i,j)$ is the $d \times d$ matrix with entries $E(i,j)_{ij} = 1$ and $E(i,j)_{kl} = 0$ if $(k,l) \neq (i,j)$. The subgroups $U_k = \langle I+E(i,j) \mid j-i \geq k \rangle$ form a central series through $U(d,\mathbb{Z})$ and $U_k/U_{k+1} \cong \mathbb{Z}^{d-k}$. These properties can be used to determine a constructive polycyclic sequence for $U(d,\mathbb{Z})$.
If $U \leq U(d,\mathbb{Z})$ is given by generators or normal subgroup generators under the action of $G \leq GL(d,\mathbb{Z})$, then we can use the constructive polycyclic sequence of $U(d,\mathbb{Z})$ to determine an induced polycyclic sequence of U as outlined in Sections 3.1 and 3.3. These methods also yield a test of membership for unipotent integral matrices in U.
Now we extend this approach to polycyclic subgroups U in the group $U(d,\mathbb{Q})$ of rational unitriangular matrices. Since U is finitely generated in this case, we can read off an integer a such that au is an integral matrix for each generator u of U. In the next lemma we observe that we can simultaneously conjugate all generators of U (and hence U itself) into $U(d,\mathbb{Z})$.

5.19. Lemma: *Let $u \in GL(d,\mathbb{Q})$ be a unipotent matrix. Let $a \in \mathbb{N}$ such that au has integer entries only and let $(e_1,\ldots,e_d)$ be a basis of $\mathbb{Q}^d$ which exhibits a flag for u. Consider the matrix g with rows $e_1, ae_2, \ldots, a^{d-1}e_d$. Then $u^g \in GL(d,\mathbb{Z})$.*

Proof. Without loss of generality we assume that $e_1,\ldots,e_d$ is the standard basis of $\mathbb{Q}^d$ and u is a matrix in upper unitriangular form. The matrix g is invertible in $GL(d,\mathbb{Q})$ and u^g is in upper unitriangular form. Thus u^g is unipotent and $det(u^g) = 1$. Hence it is sufficient to show that u^g has integer entries only. Let $f_i = a^{-(i-1)}e_i$ and consider the basis $f_1,\ldots,f_d$ of $\mathbb{Q}^d$. We have to show that u acts as integral matrix on this new basis. Let u_i be the i-th row u. Then $f_iu = a^{-(i-1)}u_i = f_i + v_i$ where v_i is a vector of depth at least $i+1$. Since au_i is integral, we obtain that a^iv_i is integral, i.e. $a^iv_i = b_{i+1}e_{i+1} + \ldots + b_de_d$ with $b_{i+1},\ldots,b_d \in \mathbb{Z}$. Thus $v_i = b_{i+1}f_{i+1} + ab_{i+2}f_{i+2} + \ldots + a^{d-i}b_df_d$ and v_i is an integral linear combination of the new basis elements. □

5.5.3 A summary of the algorithm

ConstructivePcSequence(G)
- determine an admissible prime p for G and construct $\psi_p : G \to GL(d,p)$
- determine a constructive polycyclic sequence for the image $I_p(G)$ (Chapter 4)
- calculate normal subgroup generators for $K_p(G)$
- compute the radical series $R_1 > \ldots > R_l > R_{l+1}$ for G using $K_p(G)$ (Section 5.4.3)
- let $\nu : G \to G_{R_1/R_2} \times \ldots \times G_{R_l/R_{l+1}}$ (Section 5.4.2)
- determine a constructive polycyclic sequence for $K_p(G)^\nu$ (Section 5.5.1)
- denote $U = ker(\nu|_{K_p(G)})$
- combine $I_p(G)$ and $K_p(G)^\nu$ to obtain a constructive polycyclic sequence of G/U
- calculate normal subgroup generators for U
- determine a constructive polycyclic sequence for the unipotent group U (Section 5.5.2)
- combine G/U and U to a constructive polycyclic sequence for G
- return the resulting constructive polycyclic sequence

Further references and comments

It would also be interesting to investigate the structure of the natural $\mathbb{Z}G$-module for an integral polycyclic matrix group G in a similar form as in Sections 5.2 and 5.3. However, the approaches described here do not extend to integral group rings. In particular, Lemma 5.3 does not hold for group rings $\mathbb{Z}G$ and hence the fundamental observation for our approach fails in this case.
Further, if G is a polycyclic semisimple rational matrix group with natural module V, then we can determine a direct factorization of V under action of $K_p(G)$. Although it is not necessary in our current applications for polycyclic groups, it might be interesting to determine a direct factorization of V under action of G. For this purpose the heuristic approach of Parker [61] for finite groups may be useful or applications of Clifford theory to this problem could be considered.

Chapter 6

Cohomology groups

Each polycyclic group has a normal series of finite length with elementary or free abelian factors. Thus we can think of a polycyclic group as a tower of elementary or free abelian groups. Cohomology theory can be used to describe the connections between the abelian factors in this tower. It provides a powerful tool for many computations in polycyclic groups.
We first recall the definition of cohomology groups and their group theoretic interpretations. Then we describe methods to determine the first and second cohomology group of a polycyclic group. These algorithms can be used to construct complements or extensions for polycyclic groups and hence they have a variety of applications.
We are mostly interested in cohomology groups of a polycyclic group acting on an elementary or free abelian module. We investigate these two cases of modules in more detail. In particular, we outline various conditions on the group and its module which force the corresponding cohomology groups to be finite or even trivial. These finiteness conditions will be useful for algorithmic purposes in later applications.
Almost complements are a generalization of complements which are of interest in infinite group theory. We consider almost complements to free abelian normal subgroups and describe a condition for their existence. This condition can be checked with the cohomology group algorithms introduced here. Further, we present an algorithm to determine an almost complement in a polycyclic group if it exists.

References: Algorithms to determine the first and the second cohomology groups for finite polycyclic groups and finite modules have been described and exploited by Celler, Neubüser & Wright in [9] and by Plesken and Brückner in [62, 7], respectively. Our methods are extensions of these algorithms to the infinite case. They have also been used in [17] where a report on implementations of these methods can be found.
Many theoretical investigations of complements and almost complements in infinite groups are due to Robinson. For general background on this topic we refer to [67]. More detailed information on almost complements and finite cohomology groups can be found in [66].

6.1 Definition of cohomology groups

Let G be a group and M a G-module. For $i \in \mathbb{N}$ we let G^i be the i-fold direct product of G and define

$$C^i(G,M) = \{\gamma : G^i \to M \mid (g_1,\ldots,g_i)^\gamma = 0 \text{ if some } g_j = 1\}.$$

Additionally, we denote $C^0(G,M) = M$. Then $C^i(G,M)$ has the structure of an abelian group with addition $x^{\gamma_1+\gamma_2} = x^{\gamma_1} + x^{\gamma_2}$. The i-th cohomology map $\alpha_i : C^{i-1}(G,M) \to C^i(G,M)$ is obtained by $(g_1,\ldots,g_i)^{\gamma^{\alpha_i}} = (g_2,\ldots,g_i)^\gamma + \sum_{j=1}^{i-1}(-1)^j(g_1,\ldots,g_{j-1},g_jg_{j+1},g_{j+2},\ldots,g_i)^\gamma + (-1)^i((g_1,\ldots,g_{i-1})^\gamma)^{g_i}$.

The map α_i is a homomorphism of abelian groups. We define the groups of i-th cocycles and i-coboundaries, respectively, by

$$Z^i(G,M) = ker(\alpha_{i+1}) \leq C^i(G,M) \quad \text{and} \quad B^i(G,M) = im(\alpha_i) \leq C^i(G,M).$$

We observe that $B^i(G,M) \leq Z^i(G,M)$ and define the i-th cohomology group as their factor group by $H^i(G,M) = Z^i(G,M)/B^i(G,M)$. We also often jointly denote $Z^i(G,M)$, $B^i(G,M)$ and $H^i(G,M)$ as *i-th cohomology groups*.

6.1.1 The first and second cohomology groups

The first and second cohomology groups play a special role in group theory, since they have important group theoretic interpretations. Thus we include the explicit definitions of these cohomology groups here additionally.

$$\begin{aligned}
Z^2(G,M) &= \{\gamma \in C^2(G,M) \mid (h,k)^\gamma + (g,hk)^\gamma = (gh,k)^\gamma + ((g,h)^\gamma)^k \text{ for } g,h,k \in G\},\\
B^2(G,M) &= \{\gamma \in C^2(G,M) \mid (g,h)^\gamma = (gh)^\delta - (g^\delta)^h - h^\delta \text{ for some } \delta \in C^1(G,M)\},\\
Z^1(G,M) &= \{\delta \in C^1(G,M) \mid (gh)^\delta = (g^\delta)^h + h^\delta \text{ for } g,h \in G\},\\
B^1(G,M) &= \{\delta \in C^1(G,M) \mid g^\delta = m^g - m \text{ for some } m \in M = C^0(G,M)\},\\
Z^0(G,M) &= \{m \in C^0(G,M) \mid m^g = m \text{ for } g \in G\}.
\end{aligned}$$

Further, we note that 1-cocycles and 1-coboundaries are often called *derivations* and *inner derivations*, respectively.

6.1.2 Compatible pairs acting on cohomology groups

Consider the group $A = Aut(G) \times Aut(M)$. Then A acts as $Aut(G)$ on G and as $Aut(M)$ on M. We say that $a \in A$ acts as a *compatible pair* on G and M if $m^{(g^a)} = ((m^{a^{-1}})^g)^a$ for each $m \in M$ and $g \in G$. The set of compatible pairs in A forms a subgroup C of A. We can derive an action of C on $C^i(G,M)$ via

$$(g_1,\ldots,g_i)^{\gamma^c} = ((g_1^{c^{-1}},\ldots,g_i^{c^{-1}})^\gamma)^c$$

for $c \in C$ and $\gamma \in C^i(G,M)$. It is not difficult to verify that such an action is compatible with the corresponding cohomology map. Thus we obtain that $Z^i(G,M)$ and $B^i(G,M)$ are setwise invariant under this action of C. Hence we can derive an induced action of C on the factor $H^i(G,M)$. The orbits and stabilizers under this action will be exploited for group theoretic purposes later.

6.2 Extensions and complements

Let G and N be groups. An *extension* of N by G is a group E such that N embeds as normal subgroup into E and $E/N \cong G$. We can identify E with the set $\{(g,n) \mid g \in G, n \in N\}$ and obtain $N \cong \{(1,n) \mid n \in N\}$. Note that the multiplication in the set of tuples is not necessarily componentwise. The following lemma recalls that extensions of abelian groups can be read off from the second cohomology group.

6.1. Lemma: *Let G be a group and M a G-module. There is a one-to-one correspondence between $Z^2(G,M)$ and the set of all extensions of M by G via $\gamma \mapsto E_\gamma = \{(g,m) \mid g \in G, m \in M\}$ where the multiplication in E_γ is defined by $(g,m)(h,n) = (gh,(g,h)^\gamma + m^h + n)$.*

Let E be a group with normal subgroup N. A *complement* to N in E is a subgroup K of E with $K \cap N = 1$ and $KN = E$. If a complement to N in E exists, then we say that the extension E of N by E/N *splits*. In the following lemma we recall that the splitting of an extension of an abelian group can be read off from the second cohomology group.

6.2. Lemma: *Let G be a group, M a G-module and $\gamma \in Z^2(G, M)$. The extension E_γ is split if and only if $\gamma \in B^2(G, M)$. In this case γ is the image of a map $\delta \in C^1(G, M)$ and $\{(g, g^\delta) \mid g \in G\}$ is a complement to M in E_γ.*

If a group splits over an abelian normal subgroup, then we can read off the complements to this abelian normal subgroup and their conjugacy classes in the underlying group from the first cohomology group. This is made explicit in the following lemma.

6.3. Lemma: *Let M be an abelian normal subgroup of a group E and denote $G = E/M$. Suppose that there exists a complement K to M in E. Since $K \cong G$, we can write $K = \{r_g \mid g \in G\}$ for certain representatives $r_g \in E$.*

a) There is a one-to-one correspondence between $Z^1(G, M)$ and the set of all complements to M in E via $\gamma \mapsto K_\gamma = \{r_g \cdot g^\gamma \mid g \in G\}$.

b) Two complements K_{γ_1} and K_{γ_2} are conjugate in E if and only if $\gamma_1 \equiv \gamma_2 \bmod B^1(G, M)$.

In later applications of the determination of complement classes we have to compute complements to an abelian normal subgroup in a polycyclic group under the action of a parent group. For this purpose we consider the linear action on the first cohomology group as introduced in Section 6.1.2. We use this to construct an affine action which yields the desired classes as outlined in the following lemma.

6.4. Lemma: *Let M be an abelian normal subgroup of a group E. We consider U and H with $M \leq U \leq H \leq N_E(U)$ and we suppose that there exists a complement K to M in U. Denoting $G = U/M$ we can write $K = \{r_g \mid g \in G\}$. The normalizing subgroup H acts by natural conjugation on G and on M. For $h \in H$ and $\delta : G \to M$ we define*

$$\delta_h : G \to M : g \mapsto ((g^{h^{-1}})^\delta)^h \quad \text{and} \quad \tau_h : G \to M : g \mapsto r_g^{-1} \cdot (r_{g^{h^{-1}}})^h.$$

a) Then H acts on $Z^1(G, M)$ via $\delta^h = \delta_h + \tau_h$. This action is affine with linear part δ_h and translation part τ_h.

b) The orbits of H in $H^1(G, M)$ correspond one-to-one to the H-classes of complements to M in U via $(K_\delta)^h = K_{\delta^h}$ with K_δ defined as in Lemma 6.3.

c) Let $\delta \in Z^1(G, M)$ and define $S = Stab_H(\delta + B^1(G, M))$. Then each $s \in S$ defines an element $\gamma_s \in B^1(G, M)$ via $\delta^s = \delta + \gamma_s$ and the map γ_s is of the form $\gamma_s : G \to M : g \mapsto [g, m_s]$ for an element $m_s \in M$. We obtain $N_H(K_\delta) = \{s \cdot m_s^{-1} \mid s \in S\}$.

Proof. a) is an elementary observation. A direct computation shows that $(K_\delta)^h = K_{\delta^h}$ for $h \in H$ and thus b) follows. For c) we first note that the orbit of δ under the action of U is $\delta + B^1(F, A)$ and this is a block for the orbit of H. The corresponding block stabilizer in H is S. Thus by lifting the elements of S to the stabilizer of δ we obtain the normalizer of U in H. (See Section 4.1 also.) □

Hence the determination of conjugacy classes of complements under action of an automorphism group reduces to an orbit stabilizer calculation acting on elements of the abelian first cohomology group. Methods for this purpose in the cases we are interested in will be described in more detail in Chapter 7.

6.5. *Remark:* Note that it is straightforward to derive a method to determine H-normal complements from Lemma 6.4. In this case we compute those $\delta \in Z^1(G, M)$ which are fixed under the action of H.

6.3 Determining the first and second cohomology groups

As a first step to determine the desired cohomology groups we describe an algorithmically more useful representation for them. Based on these representations we obtain a new description for the corresponding cohomology maps which we then exploit for algorithmic purposes.

6.3.1 Faithful representations of $H^1(G,M)$ and $H^2(G,M)$

As a first step we introduce representations τ_i for $C^i(G,M)$ for $i = 0, 1$ and 2. To construct τ_0, no restrictions on G are needed, for τ_1 we need that G is finitely generated, say $G = \langle g_1, \ldots, g_n \rangle$, and for τ_2 we need that G is finitely presented, say $G = \langle g_1, \ldots, g_n \mid r_1, \ldots, r_l \rangle$.
We consider the case that G is finitely presented further. By definition, each relator r_i is a word in the generators and their inverses. Thus we can denote $r_i = r_{i,1}^{e_{i,1}} \cdots r_{i,l_i}^{e_{i,l_i}}$ where each $r_{i,j}$ is a generator of G and $e_{i,j} = \pm 1$. We define the tail subwords $w_{i,j} = r_{i,j}^{e_{i,j}} \cdots r_{i,l_i}^{e_{i,l_i}}$ of the relator r_i. Further, let $\epsilon_{i,j} = 1$ if $e_{i,j} = -1$ and $\epsilon_{i,j} = 0$ otherwise. We introduce the following representations.

$$\begin{aligned}
\tau_0 &: C^0(G,M) \to M^1 : \gamma \mapsto (1^\gamma) \\
\tau_1 &: C^1(G,M) \to M^n : \gamma \mapsto (g_i^\gamma \mid 1 \le i \le n) \\
\tau_2 &: C^2(G,M) \to M^l : \gamma \mapsto (\sum_{j=1}^{l_i} \epsilon_{i,j}((r_{i,j}, r_{i,j}^{-1})^\gamma)^{w_{i,j+1}} - \sum_{j=1}^{l_i-1} (r_{i,j}^{e_{i,j}}, w_{i,j+1})^\gamma \mid 1 \le i \le l)
\end{aligned}$$

We consider the images of the cohomology groups under these maps τ_i further. Since G and M are usually well-defined by the context, we use the following shortened notation.

$$Z_i = Z^i(G,M)^{\tau_i} \quad \text{and} \quad B_i = B^i(G,M)^{\tau_i}.$$

The following lemma yields group theoretic interpretations of the images of the maps τ_i in correspondence to the descriptions of Section 6.2.

6.6. Lemma: *Let M be a G-module and G as required above.*

a) Let E be a split extension of M by G with complement $K = \{r_g \mid g \in G\}$ as in Lemma 6.3. If $(s_1, \ldots, s_n) = \gamma^{\tau_1}$ for some $\gamma \in Z^1(G,M)$, then $K_\gamma = \langle r_{g_i} \cdot s_i \mid 1 \le i \le n \rangle$.

b) If $(t_1, \ldots, t_l) = \gamma^{\tau_2}$ for a cocycle $\gamma \in Z^2(G,M)$, then $(r_{i,1}, 0)^{e_{i,1}} \cdots (r_{i,l_i}, 0)^{e_{i,l_i}} \cdot (1, t_i)$ is a relator of the extension E_γ as defined in Lemma 6.1.

Proof. a) By Lemma 6.3 and the definition of τ_1 we obtain that $r_{g_i} s_i \in K_\gamma$. Further, the set $\{g_1, \ldots, g_n\}$ generates G and, moreover, $G \cong K_\gamma$ via $g_i \mapsto r_{g_i} s_i$. Thus the images under this isomorphism generate K_γ which proves a).
b) We consider the word $w_i = (r_{i,1}, 0)^{e_{i,1}} \cdots (r_{i,l_i}, 0)^{e_{i,l_i}}$ in E_γ. Note that $(r_{i,j}, 0)^{-1} = (r_{i,j}^{-1}, -(r_{i,j}, r_{i,j}^{-1})^\gamma)$. Now a straightforward induction shows that $w_i = (r_i, -t_i)$. Since r_i is a relator of G, we obtain $w_i = (1, -t_i)$. □

The maps τ_i are epimorphisms of abelian groups. But they are in general not injective on $C^i(G,M)$. The following lemma shows that their restrictions to $Z^i(G,M)$ are almost injective and that we obtain a faithful representations of $H^i(G,M)$ using τ_i.

6.7. Lemma: *Let M be a G-module and G as required above.*

a) $ker(\tau_0) = 0$.

b) $ker(\tau_1) \cap Z^1(G,M) = 0$.

c) $ker(\tau_2) \cap Z^2(G,M) \leq B^2(G,M)$. *Further, if* $\gamma \in ker(\tau_2) \cap Z^2(G,M)$ *and* γ *is the image of* $\delta \in C^1(G,M)$, *then* $\delta \in ker(\tau_1)$.

In particular, $Z_i/B_i \cong H^i(G,M)$ *for* $i = 1, 2$.

Proof. a) and b) are straightforward to observe and we consider only c). Let $\gamma \in ker(\tau_2) \cap Z^2(G,M)$ and define $K = \langle (g_i, 1) \mid 1 \leq i \leq n \rangle \leq E_\gamma$. The natural epimorphism $E_\gamma \to G : (g_i, 1) \mapsto g_i$ factors through K. Further, Lemma 6.6 b) yields that K fulfills all relators of a presentation of G on the images of the epimorphism. Thus this epimorphism is injective on K and K is naturally isomorphic to G. Hence K is a complement to M in E_γ and $\gamma \in B^2(G,M)$ by Lemma 6.2. Thus, by definition, γ is the image of some $\delta \in C^1(G,M)$ and $\delta \in ker(\tau_1)$. □

Our aim in the next sections is to determine maps β_1 and β_2 such that the following diagram commutes.

$$\begin{array}{ccccc} C^0(G,M) & \stackrel{\alpha_1}{\to} & C^1(G,M) & \stackrel{\alpha_2}{\to} & C^2(G,M) \\ \downarrow \tau_0 & & \downarrow \tau_1 & & \downarrow \tau_2 \\ M & \stackrel{\beta_1}{\to} & M^n & \stackrel{\beta_2}{\to} & M^l \end{array}$$

6.3.2 Representing α_1 by β_1

Let $G = \langle g_1, \ldots, g_n \rangle$ be a finitely generated group. We determine an explicit map $\beta_1 : M \to M^n$ which corresponds to $\alpha_1 : C^0(G,M) \to C^1(G,M)$. We consider the following $1 \times n$-matrix over $\mathbb{Z}G$.

$$\beta_1 = \begin{pmatrix} g_1 - 1 & \ldots & g_n - 1 \end{pmatrix}$$

It is straightforward to observe that this is a representation of α_1; that is, we obtain $\alpha_1 \cdot \tau_1 = \tau_0 \cdot \beta_1$. In particular, this yields $Z_0 = ker(\beta_1)$ and $B_1 = im(\beta_1)$.

6.3.3 Representing α_2 by β_2

Let $G = \langle g_1, \ldots, g_n \mid r_1, \ldots, r_l \rangle$ be a finitely presented group. We introduce a map $\beta_2 : M^n \to M^l$ which corresponds to $\alpha_2 : C^1(G,M) \to C^2(G,M)$.
As above, we write the relators as words $r_i = r_{i,1}^{e_{i,1}} \cdots r_{i,l_i}^{e_{i,l_i}}$ where $r_{i,j}$ is a generator of G and $e_{i,j} = \pm 1$. Further, we consider the tail subwords $w_{i,j} = r_{i,j}^{e_{i,j}} \cdots r_{i,l_i}^{e_{i,l_i}}$ and we let $\epsilon_{i,j} = 1$ if $e_{i,j} = -1$ and $\epsilon_{i,j} = 0$ otherwise. We use this notation to define elements $b_{i1}, \ldots, b_{il_i}$ in the group algebra $\mathbb{Z}G$ by $b_{ij} = e_{i,j} r_{i,j}^{-\epsilon_{i,j}} w_{i,j+1}$. Based on this definition we introduce elements $a_{ik} \in \mathbb{Z}G$ defined by $a_{ik} = \sum \{ b_{ij} \mid w_j = g_k^{\pm 1} \}$. Thus we obtain the following $n \times l$-matrix over $\mathbb{Z}G$.

$$\beta_2 = \begin{pmatrix} a_{11} & \ldots & a_{l1} \\ \vdots & & \vdots \\ a_{1n} & \ldots & a_{ln} \end{pmatrix}$$

6.8. Lemma: *The map β_2 represents α_2. In particular, $Z_1 = ker(\beta_2)$ and $B_2 = im(\beta_2)$.*

Proof. We want to show that $\alpha_2 \cdot \tau_2 = \tau_1 \cdot \beta_2$. Let $\delta \in C^1(G,M)$. Then δ defines $\gamma \in C^2(G,M)$ via $(g,h)^\gamma = (gh)^\delta - (g^\delta)^h - h^\delta$. Now we compute the image $(t_1,\ldots,t_l)$ of γ under τ_2. Recall that $1^\delta = 0$ by definition.

$$\begin{aligned}
t_i &= -\sum_{j=1}^{l_i-1}(r_{i,j}^{e_{i,j}}, w_{i,j+1})^\gamma + \sum_{j=1}^{l_i}\epsilon_{i,j}((r_{i,j}, r_{i,j}^{-1})^\gamma)^{w_{i,j+1}} \\
&= -\sum_{j=1}^{l_i-1}(r_{i,j}^{e_{i,j}} \cdot w_{i,j+1})^\delta - ((r_{i,j}^{e_{i,j}})^\delta)^{w_{i,j+1}} - (w_{i,j+1})^\delta \\
&\quad + \sum_{j=1}^{l_i}\epsilon_{i,j}(((r_{i,j}\cdot r_{i,j}^{-1})^\delta)^{w_{i,j+1}} - ((r_{i,j})^\delta)^{r_{i,j}^{-1}w_{i,j+1}} - ((r_{i,j}^{-1})^\delta)^{w_{i,j+1}}) \\
&= -\sum_{j=1}^{l_i-1}(w_{i,j})^\delta - ((r_{i,j}^{e_{i,j}})^\delta)^{w_{i,j+1}} - (w_{i,j+1})^\delta + \sum_{j=1}^{l_i}\epsilon_{i,j}(-((r_{i,j})^\delta)^{r_{i,j}^{-1}w_{i,j+1}} - ((r_{i,j}^{-1})^\delta)^{w_{i,j+1}}) \\
&= -(w_{i,1})^\delta + (w_{i,l_i})^\delta - (r_{i,l_i}^{e_{i,l_i}})^\delta + \sum_{j=1}^{l_i}\epsilon_{i,j}(-((r_{i,j})^\delta)^{r_{i,j}^{-1}w_{i,j+1}} - ((r_{i,j}^{-1})^\delta)^{w_{i,j+1}}) + ((r_{i,j}^{e_{i,j}})^\delta)^{w_{i,j+1}}
\end{aligned}$$

First note that $(w_{i,1})^\delta = r_i^\delta = 0$ and $(w_{i,l_i})^\delta = (r_{i,l_i}^{e_{i,j}})^\delta$. Thus the first three summands in the last line of the equation cancel out. We consider one summand of the indexed sum in the last line of the equation. If $e_{i,j} = 1$ and $\epsilon_{i,j} = 0$, then this summand amounts to $((r_{i,j})^\delta)^{w_{i,j+1}}$. Otherwise $e_{i,j} = -1$ and $\epsilon_{i,j} = 1$ which yields $-((r_{i,j})^\delta)^{w_{i,j}}$ after cancellation. In both cases we have that the summand can be written as $((r_{i,j})^\delta)b_{ij}$ where we act with elements of $\mathbb{Z}G$ by multiplication from the right.
Therefore, we obtain $t_i = \sum_{j=1}^{l_i}((r_{i,j})^\delta)b_{ij}$. By collecting this sum we get $t_i = \sum_{j=1}^{n}(g_j^\delta)a_{ij}$. Since $\delta^{\tau_1} = (g_1^\delta,\ldots,g_n^\delta)$, this yields $t = (\delta^{\tau_1})^{\beta_2}$ as desired. □

6.3.4 Computing Z_2

We include a brief account of an algorithm to determine Z_2 without formally introducing an explicit representation for the cohomology map α_3. We suppose for this purpose that G is a polycyclic group given by a constructive polycyclic sequence. Thus we can determine a consistent polycyclic presentation of G with generators $g_1,\ldots,g_n$ and relators $r_1,\ldots,r_l$, say.
Let M be a G-module and $t = (t_1,\ldots,t_l) \in M^l$. We consider a group E_t defined on the generators $\{(g_1,0),\ldots,(g_n,0),(1,m) \mid m \in M\}$ and having relations of the following three types.

I. The extended relators of G: $r_i((g_1,0),\ldots,(g_n,0))(1,t_i)$ for $1 \le i \le l$.

II. The action of G on M: $(1,m)^{(g_j,0)}(1,-m^{g_j})$ and $(1,m)^{(g_j^{-1},0)}(1,-m^{g_j^{-1}})$ for $m \in M, 1 \le j \le n$.

III. Relators of M: $(1,m)^{(1,n)}(1,-m)$ for $m,n \in M$ and $(1,m)^e$ if $m \in M$ has order e.

We obtain $M \cong \{(1,m) \mid m \in M\} \trianglelefteq E_t$ by the relations of type II and III. Thus it remains to consider the factor E_t/M in more detail. The relators introduced have the form of a polycyclic presentation for E_t. Thus, in particular, the initial segment $((g_1,0),\ldots,(g_n,0))$ of the defining generators of this presentation forms a polycyclic sequence for E_t mod M. This, in turn, yields by Lemma 1.3 that each element of E_t can be written in the form $(g_1,0)^{e_1}\cdots(g_n,0)^{e_n}\cdot(1,m)$ for certain exponents $(e_1,\ldots,e_n)$ which are restricted by the relative orders of the polycyclic sequence. In the following lemma we obtain a condition on this sequence that asserts that E_t is an extension of M by G.

6.9. Lemma: *Let* $\psi_t : E_t \to G : (g_1,0)^{e_1} \cdots (g_n,0)^{e_n} \cdot (1,m) \mapsto g_1^{e_1} \cdots g_n^{e_n}$.

a) ψ_t is well-defined, if the relative orders of the polycyclic sequence $(g_1,\ldots,g_n)$ coincide with the relative orders for the initial segment $((g_1,0),\ldots,(g_n,0))$ of the polycyclic sequence of E_t.

b) If ψ_t is a well-defined map, then it is an epimorphism with kernel M. In particular, E_t is an extension of M by G in this case.

Proof. a) Lemma 1.3 shows that each element of G can be written in a unique form as a word in the polycyclic sequence $(g_1,\ldots,g_n)$ depending on the relative orders of the polycyclic sequence. If the relative orders on both sides of the map ψ_t coincide, then also the preimages can be written in corresponding unique forms.
b) Suppose that ψ_t is well-defined. Then the images of the generators of E_t satisfy the relations of E_t and hence ψ_t is a homomorphism. Then, clearly, ψ_t is also surjective and $M \leq ker(\psi_t)$. We consider $w \in ker(\psi_t)$. Let $w = (g_1,0)^{e_1} \cdots (g_n,0)^{e_n} \cdot (1,m)$ be the unique normal form in the generators of E_t. Then $w^{\psi_t} = g_1^{e_1} \cdots g_n^{e_n} = 1$. As the latter is the unique normal form in the generators of G, we obtain $e_i = 0$ for $1 \leq i \leq n$. Hence $ker(\psi_t) = M$. □

Hence, by Lemma 6.9, it remains to determine those vectors $t \in M^l$ with the property that the relative orders of the sequences $(g_1,\ldots,g_n)$ and $((g_1,0),\ldots,(g_n,0))$ coincide. By Lemma 2.5, this is the case if the presentation defined for E_t fulfills the consistency relations on the sequence $((g_1,0),\ldots,(g_n,0))$.
If a vector $t \in M^l$ is given, then we check the consistency of the corresponding presentation using the method of Section 2.3. To determine all vectors $t \in M^l$ that yield consistent presentations we generalize that process. We consider the vector t as a vector of undetermined elements of M and perform the consistency test using these variables. Since we can represent the action of G on M by elements of the group algebra $\mathbb{Z}G$, we obtain conditions on the vector $t \in M^l$ that need to be fulfilled.
More precisely, each consistency relation c_i yields a vector $(c_{i1},\cdots,c_{il}) \in (\mathbb{Z}G)^l$. We combine these vectors in an $l \times m$-matrix over $\mathbb{Z}G$:

$$\beta_3 = \begin{pmatrix} c_{11} & \ldots & c_{m1} \\ \vdots & & \vdots \\ c_{1l} & \ldots & c_{ml} \end{pmatrix}.$$

This matrix can be considered as a representation of α_3 by Lemma 6.6. Thus we denote the matrix as β_3 and we obtain $Z_2 = ker(\beta_3)$.

6.3.5 Computing with matrices over $\mathbb{Z}G$

In the Sections 6.3.2 - 6.3.4 we obtain a representation of the cohomology map α_i as matrix β_i with entries in $\mathbb{Z}G$. In all cases we want to determine kernel and image of this matrix.
If $M \cong R^d$ for a suitable ring R, then the action of G on M yields a homomorphism $G \to GL(d,R) : g \mapsto \overline{g}$. This group homomorphism extends to a ring homomorphism $\mathbb{Z}G \to M_d(R)$ and we can use this ring homomorphism to express the entries in β_i as $d \times d$-matrices over R. Hence β_i can be considered as a matrix over R.
With this representation we can then easily derive R-bases for the kernel and the image of the corresponding matrix if we can compute effectively in R. In the three cases we are usually interested in, that is, R a finite field, $R = \mathbb{Z}$ or $R = \mathbb{Q}$, the necessary operations on the matrix β_i to determine bases for kernel and image are standard linear algebra.

6.3.6 Free abelian modules

The case that G is a group and M is a G-module with $M = \mathbb{Z}^d$ will be of particular interest in later applications of cohomology groups. We can effectively compute the matrices β_i over $\mathbb{Z}G$ in this case and extend them to integral matrices using the explicit action of G on M.
The action of G on M induces two other types of modules: first, we can also consider $M_{\mathbb{Q}} = M \otimes \mathbb{Q} = \mathbb{Q}^d$ as a natural G-module and secondly $M_p = (\mathbb{Z}/p\mathbb{Z})^d$ is a G-module derived from M for every prime p. The cohomology maps for these new modules can be read off readily from the cohomology maps for M: for $M_{\mathbb{Q}}$ we only need to consider the integral matrices β_i as rational matrices and for M_p we apply the p-congruence homomorphism to β_i.

6.10. Lemma: *Let G be a group and let M be a finitely generated integral G-module. Let $i \in \{1, 2\}$ and consider the cohomology groups $Z_i(\mathbb{Z})$ and $B_i(\mathbb{Z})$ for M. We denote by $Z_i(\mathbb{Q})$ and $B_i(\mathbb{Q})$ the corresponding groups for $M_{\mathbb{Q}}$. If $Z_i(\mathbb{Q})/B_i(\mathbb{Q}) = 0$, then $Z_i(\mathbb{Z})/B_i(\mathbb{Z})$ is finite.*

Proof. By construction, $B_i(\mathbb{Z}) = \{t\beta_{i-1} \mid t \in \mathbb{Z}^n\}$ and $B_i(\mathbb{Q}) = \{t\beta_{i-1} \mid t \in \mathbb{Q}^n\}$ for some dimension $n \in \mathbb{N}$ depending on i. Similarly, $Z_i(\mathbb{Z}) = \{t \in \mathbb{Z}^l \mid t\beta_i = 0\}$ and $Z_i(\mathbb{Q}) = \{t \in \mathbb{Q}^l \mid t\beta_i = 0\}$ for some dimension $l \in \mathbb{N}$ depending on i. In particular, $B_i(\mathbb{Z}) \otimes \mathbb{Q} = B_i(\mathbb{Q})$ and similarly for cocycles. Thus $\mathbb{Z}_i(\mathbb{Z})/B_i(\mathbb{Z})$ is a torsion group and hence, since the underlying groups are finitely generated, we obtain that the factor is finite. □

6.4 Finiteness conditions for cohomology groups

Our primary application for cohomology groups will be the exploitation of their group theoretic interpretations as outlined in Section 6.2. In particular, we want to derive explicit lists of complements or extensions from cohomology groups. However, this will only be possible if the considered cohomology groups are finite. Thus we recall a number of finiteness conditions on cohomology groups in this section. The finiteness conditions for finite acting groups as recalled in the following lemma are well-known. They are proved for example in [67].

6.11. Lemma: *Let G be a finite group, M a G-module and $i \in \mathbb{N}$.*

a) $H^i(G, M)$ is a torsion group of exponent at most $exp(G)$.

b) If M is finitely generated, then $Z^i(G, M)$ is finitely generated and $H^i(G, M)$ is finite.

c) If M is finite, then $Z^i(G, M)$ is finite.

Next we consider the case that there exists a subgroup of finite index in the acting group which has trivial cohomology. In [26] Gaschütz has shown that this can be lifted to the full acting group depending on the index of the subgroup and the module.

6.12. Theorem: (Gaschütz) *Let $N \trianglelefteq G$ with $[G : N] = k < \infty$. Suppose that M is a G-module such that the map $M \to M : m \mapsto km$ is a bijection. If $i \in \{1, 2\}$ and $H^i(N, M) = 0$, then $H^i(G, M) = 0$.*

A variety of finiteness conditions arising from the action of a nilpotent normal subgroup of G have been considered in different places. The most general version of these theorems has been presented by Robinson in [66]. We recall Robinson's theorems here for the cases that we are interested in.

6.13. Theorem: *Let N be a nilpotent normal subgroup of G and let $M = R^d$ be a G-module.*

a) If R is a finite field and either $C_M(N) = 0$ or $M = [N, M]$, then $H^i(G, M) = 0$ for $i > 0$.

b) If $R = \mathbb{Z}$ or $R = \mathbb{Q}$ and $M = [N, M]$, then $H^i(G, M) = 0$ for $i > 0$.

c) If $R = \mathbb{Z}$ or $R = \mathbb{Q}$ and $C_M(N) = 0$, then $H^i(G, M)$ has finite exponent for $i > 0$.

Proof. Part a) follows from Theorem A in [66]. (Note that a nilpotent group N is a Gruenberg group. Further, $N/C_N(M)$ is finite and thus FC-hypercentral in G. Clearly, M is finite and thus Noetherian and Artinian as G-module.) Parts b) and c) follow from Theorem D in [66]. (Note that a nilpotent group N is a Baer group and $\mathbb{Z}^d$ and $\mathbb{Q}^d$ are both torsion-free and both have finite (Prüfer) rank d; that is, each finitely generated subgroup can be generated by at most d elements.) □

Finally, we note the following correspondence between the conditions in Theorem 6.13 b) and c). A proof can be found in [66], Lemma 5.12.

6.14. Lemma: *Let $M = R^d$ be an N-module for a nilpotent group N with $R = \mathbb{Z}$ or $R = \mathbb{Q}$. Then $C_N(M) = 0$ if and only if $M/[N, M]$ is finite.*

6.5 Almost complements

Let E be a polycyclic group with a normal subgroup N. A subgroup K is an *almost complement* to N in E if $K \cap N = 1$ and KN has finite index in E. If this index is trivial, then K is a complement and hence almost complements form a generalization of complements. Further, if an almost complement to N in E exists, then E is *almost split* or *nearly split* over N. Obviously, the definition of almost complements and almost splittings is only useful for an infinite group E.
Let G be a finitely generated group with a free abelian G-module M. Lemma 6.2 yields that each extension of M by G splits if and only if $H^2(G, M) = 0$. Here, in Section 6.5.1, we recall that each extension of M by G almost splits if and only if $H^2(G, M)$ is finite. This is useful in later applications, since the finiteness of $H^2(G, M)$ occurs more often than triviality; for example, this is also indicated by Theorem 6.13.
Clearly, the finiteness of $H^2(G, M)$ can be checked by the methods of Section 6.3.1 and thus we have a method to test if all extensions of M by a polycyclic group G are almost split. Further, in Section 6.5.2 we introduce an algorithm test if a given extension E of M by G is almost split and to determine an almost complement to M in E in this case.

6.5.1 The existence of almost complements

Robinson initiated the investigation of finiteness conditions for cohomology groups and the existence of almost complements in [65]. We recall one of the main theorems derived from this investigation in the following.

6.15. Theorem: *Let G be a finitely generated group, $M = \mathbb{Z}^d$ a G-module and $\gamma \in Z^2(G, M)$.*

a) $\gamma + B^2(G, M)$ has finite order if and only if $\gamma \in B^2(G, M_{\mathbb{Q}})$ where $M_{\mathbb{Q}} = M \otimes \mathbb{Q} = \mathbb{Q}^d$.

b) E_γ is almost split over M if and only if $\gamma + B^2(G, M)$ has finite order.

Proof. a) This follows directly from $B^2(G,M)\otimes\mathbb{Q} = B^2(G,M_\mathbb{Q})$.
b) $\Rightarrow$ This can be proved similar to Theorem 6.12. We include a sketch of this construction here for completeness. Suppose that E_γ is almost split over M and let $K \leq G$ a subgroup of finite index such that $C = \{(k,c_k) \mid k \in K\}$ is an almost complement to M in E_γ. Then $(k,l)^\gamma = c_{kl} - c_k^l - c_l$ for all $k,l \in K$. Let T be a transversal of K in G so that each element $g \in G$ can be written uniquely as $g = kt$ for $t \in T$ and $k \in K$. Using this setup we show that $\gamma \in B^2(G,M_\mathbb{Q})$ in two steps. First, we consider the map $\delta_1 : G \to M : kt \mapsto c_k^t + (k,t)^\gamma$ and we define $\gamma_1 = \delta_1^{\alpha_2} \in B^2(G,M)$. Then it is straightforward to observe that $(k,g)^\gamma = (k,g)^{\gamma_1}$ for all $k \in K$ and $g \in G$. We denote $\gamma_2 = \gamma - \gamma_1$ and obtain that $(k,g)^{\gamma_2} = 0$ for all $k \in K$ and, further, that $(kt,g)^{\gamma_2} = (t,g)^{\gamma_2}$ for all $k \in K$. As second step we define $\delta_2 : G \to M : g \mapsto -[G:K]^{-1}\sum_{t\in T}(t,g)^{\gamma_2}$. It is now a direct computation to show that $\gamma_2 = \delta_2^{\alpha_2} \in B^2(G,M_\mathbb{Q})$. Thus we also obtain that $\gamma = \gamma_1 + \gamma_2 \in B^2(G,M_\mathbb{Q})$ as desired.
b) $\Leftarrow$ We outline a constructive proof for this part of the Theorem. Let $\gamma \in Z^2(G,M) \cap B^2(G,M_\mathbb{Q})$ and suppose that γ is the image of a map $\delta : G \to M_\mathbb{Q}$ via $(h,k)^\gamma = (hk)^\delta - (h^\delta)^k - k^\delta$.
Let $g_1,\dots,g_n$ be a generating set of G. Since $g_i^\delta \in \mathbb{Q}^d$, there exists an integer e with $e(g_i^\delta) \in M$ for $1 \leq i \leq n$. As $(gh)^\delta \equiv (g^\delta)^h + h^\delta \bmod M$ for each $g,h \in G$, we obtain that $e(g^\delta) \in M$ for each $g \in G$. Hence $im(\delta) \leq M_e = \frac{1}{e}M$ and $\gamma \in B^2(G,M_e)$. Thus γ defines an extension E of M_e and $C = \{(g,g^\delta) \mid g \in G\}$ is a complement to M_e in E. The extension E_γ of M by G embeds into E such that $E_\gamma \cap M_e = M$ and $E_\gamma M_e = E$.
Let $R = C_G(M_e/M)$ and $L = R'R^e$. Since M_e/M is a finite group of order de, we obtain that L is a normal subgroup of G with finite index. We show that $g^\delta \in M$ for each $g \in L$. Let $h,k \in R$. Then $(hk)^\delta \equiv (h^\delta)^k + k^\delta \equiv h^\delta + k^\delta \bmod M$, since h centralizes M_e/M. Hence $[h,k]^\delta \equiv 0 \bmod M$. Further, $(h^e)^\delta \equiv e(h^\delta) \equiv 0 \bmod M$. Thus, by the definition of L, we have $g^\delta \in M$ for $g \in L$.
Hence $K = \{(g,g^\delta) \mid g \in L\} \leq E_\gamma \cap C$ and thus $K \cap M = 1$. Further, $[E_\gamma : KM] = [G:L] < \infty$. Therefore, K is the desired almost complement to M in E_γ. □

6.16. Corollary: *Let G be a finitely generated group with free abelian G-module M of finite rank. Each extension of M by G is almost split if and only if $H^2(G,M)$ is finite; that is, if $H^2(G,M_\mathbb{Q}) = 0$.*

6.5.2 Determining an almost complement

Let E be a group with free abelian normal subgroup M and denote $G = E/M$. Further, let $\gamma \in Z^2(G,M)$ be the cocycle corresponding to the extension E. By Theorem 6.15 an almost complement to M in E exists if and only if $\gamma \in B^2(G,M_\mathbb{Q})$, where $M_\mathbb{Q}$ is the rational module corresponding to M. This condition can readily be checked using the methods of Section 6.3. Further, we can compute a map $\delta \in C^1(G,M_\mathbb{Q})$ inducing γ.

6.17. Lemma: *Let G be a group with G-module $M = \mathbb{Z}^d$. Let $\gamma \in Z^2(G,M) \cap B^2(G,M_\mathbb{Q})$ and suppose that γ is the image of $\delta \in C^1(G,M_\mathbb{Q})$.*

a) Then $(gh)^\delta \equiv (g^\delta)^h + h^\delta \bmod M$ for $g,h \in G$.

b) $U = \{g \in G \mid g^\delta \in M\}$ forms a subgroup in G.

Proof. a) This is straightforward as $(gh)^\delta = (g,h)^\gamma + (g^\delta)^h + h^\delta$ by definition.
b) We observe $(gh^{-1})^\delta = (g,h^{-1})^\gamma + (g^\delta)^{h^{-1}} + (h^{-1})^\delta \equiv (g^\delta)^{h^{-1}} + (h^\delta)^{h^{-1}} \bmod M$. Hence, if $g,h \in U$, then $gh^{-1} \in U$ and thus U is a subgroup. □

The subgroup U determined in Lemma 6.17 b) is the maximal subgroup of G which yields a splitting via δ. We describe an algorithm to determine this maximal splitting subgroup and a corresponding complement in the remainder of this section.

In the notation of Lemma 6.17 let $\delta^* : G \to M_{\mathbb{Q}}/M : g \mapsto g^\delta + M$. Then δ^* is a derivation of G and U can be described as the kernel of this derivation. The kernel of a derivation can be determined as a stabilizer under an action of G. More precisely, G acts on $M_{\mathbb{Q}}/M$ via $(v + M)g = vg + g^\delta + M$ and we obtain

$$U = ker(\delta^*) = Stab_G((0, \ldots, 0) + M).$$

This description of U yields a method to determine U. By construction, we know that U has finite index in G. Thus the finite orbits algorithm of Section 4.2 can be applied to determine U as the stabilizer of a vector. This algorithm assumes that the input group G is given by a polycyclic sequence $\mathcal{G}$ and it returns the desired stabilizer U together with an induced polycyclic sequence $\mathcal{U}$. By bookkeeping, we can also determine the images u^δ for $u \in \mathcal{U}$ and thus we obtain the desired almost complement $\langle (u, u^\delta) \mid u \in \mathcal{U} \rangle$.

AlmostComplement(E, M)
  assume that M is a free abelian normal subgroup of E
  let $t = \gamma^{\tau_2}$ for the cocycle corresponding to E and M
  denote $G = E/M = \langle g_1, \ldots, g_n \rangle$
  compute $s \in \mathbb{Q}^n$ with $s\beta_2 = t$ over $\mathbb{Q}$ (Section 6.3)
  obtain $\delta : G \to \mathbb{Q}^d : g_i \mapsto s_i$
  compute $U = Stab_G((0, \ldots, 0) + M)$ with corresponding images under δ (Section 4.2)
  return the almost complement $\langle (u, u^\delta) \mid u \in \mathcal{U} \rangle$ where $\mathcal{U}$ is a polycyclic sequence of U

6.18. *Remark:* Kernels of derivations and their correspondence to affine actions will also be described and exploited in Section 7.1. The features introduced there can be applied to improve the almost complement algorithm described here.

Chapter 7

Orbits and stabilizers for polycyclic groups

The determination of orbits and stabilizers is one of the fundamental problems in algorithmic group theory. If the desired orbit is finite or, equivalently, the stabilizer has finite index in the given group, then we can list the orbit and, concurrently with its construction, calculate Schreier generators for the stabilizer. An outline of this well-known method is given in [8]. If the orbit is small, then this approach is quite effective. In general, in this process each orbit point is visited several times and, additionally, the set of Schreier generators is highly redundant.
If the acting group is polycyclic, then there is a more effective method to determine a finite orbit and its corresponding stabilizer which has been described in Section 4.2. This method uses an induction over a polycyclic sequence of the given group and exploits normal subgroups to build up orbits and stabilizers. The underlying observations of Section 4.1 on orbits and stabilizers of normal subgroups are fundamental for this chapter as well.
If the desired orbit is infinite, then this approach for finite orbits would not terminate. In general, if we have an infinite polycyclic group acting on an unstructured set, then there is no deterministic method known to compute orbits and stabilizers in this case or even to decide finiteness of an orbit. However, we observe that a randomized version of the general orbit stabilizer method using Schreier generators can still be very successful in this context as an implementation of such a randomized algorithm by Nickel in [23] shows.
If G is a polycyclic group acting as a group of automorphisms on a finitely generated abelian group A, then orbits and stabilizers of elements and subgroups of A under G can be computed using a deterministic method. This is a special case which has many applications in our later algorithms for polycyclic groups; for example, the fundamental problem of testing conjugacy of elements in polycyclic groups relies on it. In this chapter we present algorithms to solve the following problems for a polycyclic subgroup G of $Aut(A)$ where G acts on A by multiplication from the right.

- *The orbit stabilizer problem for elements:* for $a \in A$ determine a generating set for $Stab_G(a)$ or for $a, b \in A$ construct an element $g \in G$ with $ag = b$ if it exists.

- *The orbit stabilizer problem for subgroups:* for $S \leq A$ determine a generating set for $Stab_G(S) = N_G(S)$ or for $S, T \leq A$ construct an element $g \in G$ with $Sg = T$ if it exists.

- *The centralizer problem:* for $S \leq A$ construct a generating set of $C_G(S) = \{g \in G \mid ag = a \text{ for each } a \in S\}$.

As an initial step towards solving these problems we introduce a method to determine the kernel of a derivation in Section 7.1. Then we consider the orbit stabilizer problems for a polycyclic group acting on a free abelian group in Section 7.2. First, we reduce these problems to the same problems under action of a polycyclic p-congruence subgroup. Then, based on this setup, we present a solution to the stabilizer problem for elements of a free abelian group in Section 7.3. It may be worth noting here that this solution relies on the derivation kernel algorithm of Section 7.1, while the derivation kernel algorithm uses stabilizers of elements and thus may depend in certain cases on Section 7.3. However, whenever we apply the derivation kernel algorithm within the determination of a stabilizer of an element in a free abelian group, then the derivation kernel algorithm reduces to a finite orbit stabilizer computation which can be solved by the method of Section 4.2. This ensures that the derivation kernel algorithm of Section 7.1 and the element stabilizer method of Section 7.3 are not yielding an infinite recursion calling each other. Further, in Section 7.4, we introduce a solution to the stabilizer problem for subgroups of a free abelian group. Finally, in Section 7.5 we show that the introduced methods can be combined to solve the orbit stabilizer problems for elements and subgroups of a finitely generated abelian group as stated above.

References: Dixon investigated the orbit stabilizer problem for vectors in $\mathbb{Q}^d$ under the action of a finitely generated subgroup of $GL(d,\mathbb{Q})$ and presented a solution for nilpotent-by-finite groups in [16]. The ideas contained in Dixon's approach are fundamental for the methods outlined in this Chapter.
The decidability of the problems discussed in this chapter was proved by Baumslag et. al. in [3].
The methods presented in Section 7.3 to solve the orbit stabilizer problem for elements of free abelian groups under action of a polycyclic group are a joint project with Ostheimer. Further details and a report of an implementation are given in [25].

7.1 Affine actions and kernels of derivations

Let G be a group and A a G-module. As in Section 6.1 we denote a map $\delta : G \to A$ as a *derivation* if $(gh)^\delta = (g^\delta) \cdot h + h^\delta$ for all $g, h \in G$ and $1^\delta = 0$. The kernel of this derivation δ is then defined by $ker(\delta) = \{g \in G \mid g^\delta = 0\}$. The determination of the kernel of a derivation is a problem that has many applications in algorithmic group theory.
Let $\psi : G \to Aut(A)$ be the action of G on the G-module A. Then we can combine ψ and a derivation δ to an affine action ρ of G on A which is of the form $ag^\rho = ag^\psi + g^\delta$. Using this affine action corresponding to δ we can determine the kernel of δ as outlined in the following elementary lemma.

7.1. Lemma: *Let ρ be an affine action of G on A corresponding to the derivation δ. Then*

$$Stab_G(0) = \{g \in G \mid 0g^\rho = 0\} = \{g \in G \mid g^\delta = 0\} = ker(\delta).$$

Suppose that $A = R^d$ for a suitable ring R. Then the G-module action of G on A is of the form $\psi : G \to GL(d,R)$. Using this, we can describe an affine action of G on A via a derivation δ by a linear action of G on R^{d+1} of the following form:

$$\rho : G \to GL(d+1,R) : g \mapsto \left(\begin{array}{c|c} & 0 \\ g^\psi & \vdots \\ & 0 \\ \hline g^\delta & 1 \end{array}\right)$$

In this case Lemma 7.1 translates as $ker(\delta) = Stab_G(e)$ for $e = (0,\ldots,0,1) \in R^{d+1}$. Hence the determination of the kernel of a derivation can be translated into the computation of the stabilizer of the element e in R^{d+1} under a linear action of G. We consider this case in more detail as follows.

The *translation subgroup* of the affine matrix group G^ρ consists of those matrices g^ρ with $g^\psi = 1$; that is, if $K = ker(\psi)$, then K^ρ is the translation subgroup of G^ρ. Clearly, K is a normal subgroup of G. Hence we can use K in combination with the algorithm of Section 4.1 to reduce the effort inherent in the determination of $Stab_G(e)$. For this purpose we first solve the orbit and stabilizer problems for K and then we extend this solution to G. This is described in the following two sections.

7.1.1 The orbit stabilizer problem for the translation subgroup

As above, let ρ be an affine action of G with linear part $\psi : G \to GL(d, R)$ and derivation $\delta : G \to R^d$ and denote $K = ker(\psi)$. An easy computation yields that the restriction δ_K of δ to K is a homomorphism from the multiplicative group K into the additive group R^d. Thus its image $im(\delta_K)$ is a subgroup of R^d. The kernel and the image of δ_K yield the orbit and the stabilizer of e under K as follows.

$$Stab_K(e) = ker(\delta_K) \quad \text{and} \quad eK^\rho = \{(v, 1) \mid v \in im(\delta_K)\}$$

Hence it remains to compute the kernel and the image of δ_K. The image $im(\delta_K)$ is by construction a polycyclic subgroup of an abelian group and thus $im(\delta_K)$ is finitely generated abelian. Further, $im(\delta_K)$ is given by the images of the generators of K. In the cases we are interested in later, that is, $R = \mathbb{F}_p$ or $R = \mathbb{Q}$, we can compute a (lattice) basis for $im(\delta_K)$ from the generators and hence obtain a constructive polycyclic sequence for $im(\delta_K)$. Now $ker(\delta_K)$ can be determined using the methods of Section 3.5.

7.2. *Remark:*

a) If K is given by a polycyclic sequence, then $ker(\delta_K)$ can be determined using Lemma 3.13. In this case the kernel computation reduces to a nullspace computation over the integers. Further, the returned kernel has a polycyclic sequence as well.

b) If K is given by generators only, then we can use the methods of Section 3.5.3 to determine normal subgroup generators for $ker(\delta_K)$.

In our applications of this approach usually a polycyclic sequence for K is known. Thus polycyclic sequences for $im(\delta_K)$ and $ker(\delta_K)$ can be determined effectively using linear algebra methods only.

7.1.2 Extension to the general affine action

We now consider the solution of the orbit stabilizer problem for the vector $e = (0, \ldots, 0, 1)$ under an affine action $\rho : G \to GL(d + 1, R)$ with linear part ψ and derivation δ. We assume that we determined the orbit and stabilizer of e under the translation subgroup K by Section 7.1.1 and we want to extend this solution from K to G using the block stabilizer approach of Section 4.1.
As outlined in Section 4.1 we mainly have to compute the block stabilizer $Stab_G(eK^\rho)$. The stabilizer $Stab_G(e)$ can be read off from this block stabilizer using $Stab_K(e)$ and the solution to the orbit stabilizer problem in K. The block eK^ρ corresponds to $im(\delta_K) \leq R^d$. Since $(k^g)^\delta = (k^\delta)g^\psi$ for $k \in K$ and $g \in G$, we observe that $im(\delta_K)$ is invariant under the action of G.
Thus we can consider the induced linear action $\overline{\psi}$ on the factor $R^d/im(\delta_K)$ and the related affine action $\overline{\rho}$. Let $\overline{e} = (0, \ldots, 0, 1)$ in the affine space corresponding to the factor $R^d/im(\delta_K)$. Then we can obtain the desired block stabilizer as

$$Stab_G(eK^\rho) = Stab_G(\overline{e}).$$

In summary, the determination of $ker(\delta)$ reduces to the computation of a stabilizer of an element $\overline{e}$ in the affine space corresponding to $R^d/im(\delta_K)$ using the action of G/K on this affine space. The calculation of this element stabilizer may depend on G/K and on the ring R. For example, if G/K or $R^d/im(\delta_K)$ are finite, then we can use the method of Section 4.2 to determine this stabilizer. We will introduce methods to solve this remaining problem in other cases later.

We summarize the resulting approach to determine the kernel of a derivation $\delta : G \to R$ as follows.

DerivationKernel(G, K, ψ, δ)
  determine $ker(\delta_K)$ and $im(\delta_K)$ (Section 7.1.1)
  induce ψ and δ to the action on $R^d/im(\delta_K)$
  let $\overline{e} = (0, \ldots, 0, 1)$ in the affine space to $R^d/im(\delta_K)$
  compute generators $\mathcal{E}$ for $Stab_G(\overline{e})$ modulo $ker(\delta_K)$
  return $\langle \tilde{g} \mid g \in \mathcal{E} \rangle ker(\delta_K)$ (compare Section 4.1)

The following remark outlines two special situations which we will use in later applications and which admit a special solution to the block stabilizer computation.

7.3. *Remark:*

a) If G centralizes R^d via ψ, then $K = G$ and $Stab_G(e)$ can be determined by Section 7.1.1.

b) If $R = \mathbb{Z}$ and G acts irreducibly on the extended module $\mathbb{Q}^d$, then $im(\delta_K)$ is either trivial or has finite index in R^d. The the latter case we can apply the methods of Section 4.2 to determine the desired block stabilizer.

7.2 Actions on free abelian groups

In the beginning of this chapter we described the orbit stabilizer problems for a polycyclic group acting as a group of automorphisms on a finitely generated abelian group. Before we solve these problems in general in Section 7.5, we consider a special case for them here: the action of a polycyclic group on a free abelian group of finite rank. This special case will yield a major step towards solving the general problem.
Let G be a polycyclic group acting on the free abelian group $A = \mathbb{Z}^d$ via $\psi : G \to GL(d, \mathbb{Z})$. For our purposes we can identify G with the image of ψ and consider the orbit stabilizer problems for a subgroup G of $GL(d, \mathbb{Z})$. Since G is an integral polycyclic matrix group in this case, we can apply the methods of Chapter 5 to determine a polycyclic sequence for G. Additionally, these methods yield that the resulting polycyclic sequence exhibits a p-congruence subgroup $K_p(G)$ for an odd prime p. This subgroup is a normal subgroup of finite index in G. Hence we can use the block stabilizer method described in Section 4.1 to reduce the orbit stabilizer problem for G to the corresponding problems in $K_p(G)$.

7.4. *Remark:* In the application of Section 4.1 to the construction of $Stab_G(a)$ for $a \in A$ we need to determine the block stabilizer $Stab_G(aK_p(G))$. Since $Stab_G(aK_p(G)) \leq Stab_G(a + pA)$, we can split this computation in two steps: first we determine $H = Stab_G(a + pA)$ using the natural action of G on the elementary abelian group A/pA and then obtain the desired block stabilizer as $Stab_H(aK_p(G))$. A similar reduction applies to subgroups $S \leq A$.

It remains now to solve the orbit stabilizer problems for $K_p(G)$. As described in Section 5.4.3 we can determine an *irreducible block flag* $A = A_1 > \ldots > A_l > A_{l+1} = 0$ under the action of $K_p(G)$; that is, a series of $K_p(G)$-invariant subgroups through A such that each factor A_i/A_{i+1} is free abelian and irreducible as $\mathbb{Q}K_p(G)$-module. Theorem 5.1 observes that $K_p(G)$ acts as free abelian group on each factor A_i/A_{i+1}.
Our orbit stabilizer methods for $K_p(G)$ use induction on an irreducible block flag through A. For this induction purpose it will be useful to refine an irreducible block flag for $K_p(G)$ to one for a subgroup $H \leq K_p(G)$. The following lemma provides the basis for this refinement.

7.5. Lemma: *Let $G \leq GL(d,\mathbb{Z})$ such that $\mathbb{Q}(G)$ is an abelian algebra and the natural module $V = \mathbb{Q}^d$ is an irreducible $\mathbb{Q}(G)$-module. Let $H \leq G$. Then V is a homogeneous $\mathbb{Q}(H)$-module.*

Proof. The matrix algebra $\mathbb{Q}(G)$ is an abelian and irreducible. Thus each non-zero element of $\mathbb{Q}(G)$ is invertible by Schur's lemma. Hence also each non-zero element of the subalgebra $\mathbb{Q}(H)$ is invertible. Therefore, $\mathbb{Q}(H)$ is irreducible as H-module and V is homogeneous under action of $\mathbb{Q}(H)$. □

Let A_i/A_{i+1} be a factor of an irreducible block flag for $K_p(G)$ and let $H \leq K_p(G)$. Then A_i/A_{i+1} is homogeneous as $\mathbb{Q}H$-module and thus Lemma 5.11 yields that each non-trivial element of A_i/A_{i+1} is contained in an irreducible $\mathbb{Q}H$-submodule of this factor. Hence we can determine a direct factorization of A_i/A_{i+1} into irreducible $\mathbb{Q}H$-modules by spinning up elements of the factor under action of H.
This setup yields the basis for our algorithms to solve the orbit stabilizer problems for elements or subgroups of a free abelian group A under the action of a polycyclic group G. In the following two sections we now consider the two cases of elements or subgroups of A separately and describe solutions for their stabilizer problems. The orbit problems can be considered as dual and thus we omit the description of their solution here.

7.3 Stabilizers of elements in free abelian groups

In this section we describe a practical algorithm to solve the stabilizer problem for an element $a \in A = \mathbb{Z}^d$ under the action of a polycyclic group. As described in Section 7.2 it is sufficient to solve this problem for subgroups $G \leq K_p(GL(d,\mathbb{Z}))$ for an odd prime p.
Our method to determine $Stab_G(a)$ uses induction on an irreducible block flag $A = A_1 > \ldots > A_l > A_{l+1} = 0$ for G. We denote $B = A_l$ the last non-trivial subgroup in the flag and we assume by induction that we have determined $H = Stab_G(a+B)$. Then $Stab_G(a) \leq H$ and thus $Stab_G(a) = Stab_H(a)$. Hence it remains to compute $Stab_H(a)$.
By Lemma 7.5 we observe that B is a direct sum of irreducible $\mathbb{Q}H$-modules and we can exhibit such a direct factorization by spinning up vectors of B. Thus we obtain $B = B_1 \oplus \ldots \oplus B_m$ where each B_i is irreducible as $\mathbb{Q}H$-module. We can use projections on the direct summands and solve the stabilizer problem for each of the direct summands successively. This iteration yields the desired stabilizer eventually. Hence we assume without loss of generality that B itself is an irreducible $\mathbb{Q}H$-module.
The subgroup H stabilizes $a+B$ and thus we obtain for $h \in H$ that $ah \equiv a+b_h$ for some element $b_h \in B$. Thus we obtain a map $\delta : H \to B : h \mapsto b_h$. A straightforward computation shows that δ is a derivation of H and $Stab_H(a) = ker(\delta)$. The determination of such a kernel has been discussed in Section 7.1. We recall this approach in more detail as follows.
We identify $B = \mathbb{Z}^e$ and let $\nu : H \to GL(e,\mathbb{Z})$ be the action homomorphism of H on the free abelian group B. We consider $K = ker(\nu)$. Since B is irreducible as $\mathbb{Q}H$-module, we obtain that H^ν is a free abelian semisimple integral matrix group. Hence we can use the methods of Section 5.5.1 to determine the relation lattice for the generators of H^ν and then apply the algorithm of Section 3.5.4 to determine the kernel K. Once we obtained the kernel K, we can use the following theorem to solve the stabilizer problem for H.

7.6. Theorem: *We consider an affine action of a group H with linear part $\nu : H \to GL(e,\mathbb{Z})$ and derivation part $\delta : H \to \mathbb{Z}^e$. We suppose that H^ν is an abelian group which acts rationally irreducible on $\mathbb{Z}^e$ and we denote $K = ker(\nu)$. Then one of the following three cases occurs:*

a) $\mathbb{Z}^e/im(\delta_K)$ is finite.

b) $im(\delta_K) = 0$ and $ker(\delta) = H$.

c) $im(\delta_K) = 0$ and $ker(\delta) = K$.

Proof. As observed in Section 7.1, the subgroup $im(\delta_K)$ of $\mathbb{Z}^e$ is H-invariant. Since H acts rationally irreducible, we obtain that $im(\delta_K)$ has either finite index or is trivial. Suppose that $im(\delta_K) = 0$ and $ker(\delta) < H$. We have to show that $ker(\delta) = K$ and we are in case c). Since $im(\delta_K) = 0$, we have that $K \leq ker(\delta)$. On the other hand, $ker(\delta) < H$ and there exists an element $g \in H$ with $g^\delta \neq 0$. Let $h \in H \setminus K$ an arbitrary element. First note that $gh = hgk$ for some $k \in K$, since H/K is abelian. Thus, $(gh)^\delta = (hgk)^\delta = ((hg)^\delta)k^\nu + k^\delta = (hg)^\delta$, since $k^\nu = 1$ and $k^\delta = 0$. This yields $(g^\delta)h^\nu + h^\delta = (h^\delta)g^\nu + g^\delta$ and hence $(g^\delta)(h^\nu - 1) = (h^\delta)(g^\nu - 1)$. Both, g and h are elements which act non-trivially on $\mathbb{Z}^e$. Since H acts as abelian rationally irreducible group, we have that $\mathbb{Q}(H^\nu)$ is a field and thus $g^\nu - 1$ and $h^\nu - 1$ are both invertible. Thus $h^\delta = (g^\delta)(h^\nu - 1)(g^\nu - 1)^{-1} \neq 0$. Therefore, $h \notin ker(\delta)$ as desired. □

Theorem 7.6 yields a practical approach to determine the kernel of the derivation $\delta : H \to B$.
As initial test we check if δ is the trivial map; in this case we obtain that H stabilizes a and there is nothing to do. If δ is not trivial, then we determine $K = ker(\nu)$. Now we construct generators for the image $im(\delta_K)$ as described in Section 7.1.1. If this image has finite index in B, then we can determine $ker(\delta)$ as outlined in Section 7.1 using the finite orbit stabilizer algorithm as described in Remark 7.3. If $im(\delta_K) = 0$, then Theorem 7.6 c) allows us to read off $ker(\delta) = K$.

We summarize the resulting algorithm to determine $Stab_G(a)$ for a polycyclic subgroup G of $K_p(GL(d,\mathbb{Z}))$ and an element $a \in A = \mathbb{Z}^d$ as follows. We assume that G is given by a polycyclic sequence.

```
ElementStabilizer(G, a)
    assume that G ≤ K_p(GL(d,ℤ)) and let A = ℤ^d
    determine an irreducible block flag A_1, ..., A_{l+1} for G (Section 5.4.3)
    initialize H = G
    for i in {1, ..., l} do
        split A_i/A_{i+1} = B = B_1 ⊕ ... ⊕ B_m into ℚH-irreducibles (Lemma 7.5)
        for j in {1, ..., m} do
            define the derivation δ : H → B_j
            define the linear action ν : H → GL(B_j)
            if δ is not trivial then
                determine K = ker(ν) (Sections 5.5.1 and 3.5.4)
                construct the subgroup L = K^δ ≤ B_j (Section 7.1.1)
                if L has finite index in B_j then
                    compute H = ker(δ) using Remark 7.3 and the finite orbit stabilizer method
                else
                    set H = K
                end if
                refine the splitting of B_{j+1} ⊕ ... ⊕ B_m for H (Lemma 7.5)
            end if
        end for
    end for
    return the stabilizer H
```

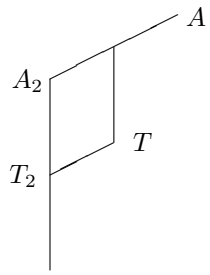
The input group G is given by a polycyclic sequence. The algorithms applied in the various steps of this method use a polycyclic sequence of their input group and return a polycyclic sequence for the output. Thus we obtain that intermediate subgroups H and K in the above algorithm are given by polycyclic sequences. We note that the considered polycyclic sequences need not be constructive for this application.

7.7. *Remark:* If G is a unipotent group, then the considered irreducible block flag is a series of subspaces whose factors are all one-dimensional trivial G-modules. Thus H induces a translation subgroup in each step and the above method reduces to solving linear equations as described in Section 7.1.1.

7.8. *Remark:* Let $G \leq K_p(GL(d,\mathbb{Z}))$ and $a \in A = \mathbb{Z}^d$. If $Stab_G(a)$ is a proper subgroup of G, then $Stab_G(a)$ has infinite index in G.

7.4 Stabilizers of subgroups of free abelian groups

In this section we describe an algorithm to solve the normalizer problem for a subgroup $S \leq A = \mathbb{Z}^d$ under the action of a polycyclic group. As described in Section 7.2 it is sufficient to solve this problem for polycyclic subgroups $G \leq K_p(GL(d,\mathbb{Z}))$ for an odd prime p.
Let T be the pure hull of the subgroup S in A. Then $N_G(S)$ normalizes T and $[N_G(T) : N_G(S)] \leq \infty$. Hence we can determine $N_G(S)$ from $N_G(T)$ using the methods of Section 4.2. Thus we can reduce to consider the normalizer problem for a pure subgroup T of A only.
Our approach uses induction on an irreducible block flag $A = A_1 > \ldots > A_l > A_{l+1} = 0$. We denote $T_i = T \cap A_i$ and we observe that the subgroups T_i are pure subgroups of A. By induction, we assume that we have computed $H = N_G(T_2)$. Then $N_G(T) \leq H$ and thus $N_G(T) = N_H(T)$. Hence it remains to compute $N_H(T)$. We induce the action of H to the factor space A/T_2 and we observe that $N_H(T) = N_H(T/T_2)$. We summarize the situation in the following picture.



Now we proceed in two steps to determine $N_H(T)$.

First step: We compute $K = N_H(T + A_2/A_2)$. By Lemma 7.5, the factor A/A_2 is a direct sum of irreducible $\mathbb{Q}H$-modules and a direct splitting of the factor is straightforward to determine by spinning up vectors. Using projections into the direct summands we can reduce the considered problem to an iterated computation of subgroup normalizers in irreducible $\mathbb{Q}H$-modules. An algorithm for this purpose is described in Section 7.4.2.

Second step: The computed subgroup K acts on the factor $T + A_2/T_2$. Using this action we can determine the normalizer $N_K(T)$ by exploiting the given complement situation. We outline a method for this case in Section 7.4.3. Since $N_H(T) \leq K$, we obtain $N_H(T) = N_K(T)$ and thus this second step yields the desired normalizer.

There are a number of special cases in which the considered normalizer computations can be simplified considerably. We include outlines of such special cases in the following sections and, in particular, we describe a method for the case that T/T_2 is one-dimensional in Section 7.4.1. Finally, a summary of the resulting method to determine $N_G(S)$ is given in Section 7.4.4.

7.4.1 The one-dimensional case

Let S be a one-dimensional subgroup of $A = \mathbb{Z}^d$ and let G be a polycyclic subgroup of $K_p(GL(d,\mathbb{Z}))$. In the following lemma we show that the determination of the normalizer $N_G(S)$ can be reduced to the construction of the stabilizer $Stab_G(a)$ for a generator $a \in S$. In turn, such a stabilizer can be computed using the methods of Section 7.3.

7.9. Lemma: *Let G be a subgroup of $GL(d,\mathbb{Z})$, let $a \in A = \mathbb{Z}^d$ and $S = \langle a \rangle$.*

a) $[N_G(S) : Stab_G(a)] \leq 2$ with equality if and only if there exists an element $g \in G$ with $ag = -a$. If such an element g exists, then $N_G(S) = \langle g, Stab_G(a) \rangle$.

b) If $G \leq K_p(GL(d,\mathbb{Z}))$ for an odd prime p, then $N_G(S) = Stab_G(a)$.

Proof. a) Let $g \in N_G(S)$. Then $ag \in S$ and hence $ag = ea$ for some $e \in \mathbb{Z}$. Since g is invertible, $e = \pm 1$. Thus the orbit of a under $N_G(S)$ has length at most 2 proving a).
b) Suppose that $g \in N_G(S)$ with $ag = -a$. Then g has an eigenspace for the value -1. However, g is of the form $g = 1 + ph$ for a matrix h and an odd prime p yielding a contradiction. □

7.4.2 The irreducible case

Let $G \leq K_p(GL(d,\mathbb{Z}))$ and $A = \mathbb{Z}^d$ the natural G-module. In this section we assume that G acts rationally irreducible on A and we want to determine $N_G(T)$ for a pure subspace $T \leq A$. Clearly, if T is one-dimensional, then we can apply the methods of Section 7.4.1. Further, if $\dim(T) > d/2$, then we can use the dual space to translate the problem into a normalizer computation for a subspace of smaller dimension $d - \dim(T) \leq d/2$. The following lemma contains another reduction for subspaces of certain dimensions.

7.10. Lemma: *Let $T \leq A = \mathbb{Z}^d$ a pure subgroup with $\dim(T) = e$. Suppose that $G \leq K_p(GL(d,\mathbb{Z}))$ acts rationally irreducible on A. If $\gcd(d,e) = 1$, then $N_G(T) = 1$.*

Proof. Let $H = N_G(T) \leq G$. Then by Lemma 7.5, the natural module A is a homogenous $\mathbb{Q}H$-module; that is, A is a direct sum of isomorphic irreducible $\mathbb{Q}H$-modules. Let $f = \dim(B)$ for an irreducible $\mathbb{Q}H$-submodule B of A. Then $f \mid d$ and $f \mid e$, since A and T are both a direct sum of isomorphic copies of B. If $\gcd(d,e) = 1$, then $f = 1$. Hence A is a direct sum of one-dimensional $\mathbb{Q}H$-modules and thus $H = 1$. □

Now we introduce an algorithm for the general case on T and A. Let $e = \dim(T)$ and consider the e-fold tensor power $T(d,e)$ of A. This is a G-module of dimension d^e under the diagonal action of G. We define the subspace D of $T(d,e)$ as the subspace generated by tensors $a_1 \otimes \ldots \otimes a_m$ with $a_i = a_j$ for some $i \neq j$. This subspace is G-invariant, since G acts diagonally. The factor $E(d,e) = T(d,e)/D$ is the e-fold *exterior power* of A. The group G acts naturally on this factor of the tensor power. The following lemma provides a link between the normalizer $N_G(T)$ and the stabilizer of a vector in $E(d,e)$. A similar construction has been used in [3].

7.11. Lemma: *Let $E(d,e)$ be the e-fold exterior power of $A = \mathbb{Z}^d$ and let $G \leq K_p(GL(d,\mathbb{Z}))$. We consider $T \leq A$ with basis $t_1, \ldots, t_e$. Then we obtain for $t = t_1 \otimes \ldots \otimes t_e + D \in E(d,e)$ that*

$$N_G(T) = Stab_G(t).$$

Proof. First we note that the diagonal action of G on $T(d,e)$ induces a subgroup of $K_p(GL(d^e,\mathbb{Z}))$ and thus the action of G on $E(d,e)$ induced a subgroup of the corresponding p-congruence subgroup as well. Hence $Stab_G(t) = N_G(\langle t\rangle)$ by Lemma 7.9. Now one can prove the desired lemma by a direct computation. □

Using Lemma 7.11 we can translate the normalizer of a subgroup to the stabilizer of a vector. The vector stabilizer can be determined using the method of Section 7.3.
We apply this approach to a subgroup G of $K_p(GL(d,\mathbb{Z}))$ which acts rationally irreducible on A. It is straightforward to observe that G acts as a semisimple integral matrix group on $E(d,e)$ in this case and G induces a subgroup of the p-congruence subgroup acting on $E(d,e)$. This simplifies the application of the method in Section 7.3 considerably. In fact, it remains to split the semisimple $\mathbb{Q}G$-module $E(d,e)$ into irreducibles as described in Section 5.3.1. Then we reduce the stabilizer computation into the irreducible summands and apply Theorem 7.6.
However, to use this approach we need to construct the explicit action of G on $E(d,e)$. Since the exterior power $E(d,e)$ has dimension $\binom{d}{e}$, this is only feasible for reasonably small values of d and e. The following lemma can be used to avoid this explicit construction in many cases.

7.12. Lemma: *Let $T \leq A = \mathbb{Z}^d$ a pure subgroup with $\dim(T) = e$. Suppose that $G \leq K_p(GL(d,\mathbb{Z}))$ acts rationally irreducible on A and let $c \in \mathbb{Q}(G)$ with $\mathbb{Q}(G) = \mathbb{Q}(c)$. By construction, $\mathbb{Q}(c)$ is an extension field of $\mathbb{Q}$ of degree d. Let $L = Gal(\mathbb{Q}(c)/\mathbb{Q})$. Then L acts faithfully on the roots of the minimal polynomial f of c and thus L embeds into the symmetric group S_d. If this permutation representation of L induces a transitive permutation representation on the set of e-subsets of $\{1,\ldots,d\}$, then $N_G(T) = 1$.*

Proof. Let $\overline{c}$ be the matrix corresponding to the action of c on $E(d,e)$. If $f = \prod_{i=1}^{d}(x - a_i)$ is the minimal polynomial of the matrix c, then $g = \prod_{i_1<\ldots<i_e}(x - a_{i_1}\cdots a_{i_e})$ is the minimal polynomial of $\overline{c}$. The Galois group L acts on the roots of g as on e-subsets of $\{1,\ldots,d\}$. If this action is transitive, then g is irreducible over $\mathbb{Q}$ and $E(d,e)$ is irreducible as module for $\mathbb{Q}(\overline{c})$. Thus $E(d,e)$ is also irreducible as $\mathbb{Q}G$-module in this case. By Lemma 7.11, $N_G(T) = Stab_G(t)$ for a non-trivial vector $t \in E(d,e)$. This yields by Theorem 7.6 that $N_G(T) = Stab_G(t) = 1$ if G acts rationally irreducible on $E(d,e)$. □

In practice, the Galois group of an irreducible polynomial is often the full symmetric group and thus acts transitively on e-subsets for all possible values of e. Hence Lemma 7.12 applies in many cases. To use Lemma 7.12 we need to compute the Galois group of $\mathbb{Q}(c)/\mathbb{Q}$. Here we can apply the methods of [27] which are effective for fields $\mathbb{Q}(c)$ up to degree $d \leq 15$ and are expected to be practical for higher degrees also.

7.4.3 The complement case

In this section we consider the case that G is a polycyclic subgroup of $K_p(GL(d,\mathbb{Z}))$ and the natural module $A = \mathbb{Z}^d$ can be written as $A = B \oplus T$ where B and T are pure subgroups of A and B is invariant under G. We describe a method to determine $N_G(T)$. First, we consider the special case that G centralizes A/B in the following lemma.

7.13. Lemma: *Let G be a polycyclic subgroup of $K_p(GL(d,\mathbb{Z}))$ and suppose that $\mathbb{Z}^d = A = B \oplus T$ where B is G-invariant. If G centralizes the factor A/B, then we obtain for a basis $\mathcal{T}$ of T that*

$$N_G(T) = \bigcap_{a\in\mathcal{T}} Stab_G(a).$$

Proof. Let $a \in T$ and define $C = \langle a, B\rangle \leq A$. Then G normalizes C and hence $N_G(T)$ normalizes $C \cap T = \langle a\rangle$. By Lemma 7.9 b) we obtain that $N_H(T) \leq Stab_H(a)$ for each $a \in \mathcal{T}$ as desired. □

In general, we can compute the normalizer of a complement as described in Lemma 6.4. For this purpose we have to determine the cohomology group $C = H^1(A/B, B)$. Since A is abelian, we have $C = Z^1(A/B, B)$ and this group can be represented as e-fold direct power B^e where $e = \dim(T)$. The direct power B^e can be considered as a free abelian group of rank $e(d-e)$. Then we have to consider the vector t in C corresponding to T and we induce the action of G to the affine action on C described in Lemma 6.4. Based on this, it remains then to compute the stabilizer of the element t under action of G. Here we can apply the element stabilizer computation of Section 7.1.

7.14. *Remark:* Using the algorithm of Section 5.4.3 we can refine the subfactors A/B and B by irreducible block flags for G. Then we can iterate the normalizer computation using these two series and break up one large stabilizer computation into a number of smaller ones.
More precisely, we consider a G-invariant flag $A = A_1 > \ldots > A_l > A_{l+1} = B = B_1 > \ldots > B_m > B_{m+1} = 0$ and define $T_{ij} = (T \cap A_i)B_j$. Then $T_{ij}/T_{i+1,j}$ is a complement to $T_{i+1,j-1}/T_{i+1,j}$ in $T_{i,j-1}/T_{i+1,j}$. Thus, once the subgroups $T_{i+1,j}, T_{i+1,j-1}$ and $T_{i,j-1}$ are normalized, we can apply the above method to this situation and determine the normalizer of T_{ij} acting on the factor $T_{i,j-1}/T_{i+1,j}$. This yields an inductive procedure which we use to normalize successively each subgroup in the lattice of subgroups T_{ij}.

7.4.4 A summary of the subgroup stabilizer algorithm

```
SubgroupStabilizer(G, S)
    assume that G ≤ K_p(GL(d,ℤ)) and denote A = ℤ^d with S ≤ A
    determine an irreducible block flag A_1, ..., A_{l+1} for G (Section 5.4.3)
    let T be the pure hull of S and initialize H = G
    for i in {1, ..., l} do
        let T_i = T ∩ A_i
        if dim(T_i/T_{i+1}) = 1 then
            determine H = N_H(T_i/T_{i+1}) using Section 7.4.1
        else
            let R = (T_i + A_{i+1})/A_{i+1}
            split A_i/A_{i+1} = B = B_1 ⊕ ... ⊕ B_m into ℚH-irreducibles (Lemma 7.5)
            for j in {1, ..., m} do
                let R_j = R ∩ B_j with dim(R_j) = e and dim(B_j) = d
                induce the action of H to B_j and let ℚ(c) = ℚ(H)
                if gcd(d, e) = 1 or Gal(ℚ(c)/ℚ) acts transitively on e-subsets then
                    determine H = N_H(R_j) as the kernel of the action of H on B_j (Lemma 7.10)
                else
                    determine H = N_H(R_j) using the exterior power (Lemma 7.11)
                end if
                refine the direct splitting B_{j+1} ⊕ ... ⊕ B_m for H (Lemma 7.5)
            end for
            induce the action of H to (T_i + A_{i+1})/T_{i+1}
            if H centralizes (T_i + A_{i+1})/A_{i+1} then
                compute H = N_H(T_i/T_{i+1}) by stabilizing a basis (Lemma 7.13)
            else
                compute H = N_H(T_i/T_{i+1}) using complements (Section 7.4.3)
            end if
        end if
    end for
    determine H = N_H(S) using the finite orbits method (Section 4.2)
    return the normalizer H of S in G
```

7.15. *Remark:* If G is a unipotent group, then the considered block flag is a series of subspaces whose factors are one-dimensional. In this case each inductive step of the above algorithm reduces to an application of Section 7.4.1. Hence this case is more effective than the general case of a triangularizable polycyclic group.

7.5 Actions on finitely generated abelian groups

Let G be a polycyclic group acting as a group of automorphisms on a finitely generated abelian group A. As an application of the previous sections we now introduce methods to compute the stabilizer of an element and the normalizer or centralizer of a subgroup of A under the action of G. This solves the stabilizer and centralizer problems described in the introduction of this chapter. The corresponding orbit determinations can be obtained by a similar approach.
First note that the set of all elements of finite order in A forms a subgroup of the abelian group A: the torsion subgroup T. If A is given by a constructive polycyclic sequence, then we can determine a polycyclic presentation of A as described in Lemma 2.1. By applying a Smith normal form computation to the relations of this presentation we obtain a generating set for the torsion subgroup T. Then A/T is free abelian and T is characteristic in A and thus invariant under the action of G.

- *The stabilizer problem for an element:* Let $a \in A$. We first apply the methods of Section 7.2 to determine the stabilizer $H = Stab_G(aT)$ acting on the free abelian group A/T. As $Stab_G(a) \leq H$, we obtain $Stab_G(a) = Stab_H(a)$. The orbit of a under H is finite, since for $h \in H$ we have $ah = at_h$ for some $t_h \in T$. Hence we can finish the stabilizer construction using the method of Section 4.2.

  We note that $\delta : H \to T : h \mapsto t_h$ is a derivation of H. Hence we can apply the method of Section 7.1 to improve the second stabilizer computation.

- *The stabilizer problem for a subgroup:* Let $S \leq A$. We first determine the stabilizer H of ST/T in G using the algorithm of Section 7.2. As $Stab_G(S) \leq H$, we obtain $Stab_G(S) = Stab_H(S)$. The orbit of S under H is finite and we determine it by two applications of Section 4.2: first we construct $K = Stab_H(S \cap T)$. Then we consider the action of K on $ST/(S \cap T)$ and normalize its subgroup $S/(S \cap T)$.

  We note that $S/(S \cap T)$ is a free abelian group complementing the finite group $T/(S \cap T)$ in $ST/(S \cap T)$. A reduction to the determination of normalizers of complements has been introduced in Chapter 6 and this can be applied here.

- *The centralizer problem:* Let $S \leq A$. Then we obtain $C_G(S)$ if we successively stabilize each element of a finite generating set of S using the solution to the stabilizer problem for elements.

7.16. *Remark:* If A is elementary or free abelian, then this approach to determine stabilizers reduces to the finite orbit stabilizer methods of Section 4.2 or the method for actions on free abelian groups in Section 7.2, respectively. Hence we can consider the above algorithm as a generalization of these two more special algorithms.

Chapter 8

General group theoretic investigations

We present a collection of general purpose algorithms designed to investigate the subgroup structure of a polycyclic group given by a constructive polycyclic sequence. In Chapter 3 we introduced a basic machinery to compute with subgroups and factor groups of such polycyclic groups. In particular, we have shown how to determine the size or the index of a subgroup and we have determined its Hirsch length. Here we consider a number of more advanced problems. In particular, we describe methods to

- construct complement classes and supplements of certain types.
- determine centralizers, normalizers and intersections of subgroups.
- test properties such as nilpotency or supersolvability.

Most of the algorithms in this chapter use an inductive approach on a series of subgroups of G. They either construct the desired information by proceedings upwards over a polycyclic series of G or downwards over an abelian normal series. A polycyclic series of G can be determined directly from the given constructive polycyclic sequence of G. The computation of algorithmically useful abelian normal series of G is discussed in Section 8.2.
Within each inductive step we try to reduce the considered problem such that it can be solved by one of the algorithms introduced in former chapters; that is, we try to solve the given problem using the available machinery for subfactors and homomorphisms of Chapter 3, using the cohomology methods of Chapter 6 or we apply the orbit stabilizer algorithms of Chapter 7.

References: Practical algorithms to compute complement classes, centralizers, normalizers and intersections in finite polycyclic groups are well-known. For example, Mecky & Neubüser [50] presented an algorithm to determine centralizers and conjugacy classes of elements. Celler, Neubüser & Wright [9] described a method to determine complement classes and they apply this method to construct normalizers. Further, Glasby & Slattery [28] introduced algorithms to compute intersections of subgroups and normalizers. All of these methods for finite groups use approaches which are similar to our algorithms.
A method to compute centralizers in finitely generated nilpotent groups is described by Sims [71]. Further, intersection and normalizer algorithms in finitely generated nilpotent groups are introduced by Lo in [44]. The decidability of the problems considered in this chapter had been known before. We refer to [3] for the algorithmic decidability of questions for polycyclic groups.

8.1 Commutator subgroups and commutator series

Let G be a group with two subgroups H and K. The commutator subgroup $[H, K]$ is defined as the subgroup $\langle [h, k] \mid h \in H, k \in K \rangle$. We recall that $[H, K] \trianglelefteq \langle H, K \rangle \leq G$. If H and K normalize each other, then we obtain further that $[H, K] \leq H \cap K$.
There are two important subgroup series which are defined by iterated commutator subgroups. The derived series of G is obtained as $G^{(1)} = G$ and $G^{(i+1)} = [G^{(i)}, G^{(i)}]$. This is a characteristic series in G with abelian factors. If G is polycyclic, then the derived series terminates at the trivial subgroup and each factor $G^{(i)}/G^{(i+1)}$ is a finitely generated abelian group.
The lower central series of G is defined by $\lambda_1(G) = G$ and $\lambda_{i+1}(G) = [G, \lambda_i(G)]$. It is also a characteristic series of G and its factors are abelian and central under the action of G. If G is polycyclic, then the factors are finitely generated abelian groups. If G is nilpotent, then this series terminates at the trivial subgroup, but otherwise, the lower central series may terminate at a non-trivial subgroup or it may not be a finite series.
The following lemma considers the determination of a commutator subgroup in a polycyclic group.

8.1. Lemma: *Let $\mathcal{H}$ and $\mathcal{K}$ be polycyclic sequence for H and K, respectively.*

a) $[H, K]$ is generated as normal subgroup in $\langle H, K \rangle$ by $[\mathcal{H}, \mathcal{K}] = \{[h^{\pm 1}, k^{\pm 1}] \mid h \in \mathcal{H}, k \in \mathcal{K}\}$.

b) If H and K normalize each other and $\mathcal{H}$ and $\mathcal{K}$ are both induced with respect to a parent polycyclic sequence $\mathcal{G}$, then $[H, K]$ is generated by $[\mathcal{H}, \mathcal{K}]$.

Proof. a) This follows from the commutator formulas $[gh, k] = [g, k]^h [h, k]$ and $[g, hk] = [g, k][g, h]^k$.
b) Let $\mathcal{G} = (g_1, \ldots, g_n)$ and $\mathcal{G}_2 = (g_2, \ldots, g_n)$. We denote $\mathcal{H}_2 = (h \in \mathcal{H} \mid \mathrm{dep}_{\mathcal{G}}(h) > 1)$ and, similarly, $\mathcal{K}_2 = (k \in \mathcal{K} \mid \mathrm{dep}_{\mathcal{G}}(k) > 1)$. Since $\mathcal{H}$ and $\mathcal{K}$ are induced with respect to $\mathcal{G}$, we obtain that $\mathcal{H}_2$ and $\mathcal{K}_2$ are induced with respect to $\mathcal{G}_2$. Further, with $G_2 = \langle \mathcal{G}_2 \rangle$ we observe that $H_2 = \langle \mathcal{H}_2 \rangle = H \cap G_2$ is normalized by K and, similarly, $K_2 = \langle \mathcal{K}_2 \rangle$ is normalized by H. Hence we can assume by induction that $N = [H_2, K_2]$ is generated by $[\mathcal{H}_2, \mathcal{K}_2]$. Note that N is normal in H and K and $[H, K]/N = [H/N, K/N]$. Thus it remains to show that $[H, K]/N$ is generated by $[\mathcal{H}/N, \mathcal{K}/N]$. To simplify notation we assume that $N = 1$ and hence H_2 and K_2 commute.
Next we show that $[H, K_2]$ is generated by $[\mathcal{H}, \mathcal{K}_2]$. If $\mathcal{H} = \mathcal{H}_2$, then $[H, K_2] = [H_2, K_2] = 1$ and there is nothing to prove. Thus suppose that there exists an element $h_1 \in \mathcal{H} \setminus \mathcal{H}_2$. Then each element x in H can be written as $x = w h_1^{\pm e}$ for $w \in H_2$ and $e \in \mathbb{N}_0$. Choose $h = h_1^{\pm 1}$ such that $x = wh^e$ and let $k \in K_2$. Then we obtain $[x, k] = [wh^e, k] = [w, k]^{h^e} [h, k]^{h^{e-1}} \cdots [h, k]^h [h, k] = [h, k^{h^{e-1}}] \cdots [h, k^h][h, k]$. The commutators arising as factors in the resulting product are all of the form $[h, z]$ for $z \in K_2$. Thus we can write z as a word in $\mathcal{K}_2$; that is, $z = k_2^{\pm e_2} \cdots k_r^{\pm e_r}$. Since $[h, k_i] \in G_2 \cap H = H_2$, we obtain that $[h, k_i]$ commutes with K_2 and thus $[h, z] = [h, k_2^{\pm 1}]^{e_2} \cdots [h, k_r^{\pm 1}]^{e_r}$. Hence we can write $[h, z]$ and thus $[x, k]$ as a word in $[\mathcal{H}, \mathcal{K}_2]$.
As above we can now assume that $[H, K_2] = 1$ and we consider $[H, K]$. By symmetry, we can use the same argument as above to show that $[H, K]$ is generated by $[\mathcal{H}, \mathcal{K}]$ and hence b) follows. □

Lemma 8.1 a) yields an algorithm to compute commutator subgroups. If H and K normalize each other and have induced polycyclic sequences, then we can determine $[H, K]$ more effectively using Lemma 8.1 b), since we avoid a normal closure computation in this case. However, the test if H and K normalize each other also costs time and thus Lemma 8.1 b) should probably only be used if it is known *a priori* that the subgroups H and K normalize each other.

```
CommutatorSubgroup(G, H, K)
    let 𝒢, ℋ and 𝒦 be polycyclic sequences for G, H and K, respectively
    compute the set [ℋ, 𝒦] and let C = ⟨[ℋ, 𝒦]⟩ (Section 3.1)
    if H and K normalize each other and ℋ and 𝒦 are induced w.r.t. 𝒢 then
        return C
    else
        return NormalClosure(⟨H, K⟩, C) (Section 3.3)
    end if
```

Straightforward applications of this algorithm include methods to compute the derived subgroup, the derived series and terms of the lower central series by their definition. Note that in all these applications the input subgroups are known to normalize each other.

8.2 Normal series with elementary or free abelian factors

Many algorithms for polycyclic groups G proceed by induction over a normal series of finite length with elementary or free abelian factors. Since the conjugation action of G on the factors of such a series yields computationally useful linear actions of G, we call such a series *a linear abelian series* for G. A polycyclic group G can have several linear abelian series. Here we introduce methods to determine an algorithmically useful one.
The effectiveness of many applications for linear abelian series depends on the number of subgroups in the series and the ranks of its factors. Hence the sum of these ranks yields a measure for the value of the series; that is, we want to determine linear abelian series with small rank sum. In a finite polycyclic group this rank sum is an invariant. But in an infinite polycyclic group only the rank sum of the infinite factors is invariant, while the rank sum of the finite factors is not bounded above.

8.2.1 Linear abelian series of abelian groups

First we recall well-known ideas for an abelian polycyclic group A. We assume that A is given by a constructive polycyclic sequence and hence we can determine a polycyclic presentation for A as in Lemma 2.1. Using a Smith normal form computation on the relators of this presentation, we can compute a direct decomposition of A as a product of cyclic groups. Thus let $A = C \times T$ where C is free abelian and T is the torsion subgroup of A and, moreover, let t be the exponent of T. Then we can derive two characteristic series of A:

$$A \rhd T \rhd 1 \quad \text{and} \quad A \rhd C^t \rhd 1.$$

To refine either of the two series to a linear abelian series we only need to refine the finite factor $T/1$ or A/C^t to elementary abelian factors. If we choose the ranks of the elementary abelian factors as large as possible, then the linear abelian series refining the first characteristic series is shortest possible for A. The linear abelian series refining the second characteristic series has the infinite factor at the end of the series.

8.2.2 Linear abelian series of arbitrary polycyclic groups

There are two issues that we need to consider when determining a linear abelian series in a polycyclic group. First, this algorithm will be used as an initial step in many algorithms for polycyclic groups and hence we need an effective approach to construct such a series. Secondly, we want to obtain a series with

reasonably small rank sum. Our proposed algorithm combines these two requirements and, additionally, it determines a series of characteristic subgroups in G.
We determine a linear abelian series $G = G_1 \rhd \ldots \rhd G_l \rhd G_{l+1} = 1$ by a top-down approach. Thus suppose that we computed G_i and let $G_i/G_i' = C/G_i' \times T/G_i'$ be the decomposition of G_i/G_i' as direct product of a free abelian and a finite abelian factor as in Section 8.2.1. We distinguish two cases in determining G_{i+1} from G_i.

- If G_i/G_i' is infinite, then G_i/T is a non-trivial free abelian group. In this case we choose $G_{i+1} = T$.
- If G_i/G_i' is finite, then we choose a prime p dividing the order of G_i/G_i' and we define G_{i+1} as the smallest subgroup of G_i whose factor G_i/G_{i+1} is an elementary abelian p-group.

```
LinearAbelianSeries(G)
    initialize L = {G} and H = G
    while H > 1 do
        determine H' (Section 8.1)
        decompose H/H' = C/H' × T/H' (Smith normal form)
        if H/T ≠ 1 then
            add T to L and reset H = T
        else
            choose p | [H : H'] and compute U = H'H^p
            add U to L and reset H = U
        end if
    end while
    return the series L
```

If the given group G is nilpotent, then the following theorem shows that the linear abelian series of G obtained by the above approach has minimal rank sum. For a proof of the theorem see [67, page 132].

8.2. Theorem: *Let G be an infinite nilpotent group. Then G/G' is infinite.*

Thus, if G is nilpotent, then we obtain a linear abelian series of G where the first factors are free abelian and the remaining factors are all finite. In particular, the obtained linear abelian series exhibits the torsion subgroup of the nilpotent group G.

8.3. *Remark:*

- Another natural approach to determine a linear abelian series of G would be to refine the derived series of G to elementary or free abelian factors. While this would also be effective, it tends to yield a series with larger rank sum.
- Linear abelian series with a variety of other useful properties are introduced in Section 9.8.

8.3 Complement classes and supplements

Let G be a polycyclic group with a normal subgroup N. The subgroup K of G is a *supplement* to N in G if $NK = G$. If additionally $K \cap N = 1$ holds, then K is a complement to N in G. In this section we discuss the determination of complements and supplements to N in G.
Our overall approach to the complement and supplement computations is to determine a G-invariant linear abelian series for the normal subgroup N and use induction down this series to calculate the desired subgroups. The following lemma recalls the well-known basis to this induction.

8.4. Lemma: *Let $A, N \trianglelefteq G$ with $A \leq N$. Let U/A be a complement to N/A in G/A and let K be a complement to A in U. Then K is a complement to N in G. (The same statement holds for supplements.)*

Thus we can reduce to the case that we compute complements to an elementary or free abelian normal subgroup A in a group U. In turn, this problem has been considered in Chapter 6 and its solution is provided by the first cohomology group.
In many later applications we are more interested in conjugacy classes of complements than in all complements. Hence we extend the induction approach of Lemma 8.4 to conjugacy classes as follows.

8.5. Lemma: *Let $A, N \trianglelefteq G$ with $A \leq N$.*

a) For each conjugacy class representative K of complements to N in G there exists a unique conjugacy class representative U/A of complements to N/A in G/A such that KA is conjugate to U in G.

b) Let U/A be a complement to N/A in G/A and $H/A = N_{G/A}(U/A)$. Then the G-conjugacy class representatives of complements K to N in G with KA conjugate to U coincide with the H-conjugacy class representatives of complements K to A in U with $KA = U$.

Proof. Part a) is obvious and we consider part b). Consider a G-conjugacy class of complements K to N with KA conjugate to U. Then $(KA)^g = K^gA = U$ for some $g \in G$. Hence the class contains a complement K^g with $K^gA = U$. It remains to show that any two such complements K_1 and K_2 are conjugate in H. By construction, there exists an element $g \in G$ with $K_1^g = K_2$ and $K_1A = U = K_2A$. Thus $U = K_2A = K_1^gA = (K_1A)^g = U^g$. Thus $g \in H$ as desired. □

Thus we can reduce the computation of conjugacy classes of complements to the determination of complements to an abelian normal subgroup under an action of a polycyclic group. This problem has been considered in Lemma 6.4 and it has been shown that it can be solved by computing orbits and stabilizers of elements in the abelian first cohomology group under an affine action. The determination of such orbits and stabilizer has been described in Chapter 7.

Let G be a polycyclic group with an abelian normal subgroup A. If A is infinite, then there may exist infinitely many complements to A in G. In this case we cannot possibly determine all complements explicitly. If we reduce to the determination of conjugacy classes of complements, then the finiteness conditions of Section 6.4 can often be used to prove that only finitely many classes exist. We will use these conditions to show that the following algorithm is successful in our later applications.

```
ComplementClasses(G, N)
    determine a G-invariant linear abelian series N = N_1 ▷ ... ▷ N_l ▷ N_{l+1} = 1 (Section 8.2)
    initialize C as the set of conjugacy classes {G^G}
    for i in {1, ..., l} do
        initialize K = ∅
        for each class C in C with representative U and H = N_G(U) do
            construct the cohomology group A = H^1(U/N_i, N_i/N_{i+1})
            if |A| = ∞, then return fail
            determine the H-orbits of elements of the abelian group A (Lemma 6.4 and Section 7.5)
            obtain a complement K/N_{i+1} for each orbit representative
            add the preimages K in G to the set K
        end for
        reset C = K
    end for
    return the set of classes C
```

8.3.1 Supplements and their conjugacy classes

An algorithm to compute supplements can be obtained as application of the complement method as we show in this section. We consider the case of supplements to an abelian normal subgroup only, as this is our main application for this method in later sections.
Let K be a supplement to the abelian normal subgroup A in a group U. Then $L = A \cap K$ is a normal subgroup of U and K/L is a complement to A/L in U/L. Thus we compute supplements in a polycyclic group U by first determining the desired U-normal subgroups L of A and then applying the complement algorithm.
Let H be a group normalizing U and A. If we want to compute H-classes of supplements only, then we first determine orbits and stabilizers of the desired U-normal subgroups of A under the action of H and then compute complement classes under the action of the stabilizers.

8.4 Intersections of subgroups

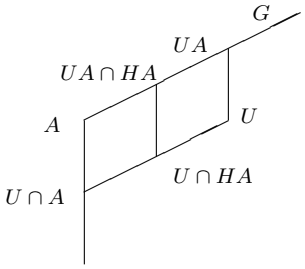
Let U and H be two subgroups of a polycyclic group G given by a constructive polycyclic sequence. We want to determine the intersection $U \cap H$. First, we consider a special case.

8.6. *Remark:* If U normalizes H or vice versa, then $U \cap H$ can be described as the kernel of the natural homomorphism $U \to UH/H : u \mapsto uH$. An induced polycyclic sequence for such a kernel can be obtained effectively using the algorithm of Section 3.5.

In the general case for U and H we introduce an algorithm which uses induction down a linear abelian series of G. Thus in the inductive step we assume that we have an elementary or free abelian normal subgroup A of G and we have determined $UA/A \cap HA/A = (UA \cap HA)/A$. To derive $U \cap H$ we proceed in two steps which we outline in the following two sections.

8.4.1 Determining $U \cap HA$ from $UA \cap HA$

We consider the chain of subgroups $1 \leq U \cap A \leq A \leq UA \cap HA \leq UA$. We observe that $U \cap A$ and A are both normal in UA, since A is abelian. Further, $U/U \cap A$ is a complement to $A/U \cap A$ in $UA/U \cap A$. We summarize the situation by the following picture.



By our setup, constructive polycyclic sequences for A and for U are given. These induce a constructive polycyclic sequence for UA/A as described in Section 3.4. Thus we can apply the methods of Section 3.5 to the natural epimorphism $\iota : U \to UA/A : u \mapsto uA$. In particular, we can compute the preimage of $(UA \cap HA)/A$ under ι. Since

$$(U \cap HA)^\iota = (U \cap HA)A/A = (UA \cap HA)/A,$$

we obtain that $((UA \cap HA)/A)^{\iota^{-1}} = U \cap HA$ as desired.

8.4.2 Determining $U \cap H$ from $U \cap HA$

To shorten notation we define $K = U \cap HA$ and we note that $U \cap H = K \cap H$.
In general, we can determine an intersection $K \cap H$ using a stabilizer computation: we let K act by right multiplication on the cosets G/H and obtain $K \cap H = Stab_K(H)$ where H is the trivial coset. We refine this general approach using our given situation.
We consider the action of K by right multiplication on the trivial coset H. If $g \in K$, then $g \in HA$ by definition of K, say $g = h_g a_g$ for $h_g \in H$ and $a_g \in A$. We obtain $Hg = Ha_g$. The element a_g is not unique in A, but its coset $a_g(A \cap H)$ is. Thus we can read off a map $\delta : K \to A/(A \cap H) : g \mapsto a_g(A \cap H)$. Note that $A \cap H$ can be computed effectively using Remark 8.6, since A is normalized by H.
A straightforward computation shows that δ is a multiplicatively written derivation; that is, $(gh)^\delta = (g^\delta)^h h^\delta$. The stabilizer of the trivial coset H can now be interpreted as the kernel of δ. This, in turn, has been discussed in Section 7.1 yielding that we can determine $ker(\delta)$ as the stabilizer of the trivial vector e under an affine operation of K on $A/A \cap H$. Thus we obtain

$$U \cap H = K \cap H = ker(\delta) = Stab_K(e).$$

The stabilizer of the vector e under action of K can now be determined as described in Chapter 7.

8.4.3 A summary of the intersection algorithm

Intersection(G, U, H)
  determine a linear abelian series $G = G_1 \rhd \ldots \rhd G_l \rhd G_{l+1} = 1$ (Section 8.2)
  initialize $I = G$
  for i in $\{1, \ldots, l\}$ do
    let $\iota : UG_{i+1}/G_{i+1} \to UG_i/G_i : uG_{i+1} \mapsto uG_i$
    determine the preimage K/G_i of I/G_i under ι (Section 3.5)
    compute $L/G_{i+1} = G_i/G_{i+1} \cap HG_{i+1}/G_{i+1}$ (Remark 8.6)
    construct $\delta : K/G_{i+1} \to G_i/L$ with $Hg \equiv Hg^\delta \bmod L$ for $g \in K$
    calculate $ker(\delta)$ and reset $I = ker(\delta)$ (Section 7.1)
  end for
  return the intersection I

8.7. *Remark:*

- We can simplify the determination of $ker(\delta)$ as described in Section 7.1 using the translation subgroup in the affine action. This translation subgroup corresponds to $C_K(G_i/L)$. In turn, this centralizer or a large subgroup of it are exhibited by the linear abelian series of Section 9.8.

- If G is nilpotent, then we can use a linear abelian series for G with central factors. In this case the computation of $ker(\delta)$ can be performed using linear algebra techniques as outlined in Section 7.1.

8.5 Centralizers and conjugacy of elements

Let G be a group and let $g \in G$. Then the centralizer $C_G(g) = \{h \in G \mid hg = gh\}$ of g in G is the stabilizer of g in G under the natural conjugation action of G on itself. The orbit of g under this action is the conjugacy class g^G of g. Thus the determination of a centralizer $C_G(g)$ and the corresponding conjugacy class g^G are dual problems and a solution to one of these two problems can be translated into

a solution of the other. Thus it is sufficient to consider one of these problems and we describe a method to determine the centralizer $C_G(g)$ of an element g for a polycyclic group G below.
The centralizer of a subgroup $U \leq G$ can be determined by successively centralizing each element of a generating set of U. Thus the centralizer of a finitely generated subgroup U of G can be derived from the centralizer of an element. If U is an elementary or free abelian normal subgroup of G, then the following approach may be more effective: first, we compute a constructive polycyclic sequence for the action of G on U using the methods of Chapter 5 and then we determine $C_G(U)$ as the kernel of this action by Section 3.5.

8.5.1 Centralizers and stabilizers under affine actions

Let G be a polycyclic group given by a constructive polycyclic sequence and let $g \in G$. Our method to determine the centralizer $C_G(g)$ of g in G proceeds by induction down a linear abelian series of G. Hence for the inductive step we assume that we have an elementary or free abelian normal subgroup A of G and we have determined $U \leq G$ with $U/A = C_{G/A}(gA)$. We want to compute $C_G(g) = C_U(g)$.
Since U centralizes g modulo A, for each $g \in U$ there exists an element $a_g \in A$ such that $gu = ga_g$. The map $\delta : U \to A : g \mapsto a_g$ satisfies $(uv)^\delta = (u^\delta)^v v^\delta$ and thus δ is a multiplicatively written derivation. We obtain

$$C_G(g) = ker(\delta).$$

This kernel can be obtained using a stabilizer computation under an affine action of U as outlined in Section 7.1. We consider this stabilizer construction in more detail in the following section.

8.5.2 Refining the kernel computation

In Section 7.1 we described a method to determine the kernel of a derivation using a stabilizer computation under an affine action of U on A. This affine action of U on A is derived by combining δ with the linear action of U on A. We identify $A \cong R^d$ where R is either a finite field or $R = \mathbb{Z}$ and let $e = (0, \ldots, 0, 1) \in R^{d+1}$ be the trivial affine vector over R^d. Then we observe that

$$ker(\delta) = Stab_U(e).$$

In Section 7.1 we introduced a refinement of the affine orbit stabilizer problem using $K = C_U(A)$ whose affine action on A induces the translation subgroup of the affine action of U. In particular, the stabilizer under action of K can be computed using linear algebra methods.
This refinement depends on a method to determine K or a possibly large U-normal subgroup of K. We remark here that we we gain more control on the centralizer K if we use a linear abelian series as introduced in Section 9.8 for our induction.

8.5.3 A summary of the centralizer algorithm

```
Centralizer(G, g)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize U = G
    for i in {1, ..., l} do
        construct δ : U → G_i/G_{i+1} : u ↦ [g, u]G_{i+1}
        determine ker(δ) and reset U = ker(δ) (Section 7.1)
    end for
    return the centralizer U
```

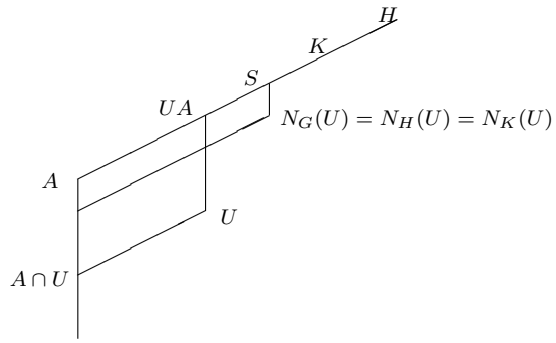
8.8. *Remark:* If G is nilpotent, then we can use a linear abelian series for G which has central factors. In this case the computation of $ker(\delta)$ can be performed using linear algebra techniques only as outlined in Section 7.1.

8.6 Normalizers and conjugacy of subgroups

Let G be a group and let $U \leq G$. Then the normalizer $N_G(U) = \{g \in G \mid gU = Ug\}$ of U in G is the stabilizer of U in G under the natural conjugation action of G. The orbit of U under this action is the conjugacy class U^G of U. Thus the determination of a normalizer and its corresponding conjugacy class are dual problems. A solution to one of these two problems can be translated into a solution of the other. We only consider a normalizer algorithm here.

8.6.1 Normalizers and stabilizers of linear and affine actions

Let G be a polycyclic group with a constructive polycyclic sequence. We use induction down a linear abelian series of G to determine $N_G(U)$. Thus in the inductive step we assume that A is an elementary or free abelian normal subgroup of G and we determined $H/A = N_{G/A}(UA/A)$. We observe that that $N_G(U) \leq H$ and thus $N_G(U) = N_H(U)$. We determine the desired normalizer $N_G(U)$ in two steps.
First, we compute $K = N_H(U \cap A)$. Since A is an elementary or free abelian normal subgroup of G, we can consider $U \cap A$ as a subspace of A and use the linear action of H on A to construct the normalizer as described in Chapter 7. Since $N_G(U)$ normalizes U and A and thus $U \cap A$, we obtain $N_G(U) \leq K$. Hence $N_G(U) = N_K(U)$.
Now we observe that $U/U \cap A$ is a complement to $A/U \cap A$ in $UA/U \cap A$. Since K normalizes UA, A and $U \cap A$, we can determine $N_G(U)$ as described in Lemma 6.4 c). More precisely, we first determine the stabilizer S in K of the element in $H^1(UA/A, A/(U \cap A))$ corresponding to $U/(U \cap A)$ using the affine action of K on the cohomology group. Then it remains to lift the determined stabilizer S to the normalizer $N_G(U)$.
We summarize the situation in the following picture.



8.6.2 A summary of the normalizer algorithm

We summarize the resulting algorithm to compute the normalizer of a subgroup U in a polycyclic group G as follows.

```
Normalizer(G, U)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize H = G
    for i in {1, ..., l} do
        construct L/G_{i+1} = U/G_{i+1} ∩ G_i/G_{i+1} (Remark 8.6)
        compute K = N_H(L/G_{i+1}) (Chapter 7)
        determine the affine action of K on H^1(UG_i/G_i, G_i/L)
        compute the element δ corresponding to U/L in H^1(UG_i/G_i, G_i/L)
        calculate the stabilizer of δ in K (Lemma 6.4 c))
        lift the stabilizer to the desired normalizer N_G(UG_{i+1}/G_{i+1})
        replace H by this normalizer
    end for
    return the normalizer H
```

The normalizer algorithm uses two stabilizer computations in each inductive step. Again, we can apply the ideas of the Sections 4.1 and 7.1 to simplify these stabilizer computations. In particular, the action of K the first cohomology group is an affine action and hence can be simplified using the translation subgroup of this affine action.

8.9. *Remark:* If G is nilpotent, then we can use a linear abelian series for G which has central factors. In this case we obtain $N_H(L/G_{i+1}) = H$ and thus this normalizer computation is redundant.

8.7 Testing nilpotency and supersolvability

Each finitely generated abelian group, each finitely generated nilpotent group and each supersolvable group is polycyclic, while the converse is not necessarily true. Testing these properties for a polycyclic group is interesting in its own right and it might also lead to improvements in further computations with this group, since, for example, information in abelian groups is usually much easier to obtain than in polycyclic groups in general.
It is quite straightforward to test if a group G given by generators is abelian by testing $gh = hg$ for each pair of generators. Algorithms to check nilpotency and supersolvability in a polycyclic group G given by a constructive polycyclic sequence are outlined in the following sections.

8.7.1 Testing nilpotency

Recall that a group G is nilpotent if and only if the lower central series of G is a finite series which terminates at the trivial subgroup. We could attempt to determine the lower central series of a polycyclic group G as described in Section 8.1, but since this series might have infinite length in a polycyclic group, this does not yield an algorithm to check nilpotency. We base our algorithm on the following theorem which is a stronger version of Theorem 8.2.

8.10. Theorem: *Let G be a group.*

a) $exp(\lambda_i(G)/\lambda_{i+1}(G)) \mid exp(\lambda_{i-1}(G)/\lambda_i(G))$.

b) If G is finitely generated nilpotent and $\lambda_{i-1}(G)/\lambda_i(G)$ is finite, then $\lambda_i(G)$ is finite.

Proof. a) To simplify notation we assume without loss of generality that $\lambda_{i+1}(G) = 1$ and let $e = exp(\lambda_{i-1}(G)/\lambda_i(G))$. Since $\lambda_i(G)$ is central in G, we obtain for $g \in G$ and $h, k \in \lambda_{i-1}(G)$ that $[g, hk] = [g, h][g, k]$. Thus $[g, h]^e = [g, h^e]$ and, further, $[g, h^e] = 1$, since $h^e \in \lambda_i(G)$. Hence $[g, h]$ has order dividing

e. Since the set of such commutators generates the abelian group $\lambda_i(G)$, we obtain that $exp(\lambda_i(G)) \mid e$.
b) If $\lambda_{i-1}(G)/\lambda_i(G)$ has finite exponent, then each factor $\lambda_{j-1}(G)/\lambda_j(G)$ for $j > i$ has finite exponent by part a) and thus is finite. Hence $\lambda_i(G)$ is finite. □

Thus based on Theorem 8.10 we obtain an algorithm to test nilpotency in a polycyclic group G as summarized below. This algorithm returns true if the lower central series of G reaches the trivial subgroup after finitely many steps. It returns false, if the lower central series terminates at a non-trivial subgroup or if the condition of Theorem 8.10 b) is violated.

```
IsNilpotent(G)
  let H = G
  repeat
    set K = [G, H]
    if |K| = 1, then return true
    if K = H, then return false
    if [H : K] < ∞ and |K| = ∞, then return false
    reset H = K
  until false
```

8.11. *Remark:* Another method to check nilpotency is provided by the Fitting subgroup algorithm in Section 9.3, since a polycyclic group G is nilpotent if and only if $G = Fit(G)$.

8.7.2 Testing supersolvability

The group G is supersolvable if it has a normal series of finite length with cyclic factors. In the following lemma we recall an elementary observation which yields an inductive approach to determine supersolvability in a polycyclic group.

8.12. Lemma: *Let G be a polycyclic group with an abelian normal subgroup A. Then G is supersolvable if and only if G/A is supersolvable and there exists a G-invariant series $A = A_1 \rhd \ldots \rhd A_l \rhd A_{l+1} = 1$ with cyclic factors A_i/A_{i+1}.*

Thus we can use induction over a linear abelian series of G to check if G is supersolvable. In the inductive step we consider an elementary or free abelian normal subgroup A of G and we check if there exists a G-invariant series of finite length with cyclic factors through A. If A is elementary abelian, then methods for this purpose are well-known: we can consider A as a G-module over a finite field and compute a G-module composition series in A. It then remains to test if all factors of this series are one-dimensional. In the free abelian case for A we use the following lemma.

8.13. Lemma: *Let G be a group with G-module $A = \mathbb{Z}^d$. There exists a G-invariant series of finite length with cyclic factors through A if and only if $G'G^2$ acts as a unipotent group on A.*

Proof. Denote $N = G'G^2$.
⇐ Suppose there exists a G-invariant series of finite length with cyclic factors through A. If each factor of this series is infinite cyclic, then G acts as a group of order two on each factor of the series. Hence N centralizes the series in this case and acts therefore as a unipotent group on A. Thus it remains to show that we can always obtain a series with infinite cyclic factors only. Let B be the last non-trivial subgroup in an arbitrary G-invariant series with cyclic factors of A and let C be its pure hull. Then C is infinite cyclic and G-invariant. Now we apply induction to A/C to obtain the desired series.
⇒ Vice versa, suppose that N acts as a unipotent group on A. Then the ascending series of fixed point

spaces in A under N is a G-invariant series of subspaces through A which is centralized by N. Hence G acts as finite elementary abelian 2-group on each factor of the series. By Maschke's theorem, each factor is a semisimple $\mathbb{Q}G$-module and thus a direct sum of one-dimensional modules. Hence each factor can be refined to a G-invariant series with cyclic factors. □

We obtain the following algorithm to test supersolvability in polycyclic groups.

```
IsSupersolvable(G)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    for i in {1,...,l} do
        if G_i/G_{i+1} is elementary abelian then
            compute a G-module composition series in G_i/G_{i+1} (Meataxe)
            if there is a composition factor of dimension at least two, then return false
        else
            determine N = G'G^2 (Section 8.1)
            compute the ascending N-central series in G_i/G_{i+1} (using fixed points)
            if this series terminates at a proper subgroup of G_i/G_{i+1}, then return false
        end if
    end for
    return true
```

Chapter 9

Structure theory for polycyclic groups

The investigation of the group theoretic structure in polycyclic groups was initiated by Hirsch in the 1930s and it has continued in various places. Its results provide a variety of interesting properties of polycyclic groups. We present a suite of algorithms designed to exhibit such aspects of the structure of a polycyclic group G. As before, we assume that the polycyclic group G is given by a constructive polycyclic sequence. In particular, we introduce methods to

- exploit the lattice of finite subgroups of G.
- compute subgroups of finite index in G and discuss the subgroup growth of G.
- calculate the Fitting subgroup of G.
- construct the centre and the FC-centre of G.
- exhibit the nilpotent-by-abelian-by-finite structure of G and related investigations.

As the methods in the previous chapter, most of these algorithms use an inductive approach on a polycyclic series or an abelian normal series of G. In each inductive step we reduce the problem that we consider to one which can be solved by the methods of the previous chapters.

References: Many of the problems considered in this chapter had been known to be decidable, see [3]. A large number of the practical algorithms introduced here have also been described in [17] and [18].

9.1 Finite subgroups

An infinite polycyclic group may have infinitely many finite subgroups. But Mal'cev proved in [49] that there exist only finite many conjugacy classes of finite subgroups in a polycyclic group. In Section 9.1.1 we present a method to determine the conjugacy classes of finite subgroups for a polycyclic group G given by a constructive polycyclic sequence.
Let $T(G)$ be the set of all finite order elements in a polycyclic group G. If G is nilpotent, then $T(G)$ is a subgroup of G, but otherwise $T(G)$ need not be a subgroup. If $T(G)$ is a subgroup, then it is a characteristic subgroup of G and it contains all finite subgroups of G. In Section 9.1.2 we describe a method to determine whether $T(G)$ forms a subgroup of G and to compute it in this case. Similarly, we construct the maximal finite normal subgroup $TN(G)$ of G or check if G is torsion-free.

9.1.1 Computing the conjugacy classes of finite subgroups

Let G be a polycyclic group given by a constructive polycyclic sequence. We use induction down a linear abelian series of G to determine the classes of finite subgroups. Thus for the inductive step we consider the last term A in this series of G and we suppose by induction that we know the conjugacy classes of finite subgroups of G/A. The following lemma is proved in the same way as Lemma 8.5.

9.1. Lemma: *Let G be a polycyclic group with abelian normal subgroup A. Suppose that the subfactors $U_1/A, \ldots, U_l/A$ are representatives for the conjugacy classes of finite subgroups of G/A. Furthermore, for $1 \leq j \leq l$ let $H_j/A = N_{G/A}(U_j/A)$.*
Then representatives for the conjugacy classes of finite subgroups of G can be obtained as representatives for the H_j-classes of finite subgroups K of U_j with $KA = U_j$ for $1 \leq i \leq j$.

Suppose that U/A is a finite subgroup of G/A with normalizer H/A. To determine the H-classes of finite subgroups K in U with $KA = U$ we distinguish two cases.

- A is elementary abelian: Then each subgroup of U is finite and hence we need to determine H-classes of supplements to A in U. A solution to this problem was described in Section 8.3.1.
- A is free abelian: Then each finite subgroup K of U satisfies $A \cap K = 1$ and hence we need to compute the H-classes of complements to A in U. An algorithm for this purpose was outlined in Lemma 6.4. Note that U/A is finite and hence there are only finitely many such complement classes by Lemma 6.11.

Hence we obtain the following algorithm to compute representatives for the conjugacy classes of finite subgroups in a polycyclic group G which is given by a constructive polycyclic sequence.

```
FiniteSubgroupClasses(G)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize C as the set of conjugacy classes {G^G}
    for i in {1, ..., l} do
        initialize K = ∅
        for each class C in C do
            let U be a representative of C with H = N_G(U)
            if G_i/G_{i+1} is finite, then determine the H-classes of supplements to G_i/G_{i+1} in U/G_{i+1}
                (Section 8.3.1)
            if G_i/G_{i+1} is infinite, then determine the H-classes of complements to G_i/G_{i+1} in U/G_{i+1}
                (Lemma 6.4)
            add the preimages in G for the new classes to the list K
        end for
        reset C = K
    end for
    return the set of finite subgroup classes C
```

9.1.2 Determination of $T(G)$ and $TN(G)$

Let G be a polycyclic group given by a constructive polycyclic sequence. If $T(G)$ is a subgroup of G, then we determine it using induction up a polycyclic series of G. We assume that this series contains factors of infinite or prime order only. The following lemma yields the inductive step.

9.2. Lemma: *Let G be a polycyclic group and $N \lhd G$.*

a) $T(G) \cap N = T(N)$.

b) If G/N is infinite cyclic, then $T(G) = T(N)$.

c) Suppose that G/N is cyclic of prime order and $T(N)$ is a group.

i) If $N/T(N)$ has no complement in $G/T(N)$, then $T(G) = T(N)$.

ii) If $N/T(N)$ has a unique complement $K/T(N)$ in $G/T(N)$, then $T(G) = K$.

iii) Otherwise $T(G)$ is not a subgroup.

Proof. a) Trivial.
b) Let $g \in G$ be of finite order. Then $gN = N \in G/N$, since G/N has no non-trivial finite subgroup. Therefore $g \in N$ and hence $g \in T(N)$.
c) By passing to $G/T(N)$ we may suppose $T(N) = 1$. Let $g \in G$ be of finite order. Since N is torsion-free, we obtain that $\langle g \rangle$ is a complement to N in G.
i) In this case no element of finite order exists in G and hence $T(G) = 1$.
ii) If there is just one complement K, then this complement is the unique maximal finite subgroup of G. Therefore, $T(G) = K$.
iii) If there are at least two complements K_1 and K_2, then $\langle K_1, K_2 \rangle$ contains an element from N and thus an infinite element. On the other hand, if $T(G)$ is a subgroup, then $K_1, K_2 \leq T(G)$ yielding a contradiction. □

Thus in the inductive step over a cyclic factor of prime order we have to compute the conjugacy classes of complements to $N/T(N)$ in $G/T(N)$. Since G/N is finite in this case, there are only finitely many complement classes by Lemma 6.11 and hence we can compute them using the method of Section 8.3. Note that G/N is a one-generator and one-relator group and thus the complement computation will be very effective. It will be an advantage for the algorithm to have a polycyclic series with few finite factors.

```
TorsionSubgroup(G)
    let G = G_1 ▷ ... ▷ G_n ▷ G_{n+1} = 1 be a polycyclic series with infinite or prime factors
    initialize T = 1 such that T = T(G_{n+1})
    for i in {n, ..., 1} do
        if G_i/G_{i+1} is finite then
            compute the complements to G_{i+1}/T in G_i/T (Section 8.3)
            if there is just one complement K/T, then reset T = K
            if there exist more than one complement, then return false
        end if
    end for
    return the torsion subgroup T of G
```

By definition $TN(G)$ is the largest normal finite subgroup of G. It is the product of all finite normal subgroups of G and thus a characteristic subgroup of G. If $T(G)$ is a subgroup in G, then $T(G) = TN(G)$. But $TN(G)$ might be a non-trivial subgroup in G even if $T(G)$ is not a subgroup. We determine $TN(G)$ using a similar approach as for $T(G)$ as shown in the following lemma. We omit its proof which is similar to the one of Lemma 9.2.

9.3. Lemma: *Let G be a polycyclic group and $N \lhd G$.*

a) $TN(G) \cap N = TN(N)$.

b) If G/N is infinite cyclic, then $TN(G) = TN(N)$.

c) Suppose that G/N is cyclic of prime order. If there exists a normal complement $K/TN(N)$ to $N/TN(N)$ in $G/TN(N)$, then $TN(G) = K$. Otherwise $TN(G) = TN(N)$.

In the inductive step of Lemma 9.3 c) we need to check if there exists a normal complement to $N/TN(N)$ in $G/TN(N)$. Note that in this case there exists at most one such complement, since $N/TN(N)$ cannot contain any normal finite subgroup.

```
NormalTorsionSubgroup(G)
    let G = G_1 ▷ ... ▷ G_n ▷ G_{n+1} = 1 be a polycyclic series with infinite or prime factors
    initialize T = 1 such that T = TN(G_{n+1})
    for i in {n, ..., 1} do
        if G_i/G_{i+1} is finite then
            compute the normal complements to G_{i+1}/T in G_i/T (Section 8.3)
            if there exists a normal complement K/T then reset T = K
        end if
    end for
    return the normal torsion subgroup T of G
```

Finally, the following remark can be used to check if a polycyclic group G is torsion-free using the torsion subgroup algorithm as outlined in Lemma 9.2.

9.4. *Remark:* A group G is torsion-free if and only if $T(G)$ is the trivial subgroup of G.

9.2 Subgroups of finite index

The subgroups of finite index play an interesting role in the theory of polycyclic groups. For example, a polycyclic group G is residually finite; that is, the subgroups of finite index in G intersect in the trivial subgroup of G. Moreover, each subgroup U of a polycyclic group G can be written as the intersection of all subgroups of finite index in G containing U. Further, each maximal subgroup of a polycyclic group G has finite index in G.
Since a polycyclic group G is finitely presented, there are only finitely many subgroups of a given index in G. In fact, the subgroup growth of a polycyclic group G is known to be polynomial as we recall in the following lemma.

9.5. Lemma: *Let G be a polycyclic group and let $f_G(x)$ be the number of subgroups of index x in G. If G has a polycyclic sequence of length n, then $f_G(x) \leq x^{2n}$.*

Proof. We use induction upwards the polycyclic sequence $\mathcal{G}$ of G. Thus let $G_2 = \langle \mathcal{G}_2 \rangle$. Then G_2 is a polycyclic group with a polycyclic sequence of length $n-1$. Hence, by induction, $f_{G_2}(x) \leq x^{2(n-1)}$ for each $x \in \mathbb{N}$. Let U be a subgroup of G of index x in G. Then $U \cap G_2$ is a subgroup of index y in G_2 for $y \mid x$. Thus we have at most $f_{G_2}(y)$ possible choices for $U \cap G_2$. Further, UG_2/G_2 is a subgroup of index x/y in G/G_2. If such a subgroup exists, then it is unique, since G/G_2 is cyclic. Let $UG_2 = \langle g_1^e, G_2 \rangle$. Then

$U = \langle g_1^e h, U \cap G_2 \rangle$ for some $h \in G_2$ and it is sufficient for defining U to determine the coset $h(U \cap G_2)$. Hence we have y possible choices for h. We obtain

$$f_G(x) \leq \sum_{y|x} y \cdot f_{G_2}(y) \leq \sum_{y|x} y \cdot y^{2(n-1)} = \sum_{y|x} y^{2n-1} \leq x \cdot x^{2n-1} = x^{2n}.$$

□

9.2.1 Computing (normal) subgroups of bounded index

Let G be a polycyclic group given by a constructive polycyclic sequence and let $x \in \mathbb{N}$. We describe an algorithm to compute the conjugacy classes of subgroups of index dividing x using induction on a linear abelian series of G. Let A be the last term in a linear abelian series and suppose the subgroups of G/A of index dividing x are known. The following lemma yields the inductive step. It is proved similarly to Lemma 8.5.

9.6. Lemma: *Let G be a polycyclic group with abelian normal subgroup A. Suppose that the subfactors $U_1/A, \ldots, U_l/A$ are representatives for the conjugacy classes of subgroups of index dividing x of G/A. Furthermore, for $1 \leq j \leq l$ let $H_j/A = N_{G/A}(U_j/A)$ and denote $y_j = [G : U_j]$.*
Then representatives for the conjugacy classes of subgroups of index dividing x of G can be obtained as representatives for the H_j-classes of subgroups K of index dividing x/y_j in U_j with $KA = U_j$ for $1 \leq i \leq j$.

Let $y \mid x$ and suppose that U is a subgroup of index y in G with $A \leq U$ and $H = N_G(U)$. By the above lemma we have to determine the H-classes of supplements of index dividing x/y in U. An algorithm for this purpose has been described in Section 8.3.1. Thus we obtain the following method to determine the conjugacy classes of subgroups of index dividing x in a polycyclic group G.

```
LowIndexSubgroups(G, x)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize C as the set of conjugacy classes {G^G}
    for i in {1, ..., l} do
        initialize K = ∅
        for each class C in C do
            let U be a representative of C with H = N_G(U) and y = [G : U]
            compute the H-classes of supplements K/G_{i+1} to G_i/G_{i+1} in U/G_{i+1} with y[U : K] | x
                (Section 8.3.1)
            add the preimages in G of the new classes to the list K
        end for
        add K to C
    end for
    return the set C of classes of subgroups of index dividing x
```

This method can be modified readily to determine normal subgroups only. For this purpose we need to restrict the supplement algorithm of Section 8.3.1 to compute normal supplements only. This, in turn, relies mainly on the calculation of normal complements as sketched in Remark 6.5.

References: An alternative approach to determine low index subgroups of polycyclic groups has been described by Lo in [43]. A comparison of Lo's algorithm and the method introduced here is included in [17]. It shows that the method introduced by Lo is faster, if the polycyclic sequence of the underlying group is short and it is less effective otherwise.

9.2.2 Determination of maximal subgroups

An infinite polycyclic group G may have infinitely many conjugacy classes of maximal subgroups. Hence we cannot determine all maximal subgroups of G explicitly. However, the following lemma yields a parameterization of the maximal subgroups of G that we can use to compute maximal subgroups of G.

9.7. Lemma: *Let G be a polycyclic group.*

a) A maximal subgroup of G has prime power index in G.

b) For a fixed prime p there are only finitely many maximal subgroups of p-power index.

Proof. Let $G = G_1 \rhd \ldots \rhd G_l \rhd G_{l+1} = 1$ be a normal series with abelian factors and let U be a maximal subgroup in G. Then there exists an index i with $G_{i+1} \leq U$ and $G_i \not\leq U$. Since U is maximal, we obtain $UG_i = G$. Further, $N = U \cap G_i$ is invariant under action of U. Since G_i/G_{i+1} is abelian, N is also invariant under G_i and hence N is normal in G. The maximality of U now yields that N/G_{i+1} is maximal G-normal in G_i/G_{i+1} and G_i/N is irreducible as G-module. In particular, G_i/N is an elementary abelian p-group for a prime p and $[G : U] = [G_i : N]$ is a prime-power proving a). Moreover, $N/G_{i+1} \geq (G_i/G_{i+1})^p$. Therefore, there are only finitely many possible choices for the intersection N of U with the given normal series of G. Further, for a fixed subgroup N there are at most finitely many maximal subgroups U having N as intersection with the normal series by Lemma 6.11 c), since U/N is a complement to G_i/N in G/N. Thus we obtain b). □

We describe a method to determine the maximal subgroups of p-power index in a polycyclic group G given by a constructive polycyclic sequence. We use induction on a linear abelian series for this purpose. Let A be the last non-trivial term in this series and suppose by induction that the maximal subgroups of p-power index in G/A are known. Clearly, their preimages in G yield the maximal subgroups U of p-power index in G with $A \leq U$. Thus it remains to determine those maximal subgroups U of p-power index in G with $A \not\leq U$. In this case $N = A \cap U$ is a proper maximal G-normal subgroup of A. Further, N has p-power index in A and thus $A^p \leq N$.
Vice versa, to construct the maximal subgroups U with p-power index in G and $A \not\leq U$, we consider the elementary abelian factor A/A^p. We determine all subgroups U of G that supplement A/A^p in such a form that the intersection of U with the factor is a maximal G-normal subgroup of A/A^p. Supplements of this type can be determined by a straightforward modification of the supplement algorithm as outlined in Section 8.3.1.

```
MaximalSubgroupClasses(G, p)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize C = ∅
    for i in {1, ..., l} do
        determine L/G_{i+1} = (G_i/G_{i+1})^p
        compute the G-classes of maximal supplements to G_i/L in G/L (Section 8.3.1)
        add the preimages in G of the new classes to the list C
    end for
    return the set C of classes of maximal subgroups of p-power index in G
```

9.8. *Remark:* Clearly, if G_i/G_{i+1} is an elementary abelian q-group for a prime $q \neq p$, then $(G_i/G_{i+1})^p = G_i/G_{i+1}$ and the factor does not yield any new maximal subgroups of the desired type. However, also if G_i/G_{i+1} is an elementary abelian p-group or a free abelian factor, it may not give rise to any maximal

subgroups of p-power order. If we use a linear abelian series as outlined in Section 9.8, then we can identify factors of this type *a priori* and thus we obtain an improvement for this method to determine maximal subgroups.

9.2.3 The p-Frattini subgroup

Let G be a polycyclic group. Then the intersection of all maximal subgroups in G is called the *Frattini subgroup* $\Phi(G)$ of G. The *p-Frattini subgroup* $\Phi_p(G)$ of G is the intersection of all maximal subgroups of p-power index. Both $\Phi(G)$ and $\Phi_p(G)$ are characteristic subgroups of G. They are related by

$$\Phi(G) = \bigcap_{p \text{ prime}} \Phi_p(G).$$

The next lemma shows that the p-Frattini subgroup can essentially be computed in a finite image of G.

9.9. Lemma: *Let G be a polycyclic group and $U_1, \ldots, U_m$ a conjugacy class of maximal subgroups of p-power index in G supplementing an elementary abelian normal p-subgroup A. Define $C = C_G(A)$ and $K = C'C^p$. Then $K \leq \cap_{i=1}^m U_i$ and K has finite index in G.*

Proof. Consider $U = U_1$ and without loss of generality assume that $A \cap U = 1$. Then $C = C_U(A) \times A$, since U is a complement to A. Thus we obtain $K = C'C^p \leq C_U(A) \leq U$, since A is an elementary abelian p-group. □

The maximal subgroups of p-power index can be determined as in Section 9.2.2 and their cores can be computed in a finite image of G by Lemma 9.9. Thus we can determine $\Phi_p(G)$ as the intersection of finitely many normal subgroups with finite index. In particular, we obtain that $\Phi_p(G)$ has finite index in G for each prime p, while $\Phi(G)$ may have infinite index in an infinite polycyclic group.

9.3 Fitting subgroup

The Fitting subgroup $Fit(G)$ of a polycyclic group G is the maximal nilpotent normal subgroup of G. This subgroup exists, since polycyclic groups satisfy the maximal condition on subgroups. The Fitting subgroup can also be described as the product of all nilpotent normal subgroups in G. We refer to [67], page 133ff, for further information.
For finite groups various characterizations for the Fitting subgroup are known. For example, the Fitting subgroup of a finite group can be described as the centralizer of a chief series. Here we show that the Fitting subgroup of a polycyclic group can also be characterized as centralizer of a certain type of series which can be considered as a generalization of the chief series in a finite group. We introduce an algorithm to compute a generalized chief series of a polycyclic group given by a constructive polycyclic sequence and hence we obtain a method to construct its Fitting subgroup.

9.3.1 A characterization of the Fitting subgroup

9.10. Theorem: *Let G be a polycyclic group and let $G = G_1 \rhd \ldots \rhd G_l \rhd G_{l+1} = 1$ a linear abelian series such that each free abelian factor is semisimple as a $\mathbb{Q}G$-module and each elementary abelian p-factor is semisimple as a $\mathbb{F}_pG$-module. Then*

$$Fit(G) = \bigcap_{i=1}^{l} C_G(G_i/G_{i+1}).$$

Proof. Let $C = \bigcap_{i=1}^{n} C_G(G_i/G_{i+1})$. Then the series $C \cap G_1, \ldots, C \cap G_{l+1}$ is a central series of C and hence we obtain that C is nilpotent. Clearly, $C \trianglelefteq G$ and thus $C \leq Fit(G)$.
It remains to show that $Fit(G) \leq C$. We prove that each nilpotent normal subgroup F of G centralizes the given normal series of G. First, let A be an arbitrary abelian normal subgroup of G. Then it is well-known that $[A, F] < A$: either, if $A \not\leq F$, then $[A, F] \leq A \cap F < A$, or, if $A \leq F$, then intersecting a central series of F with A yields an F-central series of A and hence $[A, F] < A$. Therefore, by induction we obtain that F centralizes a finite series through each finite abelian subgroup of G. This property generalizes to finite abelian factor groups of G and we use it throughout the following proof.
Let $A = G_i/G_{i+1}$ be a factor of the given series of G. Without loss of generality we can assume that $G_{i+1} = 1$ and A is a subgroup of G. By assumption, A is semisimple as KG-module for $K = \mathbb{F}_p$ or $K = \mathbb{Q}$ and thus it has a direct decomposition $A = A_1 \times \ldots \times A_r$ into subgroups $A_j \trianglelefteq G$ which are KG-irreducible. We consider $B = A_j$ for some j and show that B is centralized by F. If B is elementary abelian, then $[B, F] < B$ as observed above and $[B, F]$ is a KG-submodule of B. Since B is irreducible, we obtain that $[B, F] = 1$ and F centralizes B. It remains to consider the case that B is free abelian. Let p be a prime and consider the elementary abelian p-group B/B^p. As above, F centralizes a series in B/B^p. Hence, if d is the rank of B, then the d-fold commutator yields $[B,_d F] = [B, \ldots, B, F] \leq B^p$. As p is an arbitrary prime, we obtain $[B,_d F] = 1$ and F centralizes a subgroup series of length at most d in B. In particular, $C = C_B(F) > 1$. However, C is normal in G and thus, since B is irreducible as $\mathbb{Q}G$-module, C has finite index in B. But centralizers in free abelian groups are pure subgroups and thus we obtain $C = B$ as desired. □

9.11. *Remark:*

a) A group G centralizes a semisimple KG-module if and only if it centralizes each of the irreducible factors in a composition series of the module. Hence, without loss of generality, we could replace 'semisimple' by 'irreducible' in Theorem 9.10. For finite groups this reduces then to the well-known characterization of the Fitting-subgroup as centralizer of a chief series.

b) Each polycyclic group G has a non-trivial free abelian normal subgroup. If A is a subgroup of this type of minimal rank, then A is irreducible as a $\mathbb{Q}G$-module. Thus, by induction, we obtain that a series of the type required by Theorem 9.10 exists in each polycyclic group.

9.3.2 Determining the Fitting subgroup

Let G be a polycyclic group given by a constructive polycyclic sequence. We determine a generalized chief series of G as required in Theorem 9.10 by first computing an arbitrary linear abelian series of G and then refining the factors of this series by their radical series as G-modules. For modules over finite fields such methods are part of the 'Meataxe', see [48] or [34], and for the case of a free abelian module we apply the algorithm outlined in Section 5.4.3. Once a series with semisimple factors is given, we determine its centralizer by successively centralizing each factor in the series as described in Section 7.5.

```
FittingSubgroup(G)
    determine a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} = 1 (Section 8.2)
    initialize F = G
    for i in {1,...,l} do
        compute the G-module radical series of G_i/G_{i+1} (Meataxe or Section 5.4.3)
        determine the centralizer H in F of the resulting series (Section 7.5)
        reset F = H
    end for
    return the Fitting subgroup F
```

9.4 Centre and FC-centre

Let G be a group. Then $Z(G) = \{g \in G \mid gh = hg \text{ for all } h \in G\}$ is the *centre* of G. Further, an element $g \in G$ is an *FC-element* in G if the conjugacy class g^G is finite. It is straightforward to observe that the set of all FC-elements in G forms a characteristic subgroup: the *FC-centre* $FC(G)$. Clearly, we have $Z(G) \leq FC(G)$. As an application of the Fitting subgroup algorithm we present methods to determine the centre and the FC-centre of a polycyclic group G given by a constructive polycyclic sequence.

9.4.1 The determination of the centre

As a first observation to determine $Z(G)$ we note that we can describe $Z(G)$ as the centralizer in $Z(Fit(G))$ under the action of G:

$$Z(G) = C_{Z(Fit(G))}(G).$$

Thus we first determine $Fit(G)$ as outlined in Section 9.3.2. Since $Fit(G)$ is a nilpotent group, $Fit(G)$ has a central series of finite length; e.g. the lower central series of $Fit(G)$ can be constructed readily. The following lemma and induction over a central series then yield $Z(Fit(G))$.

9.12. Lemma: *Let H be a nilpotent group generated by $h_1, \ldots, h_r$. Let $A \leq Z(H)$ and $C/A = Z(H/A)$. Then $\psi_i : C/A \to A : cA \mapsto [c, h_i]$ is a homomorphism of abelian groups and $Z(H) = \bigcap_{i=1}^{r} ker(\psi_i)$.*

Proof. Let $h = h_i$ and $\psi = \psi_i$. Since C/A is the centre of H/A, we obtain that $[c, h] \in A$. Further, since $[ca, h] = [c, h]^a[a, h] = [c, h]$ for $c \in C$ and $a \in A$, the map ψ is well-defined. We observe that ψ is linear by $[c_1c_2, h] = [c_1, h]^{c_2}[c_2, h] = [c_1, h][c_2, h]$, since A is central. Finally, $Z(H) = C_C(H) = \bigcap_{i=1}^{r} C_C(h_i)$ and $C_C(h_i) = K_i$. □

In applying Lemma 9.12 we have to determine the kernels of homomorphisms of abelian groups. This problem has been considered in Section 3.5.4 and its solution can be reduced to solving a system of linear equations.
Once $Z(Fit(G))$ is determined, it remains to calculate its fixed points under the action of G. Since $Z(Fit(G))$ is abelian, we can consider it as a $\mathbb{Z}G$-module and obtain the fixed points submodule using linear algebra: we successively determine the fixed points under each generator g of G and this, in turn, amounts to a nullspace calculation for the action of $g - 1$.

Centre(G)
  determine $H = Fit(G)$ (Section 9.3)
  calculate $A = Z(H)$ (Lemma 9.12)
  compute $C = C_A(G)$ (using fixed points)
  return the centre C

9.4.2 The determination of the FC-centre

The structure of the FC-centre of a group has been investigated in various places and a summary of its properties can be found in [64], pages 121ff. We recall the structure theory of FC-centers which we use in our algorithm in the following lemma.

9.13. Lemma: *Let G be a polycyclic group.*

a) (Neumann) *The set of elements of finite order in $FC(G)$ forms a characteristic subgroup $T(FC(G))$ of $FC(G)$ with $FC(G)' \leq T(FC(G))$.*

b) $T(FC(G)) = TN(G)$ *and* $FC(G)/T(FC(G)) = FC(G/TN(G))$ *where* $TN(G)$ *is the maximal finite normal subgroup of* G*.*

If G is a polycyclic group, then we can determine $TN(G)$ as described in Section 9.1. By Lemma 9.13 b) we can pass to $G/TN(G)$ and thus we may assume that $TN(G)$ is trivial. In this case $FC(G)$ is a free abelian group by Lemma 9.13 a) and we determine it similarly to the approach in Section 9.4.1 using the following theorem.

9.14. Theorem: *Let* G *be a polycyclic group with* $TN(G) = 1$*. Then* $Z(Fit(G))$ *is free abelian, i.e.* $Z(Fit(G)) \cong \mathbb{Z}^d$*. Let* $\gamma : G \to GL(d, \mathbb{Z})$ *be the corresponding operation of* G *and let* $N \lhd G$ *such that* $N^\gamma = K_p(G^\gamma)$ *for an odd prime* p*. Then*

$$FC(G) = C_{Z(Fit(G))}(N).$$

Proof. Let $H = C_{Z(Fit(G))}(N)$ to shorten notation. We want to show $FC(G) = H$. Since $TN(G) = 1$, we obtain that H is free abelian by construction and $FC(G)$ is free abelian by Lemma 9.13. As H is centralized by N and N has finite index in G, the group G acts as a finite group on H. Hence each element in the abelian group H has a G-conjugacy class of finite length and thus $H \leq FC(G)$.
It remains to show that $FC(G) \leq H$. Obviously, since $FC(G)$ is abelian, we have that $FC(G) \leq Fit(G)$. Further, since each element in $FC(G)$ has a G-conjugacy class of finite length, the group G acts as a finite group on $FC(G)$. By Theorem 9.10, $Fit(G)$ centralizes a series in $FC(G)$ and thus acts as a group of unitriangular matrices on $FC(G)$. Since the group of unitriangular matrices is torsion-free, we obtain that that $Fit(G)$ acts trivially on $FC(G)$, and hence $FC(G) \leq Z(Fit(G))$.
It remains to show that $FC(G)$ is centralized by N. Recall that N acts as p-congruence subgroup on the free abelian group $Z(Fit(G))$. If $g \in Z(Fit(G))$ is an element which is not centralized by N, then the results of Section 7.3 show that the orbit of g under N has infinite length. Thus $g \notin FC(G)$. Hence $FC(G)$ is centralized by N and therefore we obtain $FC(G) \leq C_{Z(Fit(G))}(N) = H$ as desired. □

FCCentre(G)
  determine $T = TN(G)$ (Section 9.1)
  compute $H/T = Fit(G/T)$ (Section 9.3)
  calculate $A/T = Z(H/T)$ (Lemma 9.12)
  identify $A/T = \mathbb{Z}^d$ and consider the action $\gamma : G/T \to GL(d, \mathbb{Z})$
  compute $K_p((G/T)^\gamma)$ for an odd prime p (Chapter 5)
  determine the preimage N/T of $K_p((G/T)^\gamma)$ (Section 3.5)
  calculate $C/T = C_{A/T}(N/T)$ (using fixed points)
  return the FC-centre C

9.5 Exhibiting the nilpotent-by-abelian-by-finite structure

A well-known theorem of Mal'cev [49] asserts that a polycyclic group is (nilpotent-by-abelian)-by-finite; that is, there exists a subnormal series $1 \leq N \leq H \leq G$ with N nilpotent, H/N abelian and G/H finite. We introduce an algorithm to exhibit this structure in a polycyclic group given by a constructive polycyclic sequence. First, we recall in the next lemma that there also exists a normal series of this type.

9.15. Lemma: *Let* G *be polycyclic. Then* G *is nilpotent-by-(abelian-by-finite).*

Proof. Each polycyclic group G embeds into $GL(d, \mathbb{Z})$ for some d by [1]. Then $V = \mathbb{Q}^d$ can be considered as natural G-module. Let $V = V_1 > V_2 > \ldots > V_r > V_{r+1} = 0$ be the radical series of V and consider

the natural action homomorphism $G \to G_{V_1/V_2} \times \ldots \times G_{V_r/V_{r+1}}$. The image of this homomorphism is abelian-by-finite, since each group $G_{V_i/V_{i+1}}$ is abelian-by-finite by Corollary 5.8. The kernel of the action homomorphism is a unipotent subgroup of $GL(d, \mathbb{Z})$ and thus nilpotent. □

The next theorem can be considered as a more constructive version of Lemma 9.15. It shows that the structure described by Lemma 9.15 can be exhibited in a polycyclic group with the methods introduced in the previous sections.

9.16. Theorem:

a) Let G be a finitely generated nilpotent group which is abelian-by-finite. Then $[G : Z(G)] < \infty$.

b) Let G be polycyclic. Then with $N = Fit(G)$ and $H/N = Z(Fit(G/N))$ we obtain a normal series $1 \leq N \leq H \leq G$ such that N is nilpotent, H/N is abelian and G/H is finite.

Proof. a) Let $A \cong \mathbb{Z}^d$ be a free abelian normal subgroup of finite index in G. By Theorem 9.10, G centralizes a series in A. Thus G acts as a group of unitriangular matrices on A. However, the group of unitriangular matrices in $GL(d, \mathbb{Z})$ is torsion-free. Since G/A is finite, we obtain that G acts trivially on A and A is central.
b) Clearly, N is nilpotent and H/N is abelian. It remains to show that G/H is finite. By Lemma 9.15, $F = G/N$ is abelian-by-finite. Thus there exists an abelian normal subgroup $A \lhd F$ with finite index. Hence $A \leq Fit(F)$ and thus $Fit(F)$ has finite index in F. By a), $Z(Fit(F))$ has finite index in $Fit(F)$. Hence $Z(Fit(F))$ has finite index in F and H has finite index in G as desired. □

Theorem 9.16 b) can readily be translated into an algorithm as follows.

```
NilpotentByAbelianByFiniteStructure(G)
  determine N = Fit(G) (Section 9.3)
  compute F/N = Fit(G/N) (Section 9.3)
  determine H/N = Z(F/N) (Lemma 9.12)
  return H and N
```

9.6 Nilpotent almost supplements

We recall another well-known structure theorem for polycyclic groups in the following theorem. It shows that a polycyclic group can be build up in a rather simple way from nilpotent groups and finite groups. For a proof see [68].

9.17. Theorem: *Let G be a polycyclic group. Then there exist nilpotent subgroups N and U in G with $N \trianglelefteq G$ and $[G : NU]$ finite; that is, U is a* nilpotent almost supplement *to N in G.*

As outlined in Sections 9.3.2 and 9.5 we can determine $N = Fit(G)$ and $H/N = Z(Fit(G/N))$. We observed in Theorem 9.16 that H has finite index in G. Thus to exhibit Theorem 9.17 it remains to solve the following problem: given a nilpotent normal subgroup N of G and an intermediate subgroup H with H/N nilpotent and $[G : H]$ finite, construct a nilpotent subgroup U of H with $[H : UN]$ finite.
We use induction down a linear abelian series of N which is H-invariant and has central factors. For the inductive step let A be the last non-trivial term in this series and suppose we have constructed a subgroup U such that U/A is nilpotent and $[H : UN] < \infty$. First we determine the U-hypercentre of A; that is, we compute an ascending series with U-central factors in A. This series terminates at a subgroup L of A with $C_{A/L}(U) = 1$. Now we distinguish two cases.

a) A is elementary abelian. Then by Theorem 6.13 a), there exists a complement K/L to A/L in U/L. This complement can be determined using Section 6.3. Since K has finite index in U, we obtain that KN has finite index in H.

b) A is free abelian. Then A/L is also free abelian, since fixed point subspaces in lattices are pure. By Theorems 6.13 b) and 6.15 b) there exists an almost complement K/L to A/L in U. This can be determined by the method in Section 6.5. We obtain that KA has finite index in U and hence $[H : KN] = [H : UN][UN : KN]$ is finite.

Thus in both cases we obtain a subgroup K with $A \cap K = L$ and KN of finite index in H. Since L has a series with K-central factors and $K/L \cong KA/A \leq U/A$ is nilpotent, we obtain that K is nilpotent.

NilpotentAlmostSupplement(G, N, H)
  assume that N and H/N are nilpotent and $[G : H] < \infty$ (compare Section 9.5)
  determine an H-invariant linear abelian series $N = N_1 \rhd \ldots \rhd N_l \rhd N_{l+1}$ with central factors
  initialize $U = H$
  for i in $\{1, \ldots, l\}$ do
    compute the U-hypercentre L/N_{i+1} in N_i/N_{i+1} (using fixed points)
    if N_i/N_{i+1} is finite, then determine a complement K/L to N_i/L in U/L (Section 6.3)
    otherwise, determine an almost complement K/L to N_i/L in U/L (Section 6.5)
    reset $U = K$
  end for
  return the nilpotent almost supplement U

9.7 Poly-(free-abelian) normal subgroups of finite index

As observed in [67], page 148, each polycyclic group has a normal subgroup N of finite index in G such that N is poly-$\mathbb{Z}$; that is, there exists a subnormal series with infinite cyclic factors in N. Using such a subgroup we can derive a polycyclic series of G in which all finite factors are at the top and all infinite factors are at the bottom of the series. Note that the converse structure cannot always be achieved.
We describe an algorithm to compute such a subgroup N in a polycyclic group G given by a constructive polycyclic sequence. In fact, we determine a characteristic series $G > N = N_1 \rhd \ldots \rhd N_l \rhd 1$ of G such that $[G : N]$ is finite and each factor N_i/N_{i+1} is free abelian. Clearly, N is poly-$\mathbb{Z}$ in this case.
We start with a linear abelian series of G as described in Section 8.2.1. In the following lemma we introduce a method to move a free abelian section downwards across an elementary abelian section. By an iterated application of this lemma we can successively move down all the free abelian factors of the given linear abelian series.

9.18. Lemma: *Let A be a characteristic elementary abelian p-subgroup of a polycyclic group G such that G/A is free abelian. Define $C = C_G(A)$ and let $K = C^q = \langle g^q \mid g \in C \rangle$, where $q = p$, if $p > 2$, and $q = 4$ otherwise.*

a) $K = \{g^q \mid g \in C\}$.

b) K is a characteristic free abelian subgroup of G.

c) K has finite index in G; more precisely, $[G : K] = [G : C] \cdot p^t \cdot q^s$, where t and s are the ranks of A and G/A, respectively.

Proof. a) Let $g, h \in C$. Since $[h, g] \in A$ is centralized by C, we have

$$(gh)^q = g^q \cdot h^q \cdot [h, g]^{q(q-1)/2} = g^q \cdot h^q.$$

Thus the q-th powers in C form a subgroup and a) is proved.
b) Let $g \in A \cap K$. Then $g = h^q$ and h is an element of finite order. Since C/A is free abelian, we obtain $h \in A$. But A is elementary abelian and thus $g = 1$. Therefore $K \cap A = 1$ and $K \cong KA/A \leq C/A$ is free abelian. K is characteristic in G, since C is characteristic.
c) We use $[G : K] = [G : C][C : K]$ and $[C : K] = [C : KA][A : A \cap K] = q^s \cdot p^t$. □

In Lemma 9.18 c) we observe that the index of K in G might be rather large. However, every normal free abelian subgroup L of G intersects trivially with A and centralizes A. Thus the index of L in G is at least $[G : C] \cdot p^t$. Hence the subgroup K as determined in Lemma 9.18 has index at most a factor of q^s times the optimal index. Vice versa, we can observe that this index is optimal in certain groups G. Let $G = \langle g, h, a \rangle$ such that $A = \langle a \rangle$ is a central subgroup of G of order 3 and G/A is free abelian of rank 2 with and $[g, h] = a$. Then G/K is a non-abelian group of order 27 and thus K is a maximal free abelian normal subgroup of G. A direct computation shows that there exists no normal free abelian subgroup L in G with $[G : L] < 27$. Hence K has optimal index in this case.
Lemma 9.18 yields a method to compute a characteristic free abelian subgroup of an (elementary abelian)-by-(free abelian) polycyclic group G. We first compute the centralizer $C = C_G(A)$ of the finite elementary abelian group A. Then, if $C = \langle c_1, \ldots, c_s, A \rangle$, we obtain K as $\langle c_1^q, \ldots, c_s^q \rangle$. An iterated application of Lemma 9.18 can be used to determine a poly-(free abelian) normal subgroup N of a polycyclic group G as we outline in the following.

```
PolyFreeNormalSubgroup(G)
    compute a linear abelian series G = G_1 ▷ ... ▷ G_l ▷ G_{l+1} (Section 8.2)
    let i such that G_i/G_{i+1} is free abelian and G_j/G_{j+1} is elementary abelian for all j > i
    initialize N = G_i
    for j in {i+1, ..., l} do
        determine a free abelian normal subgroup H/G_{j+1} in N/G_{j+1} (Lemma 9.18)
        reset N = H
    end for
    determine recursively K/N as PolyFreeNormalSubgroup(G/N)
    return the poly-ℤ subgroup K
```

9.8 Special abelian normal series

Let G be a polycyclic group given by a constructive polycyclic sequence. Then using the methods of Section 8.2 we can determine a short normal series with abelian factors for G. While it is often useful for algorithmic purposes to have a short series, there are also a number of algorithms where we would like to have control over other properties of G; for example, the centralizers of the factors of the series are of interest in many cases. Here we introduce abelian normal series which allows us read off properties of the group G.
The upper nilpotent series of G is defined by $\mathcal{F}_1(G) = 1$ and $\mathcal{F}_{i-1}(G)/\mathcal{F}_i(G) = Fit(G/\mathcal{F}_i(G))$. Clearly, the upper nilpotent series is a characteristic series of G. If G is polycyclic, then its Fitting subgroup is non-trivial and G satisfies the maximal condition on subgroups. Thus the upper nilpotent series of G terminates at G, say $\mathcal{F}_{l+1}(G) = G$. We can construct the upper nilpotent series of a given polycyclic group G by an iterated application of the Fitting subgroup algorithm of Section 9.3. The nilpotent factors of this series are denoted as $F_i = \mathcal{F}_{i-1}(G)/\mathcal{F}_i(G)$.

The two most natural abelian normal series in a nilpotent group F are lower and upper central series. If F has a constructive polycyclic sequence, then the lower central series $F = \lambda_1(F) \rhd \ldots \rhd \lambda_c(F) \rhd \lambda_{c+1}(F) = 1$ can be determined as in Section 8.1. The upper central series $1 = \gamma_1(F) \lhd \ldots \lhd \gamma_c(F) \lhd \gamma_{c+1}(F) = F$ defined by $\gamma_{i-1}(F)/\gamma_i(F) = Z(F/\gamma_i(F))$ can be constructed using an iterated application of the centre algorithm for nilpotent groups described in Lemma 9.12.
By refining each nilpotent factor F_i of G by its upper or lower central series we obtain a characteristic abelian normal series of G. Clearly, if desired, then we can refine the abelian factors to elementary or free abelian factors as outlined in Section 8.2. In the following two sections we consider the properties of the resulting series of G in more detail.

9.8.1 The refinement by the lower central series

Consider the factor $F_i = \mathcal{F}_{i-1}(G)/\mathcal{F}_i(G)$ of the upper nilpotent series in the polycyclic group G. We define subgroup $F_{i,j}$ of G by $F_{i,j}/\mathcal{F}_i(G) = \lambda_j(F_i)$ and use these subgroups to refine the upper nilpotent series by the lower central series. We call the series of subgroups $F_{i,j}$ in G the *lower refined series*.
Recall that the Frattini subgroup of a finite abelian group can be described as its smallest subgroup whose factor group is a direct product of elementary abelian groups. Thus the Frattini subgroup of a finite abelian group can be read off from a constructive polycyclic sequence of the group. In the following lemma we recall the determination of the Frattini subgroup of a finitely generated nilpotent group.

9.19. Lemma: *Let F be a finitely generated nilpotent group and denote $F/\lambda_2(F) = A \times T$ where A is free abelian and T is finite. Then $\lambda_2(F) \leq \Phi(F)$ and $\Phi(F)/\lambda_2(F) = \Phi(T)$.*

Thus we can refine the lower refined series of G one step further by inserting the groups $F_i^*/\mathcal{F}_i(G) = \Phi(F_i)$. Then we obtain that the *head factors* F_i/F_i^* form direct products of elementary or free abelian groups. Each non-head factor of the series is called a *tail factor*.

9.20. Lemma:

a) $G/\mathcal{F}_{i-1}(G)$ acts faithfully on the head factor F_i/F_i^*.

b) F_i centralizes each tail factor $F_i^*/F_{i,2}$ and $F_{i,j}/F_{i,j+1}$.

Proof. a) Without loss of generality we can suppose that $\mathcal{F}_i(G) = 1$ and thus $F_i = Fit(G)$. If there is a non-trivial element $g\mathcal{F}_{i-1}(G) \in G/\mathcal{F}_{i-1}(G)$ which centralizes F_i/F_i^*, then g centralizes a series through $F_i/F_{i,2}$ by definition of F_i^*. Thus g also centralizes the other factors of the lower central series of F, since they can be obtained as factors of tensor powers of $F_i/F_{i,2}$, see [67], page 131. Thus g centralizes a series in F_i and hence $\langle F_i, g\rangle$ is nilpotent.
b) This is trivial, since the factors refine a central series of F_i. □

Lemma 9.20 a) also yields an embedding of $G/\mathcal{F}_{i-1}(G)$ into a direct product of linear groups $GL(d, R)$ where R is a field of prime order or $R = \mathbb{Z}$.

9.21. Lemma: *Let M be a maximal subgroup of G with $F_{i+1} \leq M$ and $F_i \not\leq M$. Then $F_i^* \leq M$.*

Proof. Without loss of generality we can assume that $F_{i+1} = 1$. Thus $F_i \trianglelefteq G$ and hence $F_i^* = \Phi(F_i) \leq \Phi(G)$. Hence $F_i^* \leq M$. □

This yields an improvement of the maximal subgroup computation as outlined in Section 9.2.2, since we can identify a large number of factors in a series of G which have to be covered by each maximal subgroup of G.

9.8.2 The refinement by the upper central series

Consider the factor $F_i = \mathcal{F}_{i-1}(G)/\mathcal{F}_i(G)$ of the upper nilpotent series in the polycyclic group G. We define subgroups $G_{i,j}$ of G by $G_{i,j}/\mathcal{F}_i(G) = \gamma_j(F_i)$ and we call the series of subgroups $G_{i,j}$ in G the *upper refined series*.
The determination of the upper central series in a nilpotent group by an iterated application of the centre algorithm in Section 9.4 is more time-consuming than the construction of the lower central series; in fact, the determination of the lower central series is used in the centre algorithm of Section 9.4. Hence the series considered in this section is usually less important than the series constructed in Section 9.8.1. One advantage of the upper refined series over the lower refined series is that the resulting series is usually shorter. In the following lemma we recall this for an important special case.

9.22. Lemma: *Let F be a finitely generated nilpotent group.*

a) F is torsion-free if and only if $Z(F)$ is torsion-free.

b) If F is torsion-free, then each factor of the upper central series of F is torsion-free.

Proof. a) Clearly, if F is torsion-free, then $Z(F)$ is torsion-free. Vice versa, if F is not torsion-free, then $T(F)$ is a non-trivial finite subgroup of F. Since $T(F)$ is nilpotent, $Z(T(F))$ is a non-trivial finite subgroup of F. Further, since F is nilpotent, we obtain that $Z(T(F))$ contains a non-trivial fixed point under F by [67], 5.2.1. This fixed point is a non-trivial element of finite order in $Z(F)$.
b) By induction, it is sufficient to show that if F is torsion-free, then $F/Z(F)$ is torsion-free. Let $gZ(F)$ be a central element of finite order e in $F/Z(F)$. Then for $h \in F$ we have $g^h = gz_h$ for $z_h \in Z(F)$. Further, $g^e \in Z(F)$ and hence $g^e = (g^e)^h = (g^h)^e = (gz_h)^e = g^e z_h^e$. Thus z_h has finite order e. Since $Z(F)$ is torsion-free, we obtain $z_h = 1$ and thus g is central in F. □

9.8.3 A summary of applications

The properties of the lower refined series can be used to improve the centralizer algorithm of Section 8.5, the normalizer algorithm of Section 8.6, the intersection algorithm of Section 8.4 and the determination of maximal subgroups of Section 9.2.2.
The upper refined series can be considered as the extension of the upper central series for nilpotent groups to polycyclic groups. Hence it may have an interest in its own right, but it also might be useful to increase the effectiveness of certain algorithms. For example, the computation of all finite subgroups will be improved using the upper refined series, since the factors of the used linear abelian series which yield a large number of new groups are the central factors at the end of the series.
Further, the lower or upper refined series exhibit a large part of the structure theory for polycyclic groups as introduced in Sections 9.3, 9.4 and 9.5. Vice versa, the determination of the lower or upper refined series uses the algorithms introduced above; in particular, it employs an iterated application of the Fitting subgroup method of Section 9.3. Thus its initial determination will cost additional time, but it improves a variety of algorithms which eventually will reduce the time used in computations with the given group.

Further references and comments

Some of the problems considered in this chapter are only interesting for infinite polycyclic groups; for example, the determination of the FC-centre is trivial in finite polycyclic groups. But there are also problems which have been considered for finite polycyclic groups.
For finite polycyclic groups the computation of the conjugacy classes of finite subgroups is just the determination of all subgroup classes. This problem is well-investigated and we refer to Hulpke [37] for further details.

The Fitting subgroup in a finite polycyclic group G can also be computed as the direct product of the cores of the Sylow subgroups of G. This is an effective methods for this purpose, but it does not extend to the infinite case. Thus we needed to find an independent approach for the infinite group case.
The most recent methods to determine the centre or the maximal subgroups of a finite polycyclic group are based on the so-called 'special polycyclic sequences' which have been introduced by Leedham-Green. These methods do not extend to infinite groups. But the approach introduced in Section 9.8 can be considered as an approach towards a dual development for infinite groups. We refer to [19] for an account on special polycyclic sequences.

Chapter 10

Examples and applications

In the previous chapters we introduced a number of algorithms for polycyclic groups. Here we describe some applications for these algorithms in computations with infinite polycyclic groups.
First, we consider the application of our methods to computations with crystallographic and almost crystallographic groups. The groups of these types are infinite groups which are often polycyclic and hence they form a class of interesting examples for our methods. Then we give an example for an investigation of a group defined by a polycyclic presentation. This shows that polycyclic presentations provide a useful tool in the algorithmic theory of polycyclic groups. Finally, we outline an example application of our methods to integral polycyclic matrix groups.
We also use the introduced examples to report on the performance of some of our methods. The computations described below use implementations of our methods in the language provided by the computer algebra system GAP [74]. A part of these implementations is available in the Polycyclic package [23]. The computations have been performed on a PC with 64 MB Ram and a Celeron 500 Mhz processor running under Linux.

10.1 Applications to crystallographic groups

Crystallographic groups arise in the study of symmetries of crystals. An affine crystallographic group G can be defined as subgroup of the group of all Euclidean motions of a d-dimensional space such that the subgroup T of all pure translations is a discrete normal subgroup of finite index in G. If the free abelian group T has maximal rank d, then G is also called a space group. The finite factor group G/T is often denoted the point group of G.
Hence each crystallographic group is (free abelian)-by-finite and it is polycyclic if and only if its point group is polycyclic. In particular, all 2- and 3-dimensional crystallographic groups are polycyclic and thus the methods of this book apply to these groups. For further information on crystallographic groups we refer to [6] and we also note that there is a GAP share package [21] which can be used for certain computations with crystallographic groups.

By definition, a space group is an extension of $\mathbb{Z}^d$ by a finite subgroup of $GL(d,\mathbb{Z})$. We use this feature to describe the example groups which we use for the computations in this section. We consider the groups G and H of types $(4,4,1)$ and $(6,7,1)$ from the GAP catalog of irreducible maximal finite subgroups of $GL(d,\mathbb{Z})$. Then $G \leq GL(4,\mathbb{Z})$ is a group of order 384 and $H \leq GL(6,\mathbb{Z})$ has order 4608. We define G_1 and H_1 as the split extensions of the natural module by G and H, respectively, and we let G_2 and H_2 be two arbitrary non-split extensions of this type. We determine polycyclic sequences for all of the

considered groups and we obtain sequences with the relative orders

$$(2, 2, 3, 2, 2, 2, 2, 2, \infty, \infty, \infty, \infty)$$

for G_1 and G_2. For H_1 and H_2 we obtain sequences having the relative orders

$$(2, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, \infty, \infty, \infty, \infty, \infty, \infty).$$

We consider the computation of finite subgroups, subgroups of finite index and central extensions of these groups and we provide runtimes in seconds and results of such computations in the following paragraphs and tables. We also note here that a report with similar computations is included in [17].

In the following table we outline runtimes for the determination of various finite subgroups of the groups under consideration. These subgroups have been obtained using the methods of Section 9.1. In all cases we obtain that the set of torsion elements does not form a subgroup.

	$T(G)$		$TN(G)$		Classes of finite subgroups	
	is a subgroup	time	order	time	number	time
G_1	no	0.2	1	0.3	686	37.8
G_2	no	0.2	1	0.3	146	33.0
H_1	no	0.3	1	0.8	27508	80:46.06
H_2	no	0.3	1	0.8	2978	38:12.17

Next, we consider the computation of conjugacy classes of maximal subgroups of p-power index in the following table. We list the results for the split extensions G_1 and H_1 only, since the remaining two groups show a very similar behavior. This computation is an application of the method of Section 9.2.2.

	Classes of maximal subgroups of p-power index					
	$p = 2$		$p = 3$		$p = 5$	
	number	time	number	time	number	time
G_1	8	0.6	2	0.3	1	0.2
H_1	10	1.3	2	0.9	1	0.8

Finally, we consider the computation of the second cohomology group of G_1 and H_1 with their trivial module over $\mathbb{Z}$ as described in Section 6.3. For G_1 this computation takes 7.0 seconds, while it needs 57.3 seconds for H_1. In both cases we obtain a finite second cohomology group with abelian invariants $(2, 2, 2)$.

In summary, the runtimes obtained in this section show that the considered methods are practical for such applications. The determination of torsion subgroups, normal torsion subgroups, maximal subgroups and extensions can easily be extended to space groups of higher dimensions, while the computation of finite subgroup classes is limited by the number of subgroups arising.

- Crystallographic groups are often defined as affine rational matrix groups. Using their structure as (free abelian)-by-finite groups it is straightforward to determine constructive polycyclic sequences for such groups.
- The determination of extensions and torsion subgroups of crystallographic groups is also of interest in the classification of almost crystallographic and almost Bieberbach groups. Such groups will be considered in Section 10.2.

10.2 Computations with almost crystallographic groups

Almost crystallographic groups are finitely generated nilpotent-by-finite groups whose normal torsion subgroup is trivial. An almost crystallographic group is called almost Bieberbach group if it is almost crystallographic and torsion-free. By definition, the Fitting subgroup $Fit(G)$ of an almost crystallographic group G has finite index and its factor group $G/Fit(G)$ is called the holonomy group of G. Thus an almost crystallographic group is polycyclic if and only if its holonomy group is polycyclic.
Dekimpe introduced in [12] a library of almost crystallographic groups of Hirsch lengths 3 and 4. In particular, all almost Bieberbach groups of such Hirsch lengths are classified in this catalog of groups. All the groups contained in that library are polycyclic. Further, this catalog of groups is available as GAP package Aclib [13]. Each group in the Aclib catalog can be obtained in two different representations: as rational matrix group in dimension 4 or 5 or as polycyclically presented group.

Here we want to use the GAP library of almost crystallographic groups to investigate the performance of the element stabilizer algorithm introduced in Chapter 7. For this purpose we use the rational matrix group representations of the almost crystallographic groups and we consider the determination of orbits and stabilizers of vectors in their natural module. The considered matrix groups are unipotent-by-finite and hence we can apply the idea of Section 5.5.2 to conjugate them such that the unipotent normal subgroup is an integral matrix group. In this setting we can then apply the method of Chapter 7.
We consider the almost crystallographic groups of Hirsch length 4 in their matrix representation in $GL(5,\mathbb{Q})$. The Aclib classification contains 95 different types of such groups. The runtimes to determine the stabilizer of a random vector in $\mathbb{Z}^5$ under a group from this list are ranging between 0.03 seconds and 1.1 seconds. This shows that the stabilizer method for such groups is practical. In the following table we consider the groups G_1 and G_2 of types $(4,41)$ and $(4,82)$ from this list in more detail. These groups have polycyclic sequences with relative orders $(2,2,2,\infty,\infty,\infty,\infty)$ and $(2,6,\infty,\infty,\infty,\infty)$, respectively. The following table contains runtimes for some orbit stabilizer computations of elements in $\mathbb{Z}^5$ and it includes the relative orders for polycyclic sequences of the obtained stabilizers.

	$(4,-7,2,-1,3)$		$(1,0,0,0,0)$		$(0,0,1,0,0)$	
	stabilizer	time	stabilizer	time	stabilizer	time
G_1	(∞)	0.5	$(2,\infty)$	0.3	$(2,\infty,\infty,\infty)$	0.1
G_2	(∞)	0.7	(∞)	1.1	$(2,\infty,\infty,\infty)$	0.2

10.3 Investigations of polycyclically presented groups

In this section we investigate a group G defined by a polycyclic presentation whose underlying polycyclic sequence contains 24 generators and has the relative orders

$$(\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,5,4,\infty,5,5,4,\infty,6,5,5,4,\infty,10,6).$$

The group is available as 'NqExamples[1]' in the Polycyclic share package of GAP. Its polycyclic presentation has been determined by Nickel using the nilpotent quotient algorithm [54]. We omit the explicit description of the polycyclic presentation here, since it contains a large number of non-trivial powers and conjugates and thus would be place-consuming. However, this large number of non-trivial relations in the presentations indicates that the structure of G is likely to be interesting.

The determination of the lower central series of G using the method of Section 8.1 needs 2.8 seconds. We obtain a series of length 10 for G and we can observe that the lower central series of G is refined by the polycyclic series corresponding to the underlying polycyclic presentation. This is due to the determination of this polycyclic presentation by the nilpotent quotient method in [54].
Further, the torsion subgroup $T(G)$ can be computed in 1.3 seconds applying the approach of Section 9.1. Once generators for $T(G)$ are known, it is easy to check that $T(G)$ is abelian and has the abelian invariants $(5, 5, 10, 30, 300)$. The centre $Z(G)$ can be determined in 0.1 seconds with the algorithm of Section 9.4 and it has the abelian invariants $(10, 10, 30, 0)$. Hence it intersects in a group of order 3000 with $T(G)$ and this can be verified readily using the method of Section 8.4. The computation of the upper central series of G by an iteration of the method of Section 9.4 needs 2.5 seconds.
We investigate the factor group $H = G/T(G)$ further. A polycyclic presentation for this factor can be computed in 0.9 seconds and we obtain a presentation with 16 generators for H. Clearly, this factor is torsion-free which can also readily be verified with the method of Section 9.1. The determination of the lower central series of H takes 0.7 seconds and it shows that H has class 10. The upper central series of H can now be obtained in 0.7 seconds. The abelian invariants of the lower central series are

$$(0, 0), (0), (0), (0), (0, 0), (0), (0, 0), (2, 0), (2, 0), (0)$$

and for the upper central series we obtain

$$(0, 0), (0), (0), (0), (0, 0), (0), (0, 0), (0), (0), (0).$$

Using the upper central series we can compute a new polycyclic sequence of length 13 for H. The calculation of the corresponding polycyclic presentation for H needs 2.0 seconds. We obtain a polycyclic presentation of H without non-trivial power relations and the underlying polycyclic sequence is shortest possible for H. This presentation can be useful for further computations with H; for example, it is trivial to read off from this new polycyclic presentation that H is torsion-free.

10.4 Polycyclic sequences for integral matrix groups

In Chapter 5 we introduced an algorithm to determine a constructive polycyclic sequence for a rational matrix group. The described method relies further on the algorithms outlined in Chapter 4. Here we give an example application of these methods and report on their performance.

We use two example groups for our computations. The group G is a polycyclic subgroup of $GL(4, \mathbb{Z})$ defined by the 6 generators appended below in Section 10.4.1. The second group H is a polycyclic subgroup of $GL(16, \mathbb{Z})$ an it is defined by 9 generators. We omit the explicit description of H here to save space. Both groups have originally been constructed by Nebe and Plesken.

As described in Section 5.1 we start our investigations by considering a p-congruence homomorphism ψ_p for an odd prime p. As a first step we determine a constructive polycyclic sequence for its image using the methods of Chapter 4. The following table includes runtimes in seconds for such computations.

	prime 3		prime 5		prime 7	
	order	time	order	time	order	time
G	144	0.08	600	0.08	384	0.08
H	69120	3.3	288000	16.9	368640	20.4

Now we consider $I_3(G)$ and its corresponding kernel $K = K_3(G)$ in more detail. First, we determine a normal subgroup generating set $\mathcal{E}$ of length 10 for K in 0.02 seconds by evaluating the relations of $I_3(G)$. As next step we investigate the module structure of $\mathbb{Q}^4$ under the action of K as described in Sections 5.2.2 and 5.4.3. We apply the spinning algorithm to determine a basis for the matrix algebra $\mathbb{Q}(K)$ in 0.07 seconds. Using this, it is straightforward to observe that K is abelian. Further, the algebra $\mathbb{Q}(K)$ has dimension 4 and we can choose an element from $\mathcal{E}$ as generator c for $\mathbb{Q}(K) = \mathbb{Q}(c)$. By factorizing the minimal polynomial of c we observe that $\mathbb{Q}^4$ is irreducible as $\mathbb{Q}(K)$-module.
Hence we obtain that K is free abelian and we apply the method of Section 5.5.1 to determine a multiplicatively independent generating set for K. We calculate the p-adic relation lattice $rl_\pi(\mathcal{E})$ for $\pi = \{2, 3, 5, 7, 11, 13\}$ and apply the LLL-algorithm to the lattice $rl_\pi(\mathcal{E})$ to find a basis of short vectors. We then test which of the obtained basis vectors is a relation for $\mathcal{E}$. This computation needs 0.4 seconds altogether and it yields 7 relations between the generators $\mathcal{E}$. We can now read off a generating set $\mathcal{K}$ for K of length 3. As next step we verify that the elements in $\mathcal{K}$ are multiplicatively independent. For this purpose we compute the Dixon bound $B = 1571963904$ for the generating set $\mathcal{K}$ in 0.07 seconds. Using this, we can verify in 4.0 seconds that $\mathcal{K}$ is multiplicatively independent by applying the p-adic relation lattice computation again.
Similarly, by using the p-adic relation finding process, we can observe that $\mathcal{K}$ generates K as a subgroup. Combining a sequence of preimages of a constructive polycyclic sequence for $I_3(G)$ with $\mathcal{K}$ we thus obtain a constructive polycyclic sequence for the considered group G.

The structure of H is similar to that of G. For exhibiting this we consider the kernel $K = K_3(H)$ and we compute a normal subgroup generating set $\mathcal{E}$ of length 21 for K in 1.3 seconds. We observe that $\mathbb{Q}^{16}$ is a homogeneous $\mathbb{Q}(K)$-module consisting of four direct irreducible summands and hence we obtain that K is free abelian.
Now we apply the methods of Section 5.5.1 to find multiplicatively independent generators for K. We determine the p-adic relation lattice $rl_\pi(\mathcal{E})$ with $\pi = \{2, 3, 5, 7, 11\}$ and apply the LLL-algorithm to $rl_\pi(\mathcal{E})$ to find relations. This takes 8.8 seconds and yields 18 relations between the generators $\mathcal{E}$. We can read off a generating set $\mathcal{K}$ of K with 3 generators.
As above, it turns out that the sequence $\mathcal{K}$ forms a polycyclic generating sequence for K. However, verifying this using the Dixon bound for $\mathcal{K}$ is more difficult. In particular, the determination of the Dixon bound for this problem is too time-consuming to be considered practical. Its problems arise from computations with integer matrices having large entries and the factorization of large numbers into primes. It might be worth noting that applying the p-adic relation lattice method for increasing sets of primes to $\mathcal{K}$ again leads to the observation that the length of the short vectors in the relation lattice $rl_\pi(\mathcal{K})$ is growing rapidly. This might be used as an indication that $\mathcal{K}$ is multiplicatively independent.

In summary, the determination of constructive polycyclic sequences for integral matrix groups is practical in small dimensions as shown above using the 4-dimensional group G. In larger dimensions it is possible to obtain useful results with this algorithm as the investigation of the 16-dimensional group H shows, but the complete algorithm is too time-consuming in this application to H.

10.4.1 Generators for $G \leq GL(4, \mathbb{Z})$

The following polycyclic subgroup of $GL(4, \mathbb{Z})$ has been described by Nebe and Plesken.

$$\langle \begin{pmatrix} 5 & 3 & 3 & -5 \\ 7 & 4 & 5 & -8 \\ 2 & 1 & 2 & -2 \\ 10 & 6 & 7 & -11 \end{pmatrix}, \begin{pmatrix} -2 & -4 & -3 & 5 \\ 4 & 7 & 5 & -8 \\ -3 & -3 & -3 & 4 \\ 1 & 2 & 1 & -2 \end{pmatrix}, \begin{pmatrix} 1 & -1 & 0 & 1 \\ -4 & -2 & -2 & 5 \\ -4 & -2 & -3 & 3 \\ -3 & -2 & -2 & 4 \end{pmatrix},$$
$$\begin{pmatrix} 8 & 8 & 3 & -10 \\ -13 & -14 & -5 & 17 \\ -13 & -14 & -5 & 18 \\ -8 & -9 & -3 & 11 \end{pmatrix}, \begin{pmatrix} 0 & -1 & -1 & 2 \\ 2 & 2 & 1 & -1 \\ 2 & 3 & 1 & -5 \\ 3 & 3 & 1 & -3 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ -7 & -8 & -3 & 11 \\ -1 & -3 & -2 & 7 \end{pmatrix} \rangle$$

Chapter 11

Open problems and final comments

The methods outlined in the previous chapter provide a basis for investigations in polycyclic groups by computational methods. But for a variety of interesting problems in polycyclic groups there is no practical algorithm known yet and some questions are not even known to be decidable in infinite polycyclic groups.

We outline a collection of open problems of this kind here. The problems are all aiming at infinite polycyclic groups. The corresponding problems for finite polycyclic groups are all satisfactorily solved.

11.1 Frattini subgroup

The Frattini subgroup $\Phi(G)$ of a polycyclic group G is the intersection of its maximal subgroups. Clearly, $\Phi(G)$ is a characteristic subgroup of G. Hirsch has shown that the Frattini subgroup of a polycyclic group is nilpotent and hence we obtain $\Phi(G) \leq Fit(G)$ if G is polycyclic.

It has been shown by Baumslag et. al. [3] that the Frattini subgroup of a polycyclic group can be determined algorithmically, but no practical method is known. Note that the determination of the Fitting subgroup and its Frattini subgroup yield an approximation to the Frattini subgroup as shown in Section 9.8.

For finite polycyclic groups a practical algorithm to determine the Frattini subgroup is described in [19].

11.2 Automorphism group and testing isomorphism

The determination of the automorphism group and testing isomorphism for polycyclic groups are problems which are related to each other like the construction of stabilizers is linked to the computation of orbits. Thus a solution for one problem is likely to yield a dual solution for the other problem.

Segal has shown in [69] that both problems are decidable for polycyclic groups and hence it would be of interest to develop practical methods for these problems.

In the algorithmic theory of finite polycyclic groups practical methods for both tasks are known. Deterministic methods to determine automorphism groups have been introduced by Eick, Leedham-Green & O'Brien [22] for p-groups and by Smith [72] for finite polycyclic groups. An algorithm for isomorphism testing of p-groups has been described by O'Brien in [56]. Random methods for these purposes have also been considered by Besche & Eick in [4].

11.3 Minimal generating sets

A polycyclic group is finitely generated and thus we can ask if we can find a generating set of minimal cardinality. It is not known whether this question is algorithmically decidable.
For finitely generated nilpotent groups the question is quite easy to solve and for finite polycyclic groups an effective approach has been introduced by Luccini & Menegazzo in [46].

11.4 Residual nilpotence

A group G is residually nilpotent if there exists a set of normal subgroups $\mathcal{N}$ such that G/N is nilpotent for each $N \in \mathcal{N}$ and $\cap_{N\in\mathcal{N}} N = 1$. Each polycyclic group has a normal subgroup of finite index which is residually nilpotent. But there exists polycyclic groups which are not residually nilpotent; e.g. the symmetric group on three symbols.
It is not known whether it is algorithmically possible to decide if a given polycyclic group is residually nilpotent.

11.5 Randomized methods

All algorithms outlined in this book are deterministic. However, in the algorithmic theory of finite groups randomized algorithm have gained importance over the last years.
Thus it may also be interesting to exploit randomized methods for infinite polycyclic groups. On the one hand, they may lead to much more effective algorithms. It could also be useful to consider such methods for problems which are not decidable by deterministic methods.

11.6 Extensions to wider classes of groups

In our algorithms for polycyclic groups we often first determine a linear abelian series and then we reduce the desired computation to questions concerning polycyclic finite dimensional matrix groups over finite fields or the integers. The basis for all these methods is provided by the fact that we can compute with finite dimensional vector spaces over finite fields or with finite dimensional integral lattices.
These underlying methods are to some extent also available for finitely generated vector spaces over the rational numbers. Thus it seems an interesting and ambitious question to investigate if we can extend our investigations to solvable groups which have a normal series whose factors are either finite dimensional vector spaces over finite fields or finite dimensional vector spaces over a subring of the rational numbers. Clearly, such groups may have subgroups which are not finitely generated.

Bibliography

[1] L. Auslander. On a problem of Philip Hall. *Ann. of Math. (2)*, 86:112 – 116, 1967.

[2] G. Baumslag. *Lecture notes on nilpotent groups*. Amer. Math. Soc., Providence, 1971.

[3] G. Baumslag, F. B. Cannonito, D. J. S. Robinson, and D. Segal. The algorithmic theory of polycyclic-by-finite groups. *J. Alg.*, 142:118 – 149, 1991.

[4] H. U. Besche and B. Eick. Construction of finite groups. *J. Symb. Comput.*, 27:387 – 404, 1999.

[5] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *J. Symb. Comput.*, 24:235 – 265, 1997.

[6] H. Brown, R. Bülow, J. Neubüser, H. Wondratschek, and H. Zassenhaus. *Crystallographic Groups of Four-Dimensional Space*. John Wiley, New York, 1978.

[7] H. Brückner. Algorithmen für endliche auflösbare Gruppen und ihre Anwendungen. Dissertation, RWTH Aachen, 1998.

[8] G. Butler. *Fundamental Algorithms for Permutation Groups*, volume 559 of *Lecture Notes in Comput. Sci.* Springer-Verlag, New York, Heidelberg, Berlin, 1991.

[9] F. Celler, J. Neubüser, and C. R. B. Wright. Some remarks on the computation of complements and normalizers in finite soluble groups. *Acta Applic. Math*, 21:57 – 76, 1990.

[10] H. Cohen. *A course in computational algebraic number theory*. Springer-Verlag, New York, Heidelberg, Berlin, 1993.

[11] W. de Graaf and W. Nickel. Constructing faithful representations of finitely-generated torsion-free nilpotent groups. *In preparation*.

[12] K. Dekimpe. *Almost-Bieberbach Groups: Affine and Polynomial Structures*, volume 1639 of *Lecture notes in Math.* Springer, 1996.

[13] K. Dekimpe and B. Eick. *Aclib*, 2000. A GAP share package, see [74].

[14] L. E. Dickson. *Algebras and their arithmetics*. University of Chicago, 1923.

[15] J. D. Dixon. *The structure of linear groups*. Van Nostrand Reinhold Company, London, 1971.

[16] J. D. Dixon. The orbit-stabilizer problem for linear groups. *Can. J. Math.*, 37(2):238 – 259, 1985.

[17] B. Eick. Computing with infinite polycyclic groups. In A. Seress and W. M. Kantor, editors, *Groups and Computation III*, Amer. Math. Soc. DIMACS Series. (DIMACS, 1999).

[18] B. Eick. On the Fitting subgroup of a polycyclic-by-finite group and its applications. *Submitted.*

[19] B. Eick. Special presentations of finite soluble groups and computing (pre-) Frattini subgroups. In L. Finkelstein and W. M. Kantor, editors, *Groups and Computation II*, volume 28 of *Amer. Math. Soc. DIMACS Series*, pages 101 – 112. (DIMACS, 1995), 1997.

[20] B. Eick, F. Gähler, and W. Nickel. Computing maximal subgroups and Wyckoff positions of space groups. *Acta Cryst. A*, 53:467 – 474, 1997.

[21] B. Eick, F. Gähler, and W. Nickel. *CrystGap*, 1998. A GAP share package, see [74].

[22] B. Eick, C. R. Leedham-Green, and E. A. O'Brien. Constructing automorphism groups of p-groups. *Submitted.*

[23] B. Eick and W. Nickel. *Polycyclic*, 2000. A GAP share package, see [74].

[24] B. Eick and A. C. Niemeyer. An infinite polycyclic quotient algorithm. *In preparation.*

[25] B. Eick and G. Ostheimer. On the orbit stabilizer problem for polycyclic linear groups. *In preparation.*

[26] W. Gaschütz. Zur Erweiterungstheorie der endlichen Gruppen. *J. reine und angewandte Math.*, 190:93 – 107, 1952.

[27] K. Geissler and J. Klüners. Galois group computation for rational polynomials. *J. Symb. Comput.*, To appear.

[28] S. P. Glasby and M. C. Slattery. Computing intersections and normalizers in soluble groups. *J. Symb. Comput.*, 9:637 – 651, 1990.

[29] K. A. Hirsch. On infinite soluble groups (I). *Proc. London Math. Soc.*, 44(2):53 – 60, 1938.

[30] K. A. Hirsch. On infinite soluble groups (II). *Proc. London Math. Soc.*, 44(2):336 – 414, 1938.

[31] K. A. Hirsch. On infinite soluble groups (III). *J. London Math. Soc.*, 49(2):184 – 94, 1946.

[32] K. A. Hirsch. On infinite soluble groups (IV). *J. London Math. Soc.*, 27:81 – 85, 1952.

[33] K. A. Hirsch. On infinite soluble groups (V). *J. London Math. Soc.*, 29:250 – 251, 1954.

[34] D. F. Holt, C. R. Leedham-Green, E. A. O'Brien, and S. Rees. *Smash – matrix groups and G-modules*, 1995. A GAP share package, see [74].

[35] D. F. Holt, C. R. Leedham-Green, E. A. O'Brien, and S. Rees. Computing matrix group decompositions with respect to a normal subgroup. *J. Alg.*, 184:818 – 838, 1996.

[36] D. F. Holt and S. Rees. Testing modules for irreducibility. *J. Austral. Math. Soc. (Series A)*, 57:1 – 16, 1994.

[37] A. Hulpke. Computing subgroups invariant under a set of automorphisms. *J. Symb. Comput.*, 27:415 – 427, 1999.

[38] R. Laue, J. Neubüser, and U. Schoenwaelder. Algorithms for finite soluble groups and the SOGOS system. In *Computational Group Theory*, pages 105 – 135, London, New York, 1984. (Durham, 1982), Academic Press.

[39] C. R. Leedham-Green, C. E. Praeger, and L. H. Soicher. Computing with group homomorphisms. *J. Symb. Comput.*, 12:527 – 532, 1991.

[40] C. R. Leedham-Green and L. H. Soicher. Collection from the left and other strategies. *J. Symbolic Comput.*, 9:665 – 675, 1990.

[41] C. R. Leedham-Green and L. H. Soicher. Symbolic collection using Deep Thought. *LMS J. Comput. Math.*, 1:9 – 24, 1998.

[42] E. H. Lo. A polycyclic quotient algorithm. Phd Thesis, Rutgers University, 1996.

[43] E. H. Lo. Enumerating finite index subgroups of polycyclic groups. Unpublished report, 1998.

[44] E. H. Lo. Finding intersection and normalizer in finitely generated nilpotent groups. *J. Symbol. Comput.*, 25:45 – 59, 1998.

[45] E. H. Lo and G. Ostheimer. A practical algorithm for finding matrix representations for polycyclic groups. *J. Symbol. Comput.*, 28:339 – 360, 1999.

[46] A. Lucchini and F. Menegazzo. Computing a set of generators of minimal cardinality in a solvable group. *J. Symb. Comput.*, 17:409 – 420, 1994.

[47] E. M. Luks. Computing in solvable matrix groups. In *Proc. 33rd IEEE Sympos. Foundations Comp. Sci.*, pages 111 – 120, 1992.

[48] K. Lux. Algorithmic methods in modular representation theory. Habilitationsschrift, RWTH Aachen, 1997.

[49] A. J. Mal'cev. On certain classes of infinite soluble groups. *Mat. Sb.*, 28:567 – 588, 1951.

[50] M. Mecky and J. Neubüser. Some remarks on the computation of conjugacy classes of soluble groups. *Bull. Austr. Math. Soc.*, 40:281 – 292, 1989.

[51] W. Müller. *Darstellungstheorie von endlichen Gruppen.* Teubner, Stuttgart, 1980.

[52] M. F. Newman. The soluble length of soluble linear groups. *Math. Z.*, 126:59 – 70, 1972.

[53] M. F. Newman and E. A. O'Brien. Application of computers to questions like those of Burnside, II. *Internat. J. Algebra Comput.*, 6:593 – 605, 1996.

[54] W. Nickel. Computing nilpotent quotients of finitely presented groups. In *Geometric and computational perspectives on infinite groups*, Amer. Math. Soc. DIMACS Series, pages 175 – 191. (DIMACS, 1994), 1995.

[55] A. C. Niemeyer. A finite soluble quotient algorithm. *J. Symb. Comput.*, 18:541 – 561, 1994.

[56] E. A. O'Brien. Isomorphism testing for p-groups. *J. Symb. Comput.*, 17:133 – 147, 1994.

[57] G. Ostheimer. Algorithms for polycyclic-by-finite groups. Phd Thesis, Rutgers University, 1996.

[58] G. Ostheimer. Algorithms for polycyclic-by-finite matrix groups. In L. Finkelstein and W. M. Kantor, editors, *Groups and Computation II*, volume 28 of *Amer. Math. Soc. DIMACS Series*, pages 297 – 307. (DIMACS, 1995), 1997.

[59] G. Ostheimer. Practical algorithms for polycyclic matrix groups. *J. Symb. Comput.*, 28:361 – 379, 1999.

[60] P. A. Parker. The computer calculation of modular characters (The Meat-Axe). In *Computational Group Theory*, pages 267 – 274, London, New York, 1984. (Durham, 1982), Academic Press.

[61] R. A. Parker. An integral meataxe. In *The atlas of finite groups: ten years on (Birmingham, 1995)*, Lecture Notes Ser. 249, pages 215 – 228. London Math. Soc., 1998.

[62] W. Plesken. Towards a soluble quotient algorithm. *J. Symb. Comput.*, 4:111 – 122, 1987.

[63] M. Pohst and H. Zassenhaus. *Algorithmic algebraic number theory.* Cambridge University Press, 1989.

[64] D. J. S. Robinson. *Finiteness conditions and generalized soluble groups I*, volume 62 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, New York, Heidelberg, Berlin, 1972.

[65] D. J. S. Robinson. Splitting theorems for infinite groups. *Symp. Math.*, 17:441 – 470, 1975.

[66] D. J. S. Robinson. The vanishing of certain homology and cohomology groups. *J. Pure Applied Math.*, 7:145 – 167, 1976.

[67] D. J. S. Robinson. *A Course in the Theory of Groups*, volume 80 of *Graduate Texts in Math.* Springer-Verlag, New York, Heidelberg, Berlin, 1982.

[68] D. Segal. *Polycyclic Groups.* Cambridge University Press, Cambridge, 1983.

[69] D. Segal. Decidable properties of polycyclic groups. *Proc. London Math. Soc.* (3), 61:497 – 528, 1990.

[70] C. C. Sims. Computing the order of a solvable permutation group. *J. Symb. Comput.*, 9:699 – 705, 1990.

[71] C. C. Sims. *Computation with finitely presented groups.* Cambridge University Press, Cambridge, 1994.

[72] M. J. Smith. Computing automorphisms of finite soluble groups. Phd Thesis, Australian National University, 1994.

[73] R. G. Swan. Representations of polycyclic groups. *Proc. Amer. Math. Soc.*, 18:573 – 574, 1967.

[74] The GAP Team. *GAP – Groups, Algorithms and Programming.* Lehrstuhl D für Mathematik, RWTH Aachen, and School of Mathematical and Computational Sciences, University of St. Andrews, 1994.

[75] H. Theißen. Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen. Dissertation, RWTH Aachen, 1997.

[76] B. A. F. Wehrfritz. *Infinite linear groups.* Springer-Verlag, New York, Heidelberg, Berlin, 1973.

Table of algorithms

Index