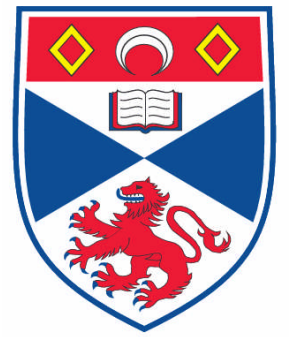
APPLICATIONS OF LIE METHODS TO COMPUTATIONS WITH POLYCYCLIC GROUPS

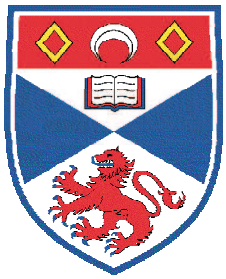
Björn Assmann

A Thesis Submitted for the Degree of PhD
at the
University of St. Andrews



2007

Applications of Lie methods to computations with polycyclic groups



A thesis to be submitted to the
UNIVERSITY OF ST ANDREWS
for the degree of
DOCTOR OF PHILOSOPHY

by
Björn Assmann

School of Computer Science
University of St Andrews
September 12, 2007

I, Björn Assmann, hereby certify that this thesis, which is approximately 31000 words in length, has been written by me, that it is the record of work carried out by me, and that it has not been submitted in any previous application for a higher degree.

date _____ *signature of candidate* _____

I was admitted as a research student in October 2004 and as a candidate for the degree of Doctor of Philosophy in October 2004; the higher study for which this is a record was carried out in the University of St Andrews between 2004 and 2007.

date _____ *signature of candidate* _____

I, Stephen Linton, hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

date _____ *signature of supervisor* _____

In submitting this thesis to the University of St. Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published, and that a copy of the work may be made and supplied to any *bona fide* library or research worker.

date _____ *signature of candidate* _____

Abstract

In this thesis we demonstrate the algorithmic usefulness of the so-called Mal'cev correspondence for computations with infinite polycyclic groups. This correspondence between $\mathbb{Q}$-powered nilpotent groups and rational nilpotent Lie algebras was discovered by Anatoly Mal'cev in 1951.

We show how the Mal'cev correspondence can be realized on a computer. We explore two possibilities for this purpose and compare them: the first one uses matrix embeddings and the second the Baker–Campbell–Hausdorff formula.

Then, we describe a new collection algorithm for polycyclically presented groups, which we call Mal'cev collection. Algorithms for collection lie at the heart of most methods dealing with polycyclically presented groups. The current state of the art is "collection from the left" as recently studied by Gebhardt, Leedham-Green/Soicher and Vaughan-Lee. Mal'cev collection is in some cases dramatically faster than collection from the left, while using less memory.

Further, we explore how the Mal'cev correspondence can be used to describe symbolically the collection process in polycyclically presented groups. In particular, we describe an algorithm that computes the collection functions for splittable polycyclic groups. This algorithm is based on work by du Sautoy. We apply it to the computation of pro-p-completions of polycyclic groups.

Finally we describe a practical algorithm for testing polycyclicity of finitely generated rational matrix groups. Previously, not only did no such method exist but it was not clear whether this question was decidable at all.

Most of the methods described in this thesis are implemented in the computer algebra system GAP and publicly available as part of the GAP packages Guarana and Polenta. Reports on the implementation including runtimes for some examples are given at the appropriate places.

Acknowledgement

I would like to thank my advisor Stephen Linton for our many inspiring discussions during my time at the University of St Andrews. In particular, I am grateful for his open-minded approach to our mathematical meetings where anything could be brought on the table. Although my work benefitted tremendously from our conversations, they are most memorable because they were genuinely enjoyable.

I would like to thank my second advisor Martyn Quick for his competent advice on infinite group theory and for carefully reading the first draft of this thesis.

Also, I am grateful to Bettina Eick for introducing me to the area of computational group theory, which became a very enjoyable mathematical playground for the years to follow.

Many thanks go to my office mate Peter Nightingale for his patient answers to my questions related to the English language and for being an excellent riddle partner.

I gratefully acknowledge the financial support of the Gottlieb Daimler- und Karl Benz-Stiftung and the UK Engineering and Physical Science Research Council.

Finally I would like to thank my parents and my fiancée Moni for their unwavering support during my work on this thesis.

Contents

Chapter 1

Introduction

A group is the algebraic concept to describe symmetry. As a consequence, the theory of groups can be applied to various areas of science, for example geometry, number theory, crystallography, quantum mechanics and constraint programming.

Because of its generality, the definition of a group allows very complicated structures. Thus, it is natural to apply a well-known approach from other sciences to the study of groups. Namely given a group G, decompose it into smaller more understandable pieces, study those, and finally study their interaction to reveal the structure of G.

The smallest pieces in group theory are called simple groups. One of the biggest algebraic projects of the last century was the classification of the finite simple groups. Its aim was the creation of a table that contains all finite simple groups up to isomorphism.

The theory of polycyclic groups aims in the opposite direction. It does not try to classify the atoms of group theory but asks: What can a group look like that is made out of easy pieces namely cyclic groups? Given the restrictiveness of this question it is rather surprising that the class of polycyclic groups has a rich theory with interesting links to other areas, in particular number theory. It shows that the mechanism for building groups out of smaller ones, essentially the interaction between factor groups and normal subgroups, is rather complex.

Historically polycyclic groups were considered right from the start of group theory. Evariste Galois showed in the first half of the nineteenth century that a rational polynomial is soluble by radicals if and only if the symmetry group of its solutions is polycyclic. The foundations of the structural explorations of polycyclic groups were laid roughly 100 year later by Kurt Hirsch, Phillip Hall, Reinhold Baer and Anatoly Mal'cev amongst others.

More recently the class of polycyclic groups has also been shown to be very fruitful for computational investigations, see for example [17, 39]. Polycyclic groups can be efficiently represented on a computer by means of a special kind of finite presentation, which is called a polycyclic presentation. If a group is given with respect to a polycyclic presentation, then various properties of the group can explored algorithmically. For example it is possible to test membership in subgroups, to compute the normal closure of subgroups and to determine the derived series.

The aim of this thesis is to show how Lie methods can be applied to the algorithmic investigation of polycyclic groups. The connection between groups and Lie rings, respectively Lie algebras, is a well-known and mathematically very useful concept. For example, a typical way to solve a problem in a Lie group is to transfer the problem to the Lie algebra of the group, study it there with the help of tools from linear algebra and transfer the result back into the Lie group.

Mechanisms of this kind have already been shown to be useful for the exploration of finite polycyclic groups. For instance, Vaughan-Lee and O'Brien used Lie ring techniques to construct a consistent polycyclic presentation of $R(2,7)$, the largest 2-generator finite group of exponent 7 [34].

In this thesis we demonstrate the algorithmic usefulness of the so-called Mal'cev correspondence for computations with infinite polycyclic groups. This correspondence between $\mathbb{Q}$-powered nilpotent groups and rational nilpotent Lie algebras was discovered by Anatoly Mal'cev in 1951 [27].

After background material on polycyclic groups in Chapter 2 and on the Mal'cev correspondence in Chapter 3, we show in Chapter 4 how the Mal'cev correspondence can be realized on a computer. We explore two possibilities for this purpose and compare them: the first one uses matrix embeddings and the second the Baker–Campbell–Hausdorff formula.

Then, in Chapter 5, we describe a new collection algorithm for polycyclically presented groups, which we call Mal'cev collection. Every element of a polycyclically presented group has a unique normal form. An algorithm for computing this normal form is called a collection algorithm. Such an algorithm lies at the heart of most methods dealing with polycyclically presented groups. The current state of the art is "collection from the left" [18, 24, 43]. Mal'cev collection is in some cases dramatically faster than collection from the left, while using less memory.

In Chapter 6 we explore how the Mal'cev correspondence can be used to describe symbolically the collection process in polycyclically presented groups. In particular, we describe an algorithm that computes the collection functions for splittable polycyclic groups. This algorithm can be seen as an extension of the algorithm "Deep Thought" by Leedham-Green and

Soicher, which computes collection functions for finitely generated torsion-free nilpotent groups. We apply it to the computation of pro-p-completions of polycyclic groups.

Finally in Chapter 7 we describe a practical algorithm for testing polycyclicity of finitely generated rational matrix groups. Previously, not only did no such method exist but it was not clear whether this question was decidable at all. The contents of this chapter are based on a joint project with Bettina Eick, see the remark at the beginning of Chapter 7.

Most of the methods described in this thesis are implemented in the computer algebra system GAP [41] and publicly available as part of the GAP packages Guarana [2] and Polenta [3]. A CD containing this software is attached to this book. Reports on the implementation including runtimes for some examples are give at the appropriate places. Several results presented in this book have been published in mathematical journals, see [1, 5, 6].

Chapter 2

Polycyclic groups

In this chapter we recall some well-known results about polycyclic groups. For more background on the theory of polycyclic groups we refer to [37, 38]; further information about computations with polycyclic groups can be found in [17, 20, 39].

2.1 Preliminaries

This section provides some basic group theoretic facts. For proofs and further information we refer to [37].

Let G be a group. A subset $H \subseteq G$ is called a *subgroup* if H contains the identity of G and is closed under multiplication and inversion; if H is a subgroup of G, then we write $H \leq G$. A subgroup $H \leq G$ is said to be a *normal subgroup* of G, denoted $H \trianglelefteq G$, if $H^g = g^{-1}Hg \subseteq H$ for all g in G.

We denote by G/H the set of all cosets of H in G, i.e. $G/H = \{gH | g \in G\}$. The *index* of H in G, denoted $[G : H]$, is the cardinality of G/H. If H is normal in G then we can define a group multiplication on G/H by $gHkH = gkH$; note that this multiplication is well defined if and only if H is normal in G. If $H \trianglelefteq G$ then we call G/H the factor group of G by H.

If X is a nonempty subset of a group G, then we denote by $\langle X \rangle$ the set of all elements of the form $x_1^{\epsilon_1} \cdots x_k^{\epsilon_k}$ where $\epsilon_i = \pm 1$, $x_i \in X$ and $k \geq 0$. For $k = 0$ the product is defined to be the identity. Note that $\langle X \rangle$ is a group and furthermore the smallest subgroup of G that contains X. We say that a group G is *finitely generated* if $G = \langle X \rangle$ for some finite subset $X \subseteq G$. A group G is said to be *cyclic* if $G = \langle \{x\} \rangle$ for some $x \in G$. In this case we usually write $G = \langle x \rangle$. Let H be a subgroup of finite index in G. Then G is finitely generated if and only if H is finitely generated.

Let $H \leq G$. We define the *core* of H in G, denoted H_G, to be the

intersection of all conjugates of H, i.e. $H_G = \cap_{g \in G} H^g$. Equivalently H_G can be defined as the biggest normal subgroup of G which is contained in H. Note that $[G : H] < \infty$ implies that $[G : H_G] < \infty$. By H^G we denote the smallest normal subgroup of G that contains H.

Let G and H be two groups. A function $\varphi : G \to H$ is called a *homomorphism* if $\varphi(xy) = \varphi(x)\varphi(y)$ for all $x, y \in G$. Sometimes we will write x^φ instead of $\varphi(x)$. A homomorphism which is bijective is called an *isomorphism*. If an isomorphism $\varphi : G \to H$ exists, then we say that G and H are *isomorphic* and write $G \cong H$. A homomorphism $\varphi : G \to G$ is said to be an *endomorphism*. An endomorphism which is bijective is called an *automorphism*.

The *kernel* of a homomorphism $\varphi : G \to H$ is $\operatorname{Ker}\varphi = \{g \in G | g^\varphi = 1\}$. The *image* of φ is $\operatorname{Im}\varphi = \{g^\varphi | g \in G\}$. The kernel $\operatorname{Ker}\varphi$ is a normal subgroup of G. The factor group $G/\operatorname{Ker}\varphi$ is naturally isomorphic to $\operatorname{Im}\varphi$ via $g \operatorname{Ker}\varphi \mapsto g^\varphi$.

We denote the set all automorphisms of a group G by $\operatorname{Aut}(G)$. Note that $\operatorname{Aut}(G)$ is a group where multiplication is the composition of functions. A subgroup $H \leq G$ is said to be *characteristic* if $H^\varphi = H$ for all φ in $\operatorname{Aut}(G)$.

The *order* of a group G, denoted $|G|$, is the number of elements in G. The order of an element $g \in G$ is the order of the cyclic group $\langle g \rangle$. If $\langle g \rangle$ is infinite then g has *infinite order*; otherwise g has *finite order*. A group is said to be a *torsion group* if all its elements have finite order. On the other hand a group is called *torsion-free* if all its elements apart from the identity have infinite order.

Let $\mathcal{P}, \mathcal{Q}$ be properties of groups. A group G is called a *$\mathcal{P}$-by-$\mathcal{Q}$-group* if G has a normal subgroup H such that H has $\mathcal{P}$ and G/H has $\mathcal{Q}$. We call G an *extension* of a group A by a group B if there exists a normal subgroup $H \trianglelefteq G$ such that $H \cong A$ and $G/H \cong B$.

A group G is said to be *abelian* if $gh = hg$ for all g, h in G; equivalently G is abelian if and only if the commutator $[g, h] = g^{-1}h^{-1}gh$ is trivial for all g, h in G. A group G is said to be *free abelian* if it is isomorphic to the direct product of a (possibly infinite) number of copies of $\mathbb{Z}$. A free abelian group G is said to be of *finite rank* r if the size of a minimal generating set of G is r for some $r \in \mathbb{N}$; such a generating set is called a *free generating set* of G. Note that in this case $G \cong \mathbb{Z}^r$.

Let $X = \{x_1, \ldots, x_r\}$ be a set of formal letters and denote by X^{-1} the set of its formal inverses $\{x_1^{-1}, \ldots, x_r^{-1}\}$. The *free group F on X* is the set of all words $w(X)$ in $X \cup X^{-1}$ with conjunction as multiplication, where two words w and v are identified if w can be obtained from v via a finite number of insertions and deletions of expressions of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$. If F is free on a set X of cardinality r, then F is a free group of *rank* r.

Let F be a free group on a finite set $X = \{x_1, \ldots, x_r\}$ and let R be a finite subset of F. Let $K = \langle R \rangle^F$ be the smallest normal subgroup of F that contains R. By $\langle X|R\rangle$ we denote the factor group F/K. We say that $\langle X|R\rangle$ is a *finitely presented group* with generators X and relators R. Sometimes the relators $r \in R$ are given via defining relations of the form $r = 1$.

Let G be a group generated by a set $S = \{g_1, \ldots, g_r\}$. The set S is said to *satisfy* the relations of $\langle X|R\rangle$ if for all words $w(x_1, \ldots, x_r) \in R$ we have $w(g_1, \ldots, g_r) = 1$ in G. If S satisfies the relations of $\langle X|R\rangle$ then there exists an epimorphism, i.e. a surjective homomorphism, $\varphi : \langle X|R\rangle \to G$ with $x_i \mapsto g_i$.

2.1.1 Poly-$\mathcal{P}$ groups

Usually abelian groups are considered to be nice in the sense that they are easier to investigate then non-abelian groups. For this reason it is natural to try to measure how close a group is to being an abelian group. For this purpose series of subgroups are often used.

Definition 2.1.1. Let $\mathcal{P}$ be a property of groups. We say that a group G is *poly-$\mathcal{P}$* if there exists a subnormal series of G

$$G = G_1 \rhd G_2 \rhd \cdots G_n \rhd G_{n+1} = 1$$

such that every factor G_i/G_{i+1} has the property $\mathcal{P}$. In particular, a group G is *polycyclic* if G_i/G_{i+1} is cyclic for $i = 1, \ldots, n$.

The *Hirsch length* $\mathrm{Hl}(G)$ of a polycyclic group G is defined to be the number of infinite factors in a subnormal series with cyclic factors. It is known to be an invariant of the group.

A polycyclic group is not necessarily abelian. However cyclic groups are, and thus a polycyclic group can be considered to be close to being an abelian group.

2.1.2 Soluble groups

Definition 2.1.2. A group G is said to be *soluble* if it is polyabelian.

The motivation for the term "soluble" originates from Galois theory; a rational polynomial is soluble by radicals if and only if its Galois group is soluble. By definition every polycyclic group is soluble.

The *derived group* G' of a group G is defined to be the subgroup generated by all commutators $[g, h] = g^{-1}h^{-1}gh$ with $g, h \in G$. For $n \geq 1$ we

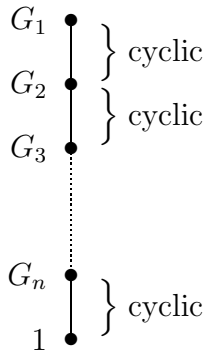


Figure 2.1: A group G is said to be polycyclic if it has subnormal series of finite length with cyclic factors, i.e. $G_i \trianglelefteq G_{i+1}$ and G_i/G_{i+1} cyclic for $i = 1, \ldots, n$.

define $G^{(n)} = (G^{(n-1)})'$, where $G^{(0)} = G$. This yields a descending series of subgroups

$$G = G^{(0)} \geq G' = G^{(1)} \geq G^{(2)} \geq \ldots$$

called the *derived series* of G. If $\varphi \in \mathrm{Aut}(G)$ then $[g,h]^\varphi = [g^\varphi, h^\varphi]$. Thus all subgroups of the derived series are characteristic.

By definition G/G' is abelian. Furthermore G' is the smallest normal subgroup with that property, i.e. if $N \trianglelefteq G$ and G/N is abelian then $N \geq G'$. It is known that a group G is soluble if and only if $G^{(n)} = 1$ for some $n \in \mathbb{N}$. The least n for which $G^{(n)} = 1$ is called the *derived length* of G. As a consequence G is soluble if and only if G has a series of normal subgroups of finite length with abelian factors.

2.1.3 Nilpotent groups

The centre $\zeta_1(G)$ of a group G is the set of elements in G which commute with everything else, i.e. $\zeta_1(G) = \{g \in G | [g,h] = 1 \text{ for all } h \in G\}$. An element $g \in G$ is called *central* if $g \in \zeta_1(G)$. The centre of a group is an abelian characteristic subgroup. The *upper central series*

$$1 = \zeta_0(G) \leq \zeta_1(G) \leq \zeta_2(G) \leq \ldots$$

of a group G is recursively defined by $\zeta_i(G)/\zeta_{i-1}(G) = \zeta_1(G/\zeta_{i-1}(G))$. It is an ascending chain of characteristic subgroups. By definition, $x \in \zeta_i(G)$ if and only if $[x,g] \in \zeta_{i-1}(G)$ for all g in G.

Definition 2.1.3. We say that a group G is *nilpotent* if $\zeta_c(G) = G$ for some

$c \in \mathbb{N}$. The smallest $c \in \mathbb{N}$ such that $\zeta_c(G) = G$ is called the *nilpotency class* of G.

A series of subgroups $1 = H_0 \leq H_1 \leq \cdots \leq H_k = G$ is said to be a *central series* of G if $[H_i, G] \subseteq H_{i-1}$ for $i = 1, \ldots, k$. The group G is nilpotent if and only if it has a central series.

The *lower central series* of a group G is defined by $\gamma_1(G) = G$ and $\gamma_{n+1}(G) = [\gamma_n(G), G]$. The group G is nilpotent if and only if $\gamma_{c+1}(G) = 1$ for some $c \in \mathbb{N}$; actually the smallest such c is the nilpotency class of G.

Definition 2.1.4. A group G is said to be a $\mathcal{T}$*-group* if it is finitely generated torsion-free nilpotent.

2.2 Polycyclic sequences

Definition 2.2.1. Let G be a polycyclic group with a subnormal series $G = G_1 \rhd G_2 \rhd \cdots \rhd G_n \rhd G_{n+1} = 1$ with non-trivial cyclic factors G_i/G_{i+1}. A list $\mathcal{G} = (g_1, \ldots, g_n)$ is called a *polycyclic sequence* for G if $G_i/G_{i+1} = \langle g_i G_{i+1} \rangle$. As a consequence $G_i = \langle g_i, \ldots, g_n \rangle$. We call the chain of subgroups $(G_i)_{1 \leq i \leq n}$ the *subgroup series belonging to* $\mathcal{G}$. For every factor G_i/G_{i+1} we denote by $r_i \in \mathbb{N} \cup \{\infty\}$ the index of G_{i+1} in G_i. We call $(r_1, \ldots, r_n)$ the *relative orders* of $\mathcal{G}$.

Lemma 2.2.2. *Let G be a polycyclic group with polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$. If $g \in G$ then we can write g uniquely as*

$$g = g_1^{e_1} \cdot \cdots \cdot g_n^{e_n}$$

where $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ and $0 \leq e_i < r_i$ if $r_i < \infty$.

Proof. We prove the lemma via induction on the number of generators n. If $n = 1$, then G is cyclic and the assertion follows. Let's now assume that $n > 1$ and let $g \in G$. Then $gG_2 = g_1^{e_1} G_2$ for a unique $e_1 \in \mathbb{Z}$ where $0 \leq e_1 < r_1$ if $r_1 < \infty$. Since $g_1^{-e_1} g \in G_2$ it follows, by induction assumption, that $g_1^{-e_1} g = g_2^{e_2} \cdot \cdots \cdot g_n^{e_n}$ where $(e_2, \ldots, e_n) \in \mathbb{Z}^n$ and $0 \leq e_i < r_i$ if $r_i < \infty$ for $i = 2, \ldots, n$. Thus the assertion of the lemma follows. □

Definition 2.2.3. The unique expression $g_1^{e_1} \cdots g_n^{e_n}$ from Lemma 2.2.2 is called the *normal form* of g with respect to $\mathcal{G}$. We denote it by $\mathrm{nf}_{\mathcal{G}}(g)$. The list $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ is called the *exponent vector* of g with respect to $\mathcal{G}$. We denote it by $\exp_{\mathcal{G}}(g)$.

Example 2.2.4. Let G be the group generated by the rational matrices

$$g_1 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad g_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Since $g_1^2 = 1$ and $g_2^{g_1} = g_2^{-1} \in \langle g_2 \rangle$ we deduce that $G \rhd \langle g_2 \rangle \rhd 1$ is a subnormal series of G with cyclic factors. Thus G is polycyclic. The list $\mathcal{G} = (g_1, g_2)$ is a polycyclic sequence for G with r relative orders $(2, \infty)$. Let

$$g = \begin{pmatrix} -1 & 2 \\ 0 & 1 \end{pmatrix}.$$

Since $g = g_1 g_2^{-2}$, g is an element of G. The exponent vector of g with respect to $\mathcal{G}$ is $\exp_{\mathcal{G}}(g) = (1, -2)$.

Corollary 2.2.5. *If a group G is polycyclic, then it is finitely generated.*

Definition 2.2.6. Let $\mathcal{G} = (g_1, \ldots, g_n)$ be a polycyclic sequence of a group G with relative orders $(r_1, \ldots, r_n)$. We say that $\mathcal{G}$ is a *basis* of G if $r_i = \infty$ for $i = 1, \ldots, n$. A basis $\mathcal{G}$ is called a *Mal'cev basis* if the subnormal series belonging to $\mathcal{G}$ is a central series of G.

2.3 Polycyclic presentations

Let G be a polycyclic group with a polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$ and relative orders $(r_1, \ldots, r_n)$. Denote by I the finite index set of $\mathcal{G}$, that is $I = \{\, i \mid 1 \leq i \leq n, r_i < \infty \,\}$.

Let $G_i = \langle g_i, \ldots, g_n \rangle$. Since $G_{j+1} \trianglelefteq G_j$, we deduce that $g_i^{g_j^{\pm 1}} \in G_{j+1}$ for $1 \leq j < i \leq n$. Further, since r_i is the order of G_i/G_{i+1}, we see that $g_i^{r_i} \in G_{i+1}$ for $i \in I$. Thus, we can write these expressions as words in the generators $g_{j+1}, \ldots, g_n$ respectively $g_{i+1}, \ldots, g_n$.

Definition 2.3.1. The equations

$$\begin{aligned} g_i^{g_j} &= g_{j+1}^{a(i,j,j+1)} \cdots g_n^{a(i,j,n)} \text{ for } 1 \leq j < i \leq n, \\ g_i^{g_j^{-1}} &= g_{j+1}^{b(i,j,j+1)} \cdots g_n^{b(i,j,n)} \text{ for } 1 \leq j < i \leq n \text{ and } j \notin I \end{aligned}$$

are called the *conjugate relations* and

$$g_i^{r_i} = g_{i+1}^{c(i,i+1)} \cdots g_n^{c(i,n)} \text{ for } i \in I,$$

are said to be the *power relations* of the polycyclic sequence $\mathcal{G}$; the right hand sides are the normal forms of the elements on the left hand sides.

The next theorem shows that the power-conjugate relations of a polycyclic sequence give rise to a finite presentation for the group G.

Theorem 2.3.2. *Let $\mathcal{G}$ be a polycyclic sequence of a polycyclic group G with power-conjugate relations as in Definition 2.3.1. Let F be a free group on the abstract generators in $X = \{x_1, \ldots, x_n\}$. Define R to be the set of relations*

$$\begin{aligned} x_i^{x_j} &= x_{j+1}^{a(i,j,j+1)} \cdots x_n^{a(i,j,n)} \text{ for } 1 \leq j < i \leq n, \\ x_i^{x_j^{-1}} &= x_{j+1}^{b(i,j,j+1)} \cdots x_n^{b(i,j,n)} \text{ for } 1 \leq j < i \leq n \text{ and } j \notin I \\ x_i^{r_i} &= x_{i+1}^{c(i,i+1)} \cdots x_n^{c(i,n)} \text{ for } i \in I. \end{aligned}$$

Then $\langle X|R\rangle$ is a finite presentation for G.

Proof. By the definition of $\langle X|R\rangle$ the generators $\{g_1, \ldots, g_n\}$ of G satisfy the relations of $\langle X|R\rangle$. Thus there exists an epimorphism $\varphi : \langle X|R\rangle \to G$ which maps x_i to g_i. We prove via induction on n that φ is injective. If $n = 1$ then G is a cyclic group of order r_1 and the claim follows. If $n > 1$ we can assume by induction that the restriction of φ to $X_2 = \langle x_2, \ldots, x_n\rangle$ is injective. Let $x \in \langle X|R\rangle$ be such that $\varphi(x) = 1$. By using the relations in R we can rewrite x as $x_1^{e_1}x'$ where $0 \leq e_1 < r_1$ if $r_1 < \infty$ and $x' \in X_2$. Thus $g_1^{e_1} = \varphi(x_1^{e_1}) = \varphi(x'^{-1}) \in \varphi(X_2)$. This implies $e_1 = 0$. Since φ is injective on X_2 and $\varphi(x') = 1$, we deduce that $x' = 1$. Thus $x = 1$ and therefore φ is injective. □

Definition 2.3.3. The finite presentation $\langle X|R\rangle$ of a polycyclic group G of Theorem 2.3.2 is called a *consistent polycyclic presentation* of G.

The term consistent in Definition 2.3.3 refers to the fact that every element in the finitely presented group $\langle X|R\rangle$ from Theorem 2.3.2 has a unique normal form $x_1^{e_1} \cdots x_n^{e_n}$ with $e_i \in \mathbb{Z}$ and $0 \leq e_i < r_i$ for $i \in I$. Throughout this book all polycyclic presentations are consistent. Therefore we will call them simply polycyclic presentations.

Remark 2.3.4. Any finite presentation on abstract generators $x_1, \ldots, x_n$ with relations of the form

$$\begin{aligned} x_i^{x_j} &= x_{j+1}^{a(i,j,j+1)} \cdots x_n^{a(i,j,n)} \text{ for } 1 \leq j < i \leq n, \\ x_i^{x_j^{-1}} &= x_{j+1}^{b(i,j,j+1)} \cdots x_n^{b(i,j,n)} \text{ for } 1 \leq j < i \leq n \text{ and } j \notin I \\ x_i^{r_i} &= x_{i+1}^{c(i,i+1)} \cdots x_n^{c(i,n)} \text{ for } i \in I, \end{aligned}$$

where $I \subseteq \{1, \ldots, n\}$, defines a polycyclic group H. However this presentation may not be consistent, i.e. an element $h \in H$ may have more then one normal form $x_1^{e_1} \cdots x_n^{e_n}$ with $e_i \in \mathbb{Z}$ and $0 \leq e_i < r_i$ for $i \in I$. For more information on this see [17, 39].

Example 2.3.5. Consider the group G defined in Example 2.2.4 with the polycyclic sequence $\mathcal{G} = (g_1, g_2)$. We obtain the power-conjugate relations

$$\begin{aligned} g_2^{g_1} &= g_2^{-1} \\ g_1^2 &= g_2^0 \end{aligned}$$

of $\mathcal{G}$. By Theorem 2.3.2, $\langle x_1, x_2 | x_2^{x_1} = x_2^{-1}, x_1^2 = 1 \rangle$ is a polycyclic presentation for G. Note that G is the infinite dihedral group D_∞.

2.4 Nilpotency, polycyclicity and solubility

In this section we explain the relationship between nilpotency, polycyclicity and solubility.

Let $\mathcal{P}$ be a property of groups. The class of groups having $\mathcal{P}$ is said to be *closed with respect to forming subgroups* if G has $\mathcal{P}$ and $H \leq G$ implies H has $\mathcal{P}$. For example the class of abelian groups is closed with respect to forming subgroups. Similarly the class of groups having $\mathcal{P}$ is called *closed with respect to forming factor groups* if all quotients of its members have $\mathcal{P}$.

Lemma 2.4.1. *Let $\mathcal{P}$ be a property of groups. Suppose that the class of groups having $\mathcal{P}$ is closed with respect to forming subgroups and factor groups. Then the class of groups being poly-$\mathcal{P}$ is closed with respect to forming subgroups and factor groups.*

Proof. Let G be a group with a subnormal series $G = G_1 \rhd G_2 \rhd \cdots \rhd G_n \rhd G_{n+1} = 1$ such that each factor G_i/G_{i+1} has the property $\mathcal{P}$.

Let H be a subgroup of G. Then the groups $(H_i := H \cap G_i)_{1 \leq i \leq n+1}$ form a subnormal series of H. Further

$$H_i/H_{i+1} = H \cap G_i / H \cap G_{i+1} \cong G_{i+1}(H \cap G_i)/G_{i+1} \leq G_i/G_{i+1}.$$

Thus H_i/H_{i+1} has property $\mathcal{P}$ and therefore H is poly-$\mathcal{P}$.

Let N be a normal subgroup of G. Then the groups $(G_iN/N)_{1 \leq i \leq n+1}$ form a subnormal series of G/N. We have

$$(G_iN/N)/(G_{i+1}N/N) \cong G_iN/G_{i+1}N = G_iG_{i+1}N/G_{i+1}N \cong G_i/(G_i \cap G_{i+1}N)$$

Thus $(G_iN/N)/(G_{i+1}N/N)$ is isomorphic to a factor group of G_i/G_{i+1}; as a consequence it has the property $\mathcal{P}$. Therefore G/N is poly-$\mathcal{P}$. □

We say that the class of groups having $\mathcal{P}$ is *closed with respect to forming extensions* if a $\mathcal{P}$-by-$\mathcal{P}$ group has $\mathcal{P}$. The infinite dihedral group D_∞ from

Example 2.3.3, is abelian-by-abelian. Since D_∞ is neither abelian nor nilpotent, we see that the class of abelian groups and the class of nilpotent group are not closed with respect to forming extensions. By definition, the class of groups being poly-$\mathcal{P}$ is closed with respect to forming extensions. Thus we get the following corollary.

Corollary 2.4.2. *The class of polycyclic groups, respectively soluble groups, is closed with respect to forming subgroups, factor groups and extensions.*

Lemma 2.4.3. *The class of nilpotent groups is closed with respect to forming subgroups and factor groups.*

Proof. Let G be a nilpotent group and let $G = G_1 \geq G_2 \geq \cdots \geq G_n \geq G_{n+1} = 1$ be a central series of G.

For a subgroup $H \leq G$ we define define $H_i = H \cap G_i$. Since G_i/G_{i+1} is central in G/G_{i+1} it follows that H_i/H_{i+1} is central in H/H_{i+1}. Thus the groups $(H_i)_{1\leq i\leq n+1}$ form a central series of H and so H is nilpotent.

Similarly for a factor G/N we see that the groups $(G_iN/N)_{1\leq i\leq n+1}$ form a central series of G/N and so G/N is nilpotent. □

Let G be a polycyclic group. By definition G is soluble; further, by Corollary 2.4.2, every subgroup of G is polycyclic and thus finitely generated. The next corollary shows that the converse of this statement is also true; for its proof we need the following lemma.

Lemma 2.4.4. *An abelian group G is polycyclic if and only if it is finitely generated.*

Proof. Assume that G is generated by a finite set $\{g_1, \ldots, g_l\}$. Then the groups $(G_i := \langle g_i, \ldots, g_l\rangle)_{1\leq i\leq l+1}$ form a subnormal series of G with cyclic factors. Thus G is polycyclic. The other direction of the statement is, as mentioned before, a consequence of Corollary 2.4.2. □

Corollary 2.4.5. *A group G is polycyclic if and only if it is soluble and every subgroup of G is finitely generated.*

Proof. It remains to show the if-part of the statement. So assume that G is a soluble group such that all subgroups of G are finitely generated. Then G has a subnormal series $G = G_1 \rhd G_2 \rhd \cdots \rhd G_n \rhd G_{n+1} = 1$ with abelian factors. By assumption each G_i is finitely generated and thus G_i/G_{i+1} is finitely generated as well. Therefore, by Lemma 2.4.4, we deduce that G_i/G_{i+1} is polycyclic. Thus, by Corollary 2.4.2, G is polycyclic. □

From the definition of nilpotency we see that every nilpotent group is soluble. The next corollary explains the relationship between nilpotent and polycyclic groups. Recall that the tensor product $A \otimes B$ of two abelian groups A, B is defined as follows. Let $A \times B$ be the direct product of A and B and let $M \leq A \times B$ be the subgroup generated by the elements

$$(a_1 + a_2, b) - (a_1, b) - (a_2, b), (a, b_1 + b_2) - (a, b_1) - (a, b_2)$$

where $a, a_1, a_2 \in A, b, b_1, b_2 \in B$. Then $A \otimes B = (A \times B)/M$.

Lemma 2.4.6. *Let G be a group and denote $G_i = \gamma_i(G)$. For each $i > 1$, there is an epimorphism*

$$\psi_i : (G_{i-1}/G_i) \otimes (G_1/G_2) \to G_i/G_{i+1},$$

induced by the map

$$(gG_i, hG_2) \mapsto [g, h]G_{i+1}.$$

Proof. See [38, Chapter 1]. □

Corollary 2.4.7. *Let G be a nilpotent group. Then G is polycyclic if and only if G is finitely generated.*

Proof. We only have to show the if-part of the statement, so assume that G is a finitely generated nilpotent group. By Lemma 2.4.6 we see that $\gamma_i(G)/\gamma_{i+1}(G)$ is finitely generated for all i and therefore polycyclic. Since $\gamma_{c+1}(G) = 1$ for some $c \in \mathbb{N}$ this implies that G is polycyclic. □

By Corollary 2.4.7 every $\mathcal{T}$-group is polycyclic. The following lemma shows that every $\mathcal{T}$-group has a very special polycyclic sequence.

Lemma 2.4.8. *If G is a $\mathcal{T}$-group then G has a Mal'cev basis.*

Proof. By [38, Chapter 1] the factors of the upper central series of G are torsion-free. Let c be the nilpotency class of G and let $g_{i1}, \ldots, g_{ik_i}$ be a free generating set for $\zeta_i(G)/\zeta_{i-1}(G)$ for $i = 1, \ldots, c$. Then the list

$$(g_{c1}, \ldots, g_{ck_c}, \ldots, g_{11}, \ldots, g_{1k_1})$$

is a Mal'cev basis for G. □

2.5 Structure of infinite polycyclic groups

In this section we recall some well known structure theorems about infinite polycyclic groups. For proofs and further background see [38].

Definition 2.5.1. Let G be a group. The *Fitting subgroup* $\mathrm{Fitt}(G)$ is the subgroup which is generated by the set of all normal nilpotent subgroups of G.

Theorem 2.5.2. *Let G be a polycyclic-by-finite group. Then $\mathrm{Fitt}(G)$ is nilpotent.*

Theorem 2.5.3. *Let G be a polycyclic group. Then $G/\mathrm{Fitt}(G)$ is abelian-by-finite. In particular G is nilpotent-by-abelian-by-finite.*

The next theorem tells us that every polycyclic group is made out of two $\mathcal{T}$-groups in a rather easy way.

Theorem 2.5.4. *Let G be a polycyclic group. Then there exists a normal $\mathcal{T}$-subgroup N and a $\mathcal{T}$-subgroup C such that CN/N is free abelian and CN has finite index in G.*

Definition 2.5.5. The group C from Theorem 2.5.4 is said to be a *nilpotent almost-supplement* for N in G. 'Almost' because C and N generate a subgroup of finite index, and 'supplement' because C may intersect N non-trivially.

This structure of polycyclic groups can also be explored algorithmically. Let G be a polycyclic group given by a polycyclic presentation. In [17, Chapter 9] Eick describes a practical algorithm to compute a nilpotent-by-abelian-by-finite series $G \geq K \geq N \geq 1$ where K is torsion-free. The algorithm computes generators for K and N as words in the generators of G; it also computes polycyclic presentations for K and N on these generators. Further, Eick describes methods to determine a nilpotent almost-supplement C for N in G. Since $[G : CN] < \infty$ we can impose the additional condition that CN is normal in G by passing to the core $(CN)_G$ of CN in G. Note that N is contained in $(CN)_G$.

Chapter 3

Mal'cev correspondence

In this chapter we recall some well known facts about the connection between $\mathbb{Q}$-powered nilpotent groups and rational nilpotent Lie algebras, the Mal'cev correspondence, discovered by Anatoly Mal'cev in 1951 [27, 28]; this correspondence plays a key role in the further chapters of this book.

In §3.1 we define $\mathbb{Q}$-powered groups and give some basic facts about Lie algebras. In §3.2 we study an illustrative special case of the Mal'cev correspondence: the connection between upper unitriangular rational matrix groups and nilpotent rational matrix Lie algebras. Then in §3.3 we move on to the general case where the Mal'cev correspondence between abstract groups and Lie algebras is realized by means of the Baker–Campbell–Hausdorff formula.

For more background we refer to [9, Chapter 4], [23, Chapter 9,10] and [38, Chapter 6],

3.1 Preliminaries

$\mathbb{Q}$-powered groups

Definition 3.1.1. A group G is said to be $\mathbb{Q}$-*powered* if for every $g \in G$, $q \in \mathbb{N}$ there exists a unique $h \in G$ such that $h^q = g$. We denote this element h by $g^{\frac{1}{q}}$ and write $g^{\frac{p}{q}}$ for $(g^{\frac{1}{q}})^p$ where $p \in \mathbb{Z}, q \in \mathbb{N}$.

If G is $\mathbb{Q}$-powered then $g^q = 1$ implies $g = 1$; thus G is torsion-free.

Definition 3.1.2. We denote by $\mathrm{Tr}_1(d, \mathbb{Q})$ the group of all upper unitriangular matrices of degree n over $\mathbb{Q}$, i.e. matrices of the form

$$\begin{pmatrix} 1 & & * \\ & \ddots & \\ 0 & & 1 \end{pmatrix}.$$

Example 3.1.3. Let $g = \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \in \mathrm{Tr}_1(2, \mathbb{Q})$ and $n \in \mathbb{N}$. Then $h = \begin{pmatrix} 1 & x/n \\ 0 & 1 \end{pmatrix}$ is the unique element in $\mathrm{Tr}_1(2, \mathbb{Q})$ such that $h^n = g$. Thus $\mathrm{Tr}_1(2, \mathbb{Q})$ is a $\mathbb{Q}$-powered group. We will see in §3.2 that $\mathrm{Tr}_1(d, \mathbb{Q})$ is $\mathbb{Q}$-powered for all $d \in \mathbb{N}$.

Note that $\mathrm{Tr}_1(d, \mathbb{Q})$ is a nilpotent group [38, Chapter 1]. Furthermore every $\mathcal{T}$-group, i.e. a finitely generated torsion-free nilpotent group, can be embedded in some $\mathrm{Tr}_1(d, \mathbb{Q})$ [38, Chapter 5].

Lie algebras

Definition 3.1.4. Let L be a vector space over a field K. Then L is said to be a *Lie algebra* if there exists a bilinear map $L \times L \to L$, written $(x, y) \mapsto [x, y]$, with the properties

(1) $[x, x] = 0$,
(2) $[x, y] = -[y, x]$,
(3) $[[x, y], z] + [[y, z], x] + [[z, x], y] = 0$.

The expression $[x, y]$ is called a *Lie commutator* or a *Lie bracket* in x and y. Equation (3) is the *Jacobi identity*. A vector subspace $V \leq L$ is called a *Lie subalgebra* if it is closed under the Lie bracket, i.e. $[V, V] \leq V$.

For repeated Lie brackets we use the left norm convention, i.e.

$$[x_1, \ldots, x_r] = [[x_1, \ldots, x_{r-1}], x_r].$$

Example 3.1.5. Denote by $\mathrm{M}_{n\times n}(\mathbb{Q})$ the set of all rational matrices of degree n. Then $\mathrm{M}_{n\times n}(\mathbb{Q})$ is a Lie algebra with Lie bracket $[x, y] = xy - yx$.

Definition 3.1.6. Let L be Lie algebra. The *lower central series* $(L_k)_{k\in\mathbb{N}}$ of L is recursively defined by $L_1 = L$ and $L_{k+1} = [L_k, L]$. The Lie algebra L is said to be *nilpotent* if $L_c = 0$ for some $c \in \mathbb{N}$. A nilpotent Lie algebra L has *nilpotency class* c if $L_{c+1} = 0$ and $L_c \neq 0$.

Definition 3.1.7. Let L be Lie algebra. The center of L is defined by $Z(L) = \{l \in L | [l, L] = 0\}$. Note that $Z(L)$ is a vector subspace of L. The *upper central series* $(L^k)_{k\in\mathbb{N}}$ of L is recursively defined by $L^1 = Z(L)$ and $L^{k+1}/L^k = Z(L/L^k)$.

If L is a nilpotent Lie algebra of nilpotency class c then $L^c = L$.

Definition 3.1.8. We denote by $\mathrm{Tr}_0(d,\mathbb{Q}) \leq \mathrm{M}_{n\times n}(\mathbb{Q})$ the vector subspace of all upper triangular matrices with zeros on the diagonal. Since $\mathrm{Tr}_0(d,\mathbb{Q})$ is closed under the Lie bracket $[x,y] = xy - yx$ we see that $\mathrm{Tr}_0(d,\mathbb{Q})$ is a Lie subalgebra of $\mathrm{M}_{n\times n}(\mathbb{Q})$.

$\mathrm{Tr}_0(d,\mathbb{Q})$ is nilpotent because the kth term of the lower central series consists of matrices of the form

$$\begin{pmatrix} 0 & \dots & 0 & & * \\ & \ddots & & \ddots & \\ & & \ddots & & 0 \\ & & & \ddots & \vdots \\ 0 & & & & 0 \end{pmatrix}$$

where the $k-1$ subdiagonals above the diagonal are equal to 0.

Definition 3.1.9. Let L be a Lie algebra over a field K and let $\mathcal{B} = \{x_1,\dots,x_n\}$ be a basis of the underlying vector space. For each pair $x_i, x_j \in \mathcal{B}$ we can write

$$[x_i, x_j] = \sum_{k=1}^{n} c_{ij}^k x_k.$$

The n^3 elements $c_{ij}^k \in K$ are called the *structure constants* of L with respect to $\mathcal{B}$. Since the Lie bracket $[,]$ in L is bilinear, it is completely specified by the structure constants.

Finite dimensional Lie algebras are typically represented on a computer via matrices or via a structure constant table. In the first case the Lie algebra is given by a basis consisting of matrices $\{X_1,\dots,X_n\}$; in the second case the Lie algebra is given as an abstract vector space with basis $\mathcal{B} = \{x_1,\dots,x_n\}$ and the structures constants with respect to $\mathcal{B}$. For both representations, which can be obtained from each other, powerful algorithms for structural investigations are available [12]. Note that every finite dimensional Lie algebra has a faithful linear representation [21].

3.2 Matrix correspondence

Recall that we denote by $\mathrm{Tr}_1(d,\mathbb{Q})$ the group of all upper unitriangular $d\times d$-matrices over $\mathbb{Q}$, and by $\mathrm{Tr}_0(d,\mathbb{Q})$ the Lie algebra of all upper triangular $d\times d$-matrices over $\mathbb{Q}$ having zeros on the diagonal.

Definition 3.2.1. We define the maps log and the exp as follows:

$$\begin{aligned}
\log &: \mathrm{Tr}_1(d,\mathbb{Q}) \to \mathrm{Tr}_0(d,\mathbb{Q}) \\
&: g \mapsto (g-1) - \frac{1}{2}(g-1)^2 + \cdots + \frac{(-1)^d}{(d-1)}(g-1)^{d-1} \\
\exp &: \mathrm{Tr}_0(d,\mathbb{Q}) \to \mathrm{Tr}_1(d,\mathbb{Q}) \\
&: x \mapsto 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{(d-1)!}x^{d-1}
\end{aligned}$$

Note that this coincides with the usual definition of log and exp on the complex numbers via power series, since $(g-1)^d = x^d = 0$. The following lemma is a consequence of well-known identities of these power series.

Lemma 3.2.2. *The mappings* log *and* exp *from Definition 3.2.1 are mutually inverse bijections. For commuting matrices $x, y \in \mathrm{Tr}_0(d,\mathbb{Q})$, we have $(\exp x)(\exp y) = \exp(x+y)$.*

Remark 3.2.3. For $q \in \mathbb{N}$ and $g = \exp(x) \in \mathrm{Tr}_1(d,\mathbb{Q})$ we have that $h = \exp(\frac{1}{q}x)$ is a q-th root of g, i.e. $h^q = g$. Since $h^q = k^q$ implies $q\log(h) = q\log(k)$ and thus $h = k$, we deduce that h is a unique q-th root of g. Therefore $\mathrm{Tr}_1(d,\mathbb{Q})$ is a $\mathbb{Q}$-powered group. For $r \in \mathbb{Q}$ we have that

$$\begin{aligned}
\log(g^r) &= r\log(g) \\
\exp(rx) &= \exp(x)^r.
\end{aligned}$$

For non-commuting matrices $x, y \in \mathrm{Tr}_0(n,\mathbb{Q})$ it is not true in general that $\exp(x+y) = \exp(x)\exp(y)$. However a similar equality holds which contains additional error terms which depend on the extent to which x and y fail to commute. For a vector of positive integers $e = (e_1, \ldots, e_r)$ we define the repeated Lie bracket $[x,y]_e$ in $\mathrm{Tr}_0(d,\mathbb{Q})$ by

$$[x,y]_e = [x, \underbrace{y, \ldots, y}_{e_1}, \underbrace{x, \ldots, x}_{e_2}, \ldots].$$

Theorem 3.2.4. *There exist constants $q_e \in \mathbb{Q}$ not depending on d such that for all $x, y \in \mathrm{Tr}_0(d,\mathbb{Q})$ we have*

$$\exp(x)\exp(y) = \exp(x + y + \sum_e q_e [x,y]_e),$$

where we take the sum over all vectors $e = (e_1, \ldots, e_r)$, with positive integer entries, such that $e_1 + \cdots + e_r < d-1$. In particular this means that $\log((\exp x)(\exp y))$ is an element of the Lie subalgebra of $\mathrm{Tr}_0(d,\mathbb{Q})$ generated by x and y.

Proof. See [23, Section 9.2] and [38, Chapter 6]. □

Definition 3.2.5. Let the rational constants q_e be defined as in the last theorem. The formal expression $H(x,y) = x + y + \sum_e q_e[x,y]_e$, where the sum is taken over all vectors $(e_1, \ldots, e_r)$ with positive integer entries where $r \in \mathbb{N}$, is called the *Baker–Campbell–Hausdorff formula*.

Remark 3.2.6. If L is a nilpotent Lie algebra of nilpotency class c then any Lie bracket of length $c+1$ in $x, y \in L$ is trivial. Thus in L the Baker–Campbell–Hausdorff formula $H(x,y)$ has only finitely many non-zero terms.

Example 3.2.7. The terms up to length 3 of the Baker–Campbell–Hausdorff formula are

$$H(x,y) = x + y + \frac{1}{2}[x,y] + -\frac{1}{12}[x,y,y] + \frac{1}{12}[x,y,x] + \ldots$$

Theorem 3.2.8. *We can define a group multiplication* $*$ *on* $\mathrm{Tr}_0(d,\mathbb{Q})$ *given by* $x * y = H(x,y)$. *The exponential map is then an isomorphism of groups between* $(\mathrm{Tr}_0(d,\mathbb{Q}), *)$ *and* $\mathrm{Tr}_1(d,\mathbb{Q})$.

Proof. Let $x \in \mathrm{Tr}_0(d,\mathbb{Q})$. From the definition of $*$ we see that $0 * x = x * 0 = x$ and that $-x$ is an inverse of x. By Lemma 3.2.2 the function $\exp : \mathrm{Tr}_0(d,\mathbb{Q}) \to \mathrm{Tr}_1(d,\mathbb{Q})$ is a bijection. Further, by Theorem 3.2.4, we have that $\exp(x * y) = \exp(x)\exp(y)$ for $x, y \in \mathrm{Tr}_0(d,\mathbb{Q})$, which implies the associativity of $*$ and that exp is an isomorphism. □

The following theorem explains the interplay of subgroups of $\mathrm{Tr}_1(d,\mathbb{Q})$ and Lie subalgebras of $\mathrm{Tr}_0(d,\mathbb{Q})$ via log and exp.

Theorem 3.2.9. *Let* $G \leq \mathrm{Tr}_1(d,\mathbb{Q})$ *and let* $\mathbb{Q}\log(G)$ *be the* $\mathbb{Q}$*-vector space spanned by* $\log(G) = \{\log(g) | g \in G\}$. *Let* L *be a Lie subalgebra of* $\mathrm{Tr}_0(d,\mathbb{Q})$. *Then the following holds:*

- $\exp(L)$ *is a* $\mathbb{Q}$*-powered nilpotent subgroup of* $\mathrm{Tr}_1(d,\mathbb{Q})$.
- $\mathbb{Q}\log(G)$ *is a Lie subalgebra of* $\mathrm{Tr}_0(d,\mathbb{Q})$.
- $G \leq \exp(\mathbb{Q}\log(G))$ *and every element of* $\exp(\mathbb{Q}\log(G))$ *has some positive power lying in* G.

Proof. See [38, Chapter 6, Theorem 2]. □

Definition 3.2.10. Let H be a torsion-free nilpotent group. A $\mathbb{Q}$-powered group $\hat{H}$, containing H, is said to be a $\mathbb{Q}$*-powered hull* of H, if for every element $h \in \hat{H}$ there exists $z \in \mathbb{N}$ such that $h^z \in H$.

Theorem 3.2.9 shows that $\exp(\mathbb{Q}\log(G))$ is a $\mathbb{Q}$-powered hull of a subgroup $G \leq \mathrm{Tr}_1(d,\mathbb{Q})$; further if G is $\mathbb{Q}$-powered then $G = \exp(\mathbb{Q}\log(G))$.

We saw that the Baker–Campbell–Hausdorff formula allows us to define a group multiplication on $\mathrm{Tr}_0(d,\mathbb{Q})$ in terms of Lie algebra operations. Vice versa, it is possible to define operations of a Lie algebra on $\mathrm{Tr}_1(d,\mathbb{Q})$ in terms of the operations of a $\mathbb{Q}$-powered group.

For a vector of positive integers $e = (e_1,\ldots,e_r)$ we define the repeated group commutator $[g,h]_e$ in $\mathrm{Tr}_1(d,\mathbb{Q})$ by $[g,h]_e = [g,\underbrace{h,\ldots,h}_{e_1},\underbrace{g,\ldots,g}_{e_2},\ldots]$. The weight $w(e)$ of e is defined to be $\sum_{i=1}^{r} e_i$. Let $\leq$ be an order on a set of vectors with positive integer entries. We say that $\leq$ agrees with the increase of the weight of the vectors if $e \leq e'$ implies $w(e) \leq w(e')$.

Theorem 3.2.11. *There exist constants $r_e, s_e \in \mathbb{Q}$ not depending on d such that for all $g,h \in \mathrm{Tr}_1(d,\mathbb{Q})$*

$$\begin{aligned}\log(g)+\log(h) &= \log(gh\prod_e [g,h]_e{}^{r_e})\\ [\log(g),\log(h)] &= \log([g,h]\prod_e [g,h]_e{}^{s_e})\end{aligned}$$

where the product is taken over all vectors $e = (e_1,\ldots,e_r)$, with positive integer entries, such that $e_1+\cdots+e_r < d-1$ in some fixed order agreeing with the increase of the weight of the vectors e.

Proof. See [23, Section 10.1]. □

Definition 3.2.12. Let the rational constants r_e and s_e be defined as in the last theorem. The formal expressions

$$\begin{aligned}h_1(g,h) &= gh\prod_e [g,h]_e{}^{r_e}\\ h_2(g,h) &= [g,h]\prod_e [g,h]_e{}^{s_e},\end{aligned}$$

where the product is taken over all vectors $e = (e_1,\ldots,e_r)$ with positive integer entries where $r \in \mathbb{N}$ in some fixed order agreeing with the increase of the weight of the vectors e, are called the *inverse Baker–Campbell–Hausdorff formulae*.

Remark 3.2.13. If G is a $\mathbb{Q}$-powered nilpotent group of nilpotency class c then any group commutator of length $c+1$ in $g,h \in G$ is trivial. Thus in G the inverse Baker–Campbell–Hausdorff formulae $h_1(g,h)$ and $h_2(g,h)$ have only finitely many non-zero terms.

3.3 Abstract correspondence

We now describe the Mal'cev correspondence between abstract $\mathbb{Q}$-powered nilpotent groups and abstract rational nilpotent Lie algebras.

Let L be a nilpotent Lie algebra over $\mathbb{Q}$ and let $x, y \in L$. Using the Baker–Campbell–Hausdorff formula $H(x, y)$ from Definition 3.2.5 we can define the operations of a $\mathbb{Q}$-powered group on L by

$$\begin{aligned} x * y &= H(x, y) && (3.1)\\ x^q &= qx \text{ for } q \in \mathbb{Q}. && (3.2) \end{aligned}$$

Conversely, let G be a $\mathbb{Q}$-powered nilpotent group and let $g, h \in G$. Then using the inverse Baker–Campbell–Hausdorff formulae $h_1(g, h), h_2(g, h)$ from Definition 3.2.12 we can define the operations of a rational Lie algebra on G by

$$\begin{aligned} g + h &= h_1(g, h) && (3.3)\\ [g, h] &= h_2(g, h) && (3.4)\\ qg &= g^q \text{ for } q \in \mathbb{Q}. && (3.5) \end{aligned}$$

Theorem 3.3.1 (Mal'cev correspondence)**.** *For every $\mathbb{Q}$-powered nilpotent group G, the corresponding rational nilpotent Lie algebra L_G is defined on the same underlying set $L_G = G$, with Lie $\mathbb{Q}$-algebra operations (3.3), (3.4), (3.5). Conversely, for every rational nilpotent Lie algebra L, the corresponding $\mathbb{Q}$-powered nilpotent group G_L is defined on the same underlying set $G_L = L$, with group operations (3.1), (3.2) of a $\mathbb{Q}$-powered group. These transformations are inverses of one another: $L_{G_L} = L$ as rational Lie algebras (that is, not only sets, but all operations coincide), and, similarly, $G_{L_G} = G$ as ($\mathbb{Q}$-powered) groups.*

Proof. See [23, Theorem 10.11.]. □

Let G be a $\mathbb{Q}$-powered nilpotent group. To avoid confusion between G and L_G in the following, we will denote by $\mathrm{Log}(g)$ the element of L_G which corresponds to $g \in G$;

$$G \ni g \quad \leftrightarrow \quad \mathrm{Log}(g) \in L_G = \mathrm{Log}(G) = \{\mathrm{Log}(g) | g \in G\}.$$

For a rational nilpotent Lie algebra L we will denote by $\mathrm{Exp}(x)$ the element of G_L which corresponds to $x \in L$. Distinguishing G and L_G in this way, in the following it will be clear from the context if we mean by $[,]$ the Lie

commutator or the group commutator. Log can be regarded as a mapping between G and $\mathrm{Log}(G)$. For $g, h \in G$ and $q \in \mathbb{Q}$ we have

$$\begin{aligned} \mathrm{Log}(gh) &= \mathrm{Log}(g) * \mathrm{Log}(h) \\ \mathrm{Log}(h_1(g,h)) &= \mathrm{Log}(g) + \mathrm{Log}(h) \\ \mathrm{Log}(h_2(g,h)) &= [\mathrm{Log}(g), \mathrm{Log}(h)] \\ \mathrm{Log}(g^q) &= q\,\mathrm{Log}(g). \end{aligned}$$

Similarly Exp can be regarded as a mapping between L and $\mathrm{Exp}(L)$.

Remark 3.3.2. Note that in the context of the matrix Mal'cev correspondence as discussed in §3.2 we use log to denote the function

$$\begin{aligned} \log &: \mathrm{Tr}_1(d, \mathbb{Q}) \to \mathrm{Tr}_0(d, \mathbb{Q}) \\ &: g \mapsto (g-1) - \frac{1}{2}(g-1)^2 + \cdots + \frac{(-1)^d}{(d-1)}(g-1)^{d-1} \end{aligned}$$

which links the $\mathbb{Q}$-powered nilpotent matrix group $\mathrm{Tr}_1(d, \mathbb{Q})$ and the nilpotent Lie algebra $\mathrm{Tr}_0(d, \mathbb{Q})$. In opposition to this we use in the context of the abstract Mal'cev correspondence the function name Log to link an abstract nilpotent $\mathbb{Q}$-powered group G and the corresponding Lie Algebra L_G.

Let H be a torsion-free nilpotent group. Recall that a $\mathbb{Q}$-powered group $\hat{H}$, containing H, is said to be a $\mathbb{Q}$-powered hull of H, if for every element $h \in \hat{H}$ there exists $z \in \mathbb{N}$ such that $h^z \in H$.

Theorem 3.3.3. *Let H be a torsion-free nilpotent group. Then H has a $\mathbb{Q}$-powered hull $\hat{H}$ of the same nilpotency class. If $\hat{H}_1, \hat{H}_2$ are $\mathbb{Q}$-powered hulls of H then the identity map in H extends to a unique isomorphism of $\hat{H}_1$ onto $\hat{H}_2$. Every automorphism of H extends to an automorphism of $\hat{H}$. If $H \leq G$ and G is $\mathbb{Q}$-powered, then G contains a $\mathbb{Q}$-powered hull of H.*

Proof. See [23, Corollary 9.19. and Theorem 9.20.] □

Given the fact that all $\mathbb{Q}$-powered hulls of a torsion-free nilpotent group H are isomorphic, we will identify them in the following and speak of the $\mathbb{Q}$-powered hull $\hat{H}$ of H. The Lie algebra $L_{\hat{H}}$ corresponding to $\hat{H}$ will be denoted by $\mathcal{L}(H)$. Note that $\mathcal{L}(H)$ is spanned by $\mathrm{Log}(H) \subseteq \mathcal{L}(H)$ over the rationals, because every element $h \in \hat{H}$ has some power h^z lying in H; thus $\mathrm{Log}(h) = \frac{1}{z}\mathrm{Log}(h^z) \in \mathbb{Q}\ \mathrm{Log}(H)$. Further we have that $\hat{H}$ and $(\mathcal{L}(H), *)$ are naturally isomorphic as groups.

Remark 3.3.4. Let N be a $\mathcal{T}$-group and $\beta : N \to \mathrm{Tr}_1(d, \mathbb{Q})$ a faithful matrix representation. By §3.2, $L = \mathbb{Q}\log(N\beta)$ is a Lie algebra and $\exp(L) \cong (L, *)$ is a $\mathbb{Q}$-powered hull of N. By Theorem 3.3.3, $(L, *)$ and $(\mathcal{L}(N), *)$ are isomorphic as $\mathbb{Q}$-powered nilpotent groups and thus, by the Mal'cev correspondence, L and $\mathcal{L}(N)$ are isomorphic as rational Lie algebras.

In Chapters 5, 6 and 7 we will use the Mal'cev correspondence for computations with automorphisms of a $\mathcal{T}$-group H. The next theorem shows that the automorphisms of $\hat{H}$ and $\mathcal{L}(H)$ are in one-one correspondence; it is a direct consequence of the fact that the Lie algebra operations in $\mathcal{L}(H)$ can be defined in terms of the group operations of $\hat{H}$ and vice versa.

Theorem 3.3.5. *Let G be a $\mathbb{Q}$-powered nilpotent group and L its corresponding Lie algebra. Then the map $\tilde{}\,: \mathrm{Aut}(G) \to \mathrm{Aut}(L)$, defined by $\varphi \mapsto \mathrm{Exp} \circ \varphi \circ \mathrm{Log}$ is an isomorphism.*

Proof. Let $\varphi \in \mathrm{Aut}(G)$ and $g, h \in G$. Then

$$\begin{aligned}(\mathrm{Log}(g) + \mathrm{Log}(h))^{\tilde{\varphi}} &= \mathrm{Log}(h_1(g,h))^{\tilde{\varphi}} \\ &= \mathrm{Log}(h_1(g,h)^{\varphi}) \\ &= \mathrm{Log}(h_1(g^{\varphi}, h^{\varphi})) \\ &= \mathrm{Log}(g^{\varphi}) + \mathrm{Log}(h^{\varphi}) \\ &= \mathrm{Log}(g)^{\tilde{\varphi}} + \mathrm{Log}(h)^{\tilde{\varphi}}.\end{aligned}$$

With a similar argument we see that $[\mathrm{Log}(g), \mathrm{Log}(h)]^{\tilde{\varphi}} = [\mathrm{Log}(g)^{\tilde{\varphi}}, \mathrm{Log}(h)^{\tilde{\varphi}}]$ and that for $q \in \mathbb{Q}$ we have $(q\,\mathrm{Log}(g))^{\tilde{\varphi}} = q(\mathrm{Log}(g)^{\tilde{\varphi}}$. Thus $\tilde{\varphi} \in \mathrm{Aut}(L)$ because by the properties of the Mal'cev correspondence for every $x, y \in L$ there exist $g, h \in G$ such that $\mathrm{Log}(g) = x$ and $\mathrm{Log}(h) = y$.

Similarly we see that for $\tau \in \mathrm{Aut}(L)$, the mapping $\mathrm{Log} \circ \tau \circ \mathrm{Exp}$ is in $\mathrm{Aut}(G)$. Therefore, since $\varphi \mapsto \mathrm{Exp} \circ \varphi \circ \mathrm{Log}$ and $\tau \mapsto \mathrm{Log} \circ \tau \circ \mathrm{Exp}$ are inverses of each other, $\tilde{}$ is a bijection between $\mathrm{Aut}(G)$ and $\mathrm{Aut}(L)$. Further for $\varphi, \psi \in \mathrm{Aut}(G)$ we have that $\widetilde{\varphi\psi} = \tilde{\varphi}\tilde{\psi}$ and thus $\tilde{}$ is an isomorphism. □

The next theorem explains how $\mathrm{Aut}(H)$ fits into the relationship between the automorphism group of $\hat{H}$ and $\mathcal{L}(H)$.

Theorem 3.3.6. *Let H be a torsion-free nilpotent group and $\hat{H}$ its $\mathbb{Q}$-powered hull. Let Γ be the setwise stabilizer in $\mathrm{Aut}(\mathcal{L}(H))$ of the set $\mathrm{Log}(H)$. Then $\varphi \in \mathrm{Aut}(\hat{H})$ induces an automorphism of H, i.e. $H^{\varphi} = H$, if and only if $\tilde{\varphi} \in \Gamma$.*

Proof. Let $\varphi \in \mathrm{Aut}(\hat{H})$ and assume that $H^{\varphi} = H$. Then we have $\mathrm{Log}(H)^{\tilde{\varphi}} = \mathrm{Log}(H^{\varphi}) = \mathrm{Log}(H)$. Thus $\tilde{\varphi} \in \Gamma$. Conversely assume that $\tilde{\varphi} \in \Gamma$. Then $H^{\varphi} = \mathrm{Exp}(\mathrm{Log}(H)^{\tilde{\varphi}}) = \mathrm{Exp}(\mathrm{Log}(H))$ and thus $H^{\varphi} = H$. □

Lemma 3.3.7. *Denote by $x_1, \dots, x_s$ abstract Lie elements. Let $\mathcal{K}$ be the set of all repeated Lie brackets $\kappa(x_1, \dots, x_s)$ of length at least $s+1$ in the arguments $x_1, \dots, x_s$ each of which appears at least once. Then there exist rational constants $(t_\kappa)_{\kappa \in \mathcal{K}}$ such that for any $\mathbb{Q}$-powered nilpotent group G and $g_1, \dots, g_s \in G$ it holds that*

$$[\operatorname{Log}(g_1), \dots, \operatorname{Log}(g_s)] = \operatorname{Log}([g_1, \dots, g_s]) + \sum_{\kappa \in \mathcal{K}} t_\kappa \kappa(\operatorname{Log}(g_1), \dots, \operatorname{Log}(g_s)). \quad (3.6)$$

Proof. See [38, Chapter 6, Corollary 2]. □

Remark 3.3.8. The sum in equation (3.6) is by definition infinite. However for a given $\mathbb{Q}$-powered nilpotent group G only a finite number of the Lie brackets κ are nonzero.

Lemma 3.3.9. *Denote by $x_1, \dots, x_s$ abstract group elements. Let $\mathcal{L}$ be the set of all repeated group commutators $\lambda(x_1, \dots, x_s)$ of length at least $s+1$ in the arguments $x_1, \dots, x_s$ each of which appears at least once. Then there exist rational constants $(u_\lambda)_{\lambda \in \mathcal{L}}$ such that for any $\mathbb{Q}$-powered nilpotent group G and $g_1, \dots, g_s \in G$ it holds that*

$$[\operatorname{Log}(g_1), \dots, \operatorname{Log}(g_s)] = \operatorname{Log}([g_1, \dots, g_s]) + \sum_{\lambda \in \mathcal{L}} u_\lambda \operatorname{Log}(\lambda(g_1, \dots, g_s)). \quad (3.7)$$

Proof. By Lemma 3.3.7 we have

$$[\operatorname{Log}(g_1), \dots, \operatorname{Log}(g_s)] = \operatorname{Log}([g_1, \dots, g_s]) + \sum_{\kappa \in \mathcal{K}} t_\kappa \kappa(\operatorname{Log}(g_1), \dots, \operatorname{Log}(g_s)). \quad (3.8)$$

Now we replace all occurrences of Lie brackets of length $\geq s+1$ in (3.8) with the corresponding right hand side of (3.6) and continue until no Lie bracket of length $\geq s+1$ appears in the equation.

This process must stop, because G and $\mathcal{L}(G)$ have finite nilpotency class c and thus group commutators and Lie brackets of lengths greater then c are trivial, and because a replacement of a Lie bracket of length k only introduces Lie brackets of length $\geq k+1$.

The only influence of the group G on this process is its nilpotency class c, which tells us that group commutators and Lie brackets of length $c+1$ can be disregarded. However this does not affect the values of those u_λ whose corresponding λ is of length $\leq c$. Thus the constants u_λ are not depending on G, and in particular not on c. □

Remark 3.3.10. The sum in equation (3.7) is by definition infinite. However for a given $\mathbb{Q}$-powered nilpotent group G only a finite number of the group commutators λ are nontrivial.

Remark 3.3.11. [38, Chapter 6, Corollary 3] claims that the constants u_λ in the last lemma depend on c.

Lemma 3.3.12. *Let $(g_1, \dots, g_l)$ be a Mal'cev basis for a $\mathcal{T}$-group H. Then $\mathcal{B} = \{\mathrm{Log}(g_1), \dots, \mathrm{Log}(g_l)\}$ is a basis for the Lie algebra $\mathcal{L}(H)$. In particular, the dimension of $\mathcal{L}(H)$ is equal to the Hirsch length of H.*

Proof. Let $g = g_1^{a_1} \cdots g_l^{a_l} \in H$. Then $\mathrm{Log}(g) = a_1\mathrm{Log}(g_1) * \cdots * a_l\mathrm{Log}(g_l)$. Thus $\mathcal{L}(H)$ is generated by $\mathcal{B}$ as a Lie algebra, i.e. the smallest $\mathbb{Q}$-vector space that contains $\mathcal{B}$ and is closed under taking Lie brackets is equal to $\mathcal{L}(H)$.

We show via induction over l, the Hirsch length of H, that $\mathcal{B}$ is a basis of $\mathcal{L}(H)$.

If $H = \langle g_1 \rangle$ then $\{\mathrm{Log}(g_1)\}$ is a basis for $\mathcal{L}(H)$. Assume that the lemma is true for all $\mathcal{T}$-groups of Hirsch length $l-1$. First we show that the $\mathbb{Q}$-span $\langle \mathcal{B} \rangle_{\mathbb{Q}}$ of $\mathcal{B}$ is equal to $\mathcal{L}(H)$. By assumption the vector spaces $\langle \mathrm{Log}(g_2), \dots, \mathrm{Log}(g_l) \rangle_{\mathbb{Q}}$ and $\langle \mathrm{Log}(g_1), \mathrm{Log}(g_3), \dots, \mathrm{Log}(g_l) \rangle_{\mathbb{Q}}$ are closed under taking Lie brackets. Thus we have to show that $[\mathrm{Log}(g_1), \mathrm{Log}(g_2)] \in \langle \mathcal{B} \rangle_{\mathbb{Q}}$. By Lemma 3.3.9 $[\mathrm{Log}(g_1), \mathrm{Log}(g_2)] = \mathrm{Log}([g_1, g_2]) + \sum u_\lambda \mathrm{Log}(\lambda(g_1, g_2))$, where $u_\lambda \in \mathbb{Q}$ and λ is a repeated group theoretic commutator in g_1, g_2 of length ≥ 3. Since $[g_1, g_2], \lambda(g_1, g_2) \in \langle g_3, \dots, g_l \rangle$ the right hand side of the last equation is contained in the Lie algebra $\mathcal{L}(\langle g_3, \dots, g_l \rangle)$ and thus in the $\mathbb{Q}$-vector space spanned by $\mathcal{B}$.

It remains to show that the elements of $\mathcal{B}$ are linearly independent. By the induction hypothesis $\mathrm{Log}(g_2), \dots, \mathrm{Log}(g_l)$ are linearly independent. So assume that $\mathrm{Log}(g_1) \in \langle \mathrm{Log}(g_2), \dots, \mathrm{Log}(g_l) \rangle_{\mathbb{Q}} = L_2$. Therefore $g_1 \in \mathrm{Exp}(L_2)$ which is equal to the $\mathbb{Q}$-powered hull of $\langle g_2, \dots, g_l \rangle$. Thus there must be an $m \in \mathbb{N}$ such that $g_1^m \in \langle g_2, \dots, g_l \rangle$. Since $(g_1, \dots, g_l)$ is a Mal'cev basis this is a contradiction. □

Chapter 4

Computing the correspondence

In this chapter we show how the Mal'cev correspondence between the radicable hull of a $\mathcal{T}$-group G and the Lie algebra $\mathcal{L}(G)$ can be set up on a computer. We assume that G is given by a polycyclic presentation with respect to a Mal'cev basis $\mathcal{G} = (g_1, \ldots, g_l)$. Note that $\mathcal{B} = \{\mathrm{Log}(g_1), \ldots, \mathrm{Log}(g_l)\}$ is a basis of $\mathcal{L}(G)$. We show how to solve the following three tasks:

- **Lie algebra presentation**: Determine a computer presentation of $\mathcal{L}(G)$.
- **Logarithm**: Given an element $g = g_1^{e_1} \cdots g_l^{e_l} \in \hat{G}$, compute the coefficient vector $(\alpha_1, \ldots, \alpha_l)$ such that $\mathrm{Log}(g) = \sum_{i=1}^{l} \alpha_i \mathrm{Log}(g_i)$.
- **Exponential**: Given an element $x = \sum_{i=1}^{l} \alpha_i \mathrm{Log}(g_i) \in \mathcal{L}(G)$, compute the exponent vector $(e_1, \ldots, e_l)$ such that $\mathrm{Exp}(x) = g_1^{e_1} \cdots g_l^{e_l}$.

We present two approaches for solving these tasks. The first uses the fact that every $\mathcal{T}$-group can be embedded in an upper unitriangular matrix group and is discussed in §4.1. The second makes use of the Baker–Campbell–Hausdorff formula and related identities, see §4.2. Further we present in §4.3 a symbolic approach for computing logarithms and exponentials. Finally, we compare these methods and report on their implementations in §4.4.

4.1 Via matrix embeddings

This method uses a vector subspace of $\mathrm{Tr}_0(d, \mathbb{Q})$ to represent $\mathcal{L}(G)$ on a computer.

Every $\mathcal{T}$-group G has a faithful matrix representation $\beta : G \to \mathrm{Tr}_1(d, \mathbb{Q})$ for some $d \in \mathbb{N}$ [38, Chapter 3]. Recall that by Remark 3.3.4 the Lie algebra $\mathbb{Q}\log(G\beta)$, i.e. the $\mathbb{Q}$-vector space spanned by $\log(G\beta)$, is isomorphic

to $\mathcal{L}(G\beta)$ and thus to $\mathcal{L}(G)$. For a $\mathcal{T}$-group G, given by a polycyclic presentation, it is possible to compute $\beta : G \to \mathrm{Tr}_1(n, \mathbb{Q})$ [13, 26, 33]. An implementation of [13] is publicly available as part of GAP [41].

In order to be able to go back and forth between G and $G\beta$, it is necessary to compute a special kind of polycyclic sequence for $G\beta$.

Definition 4.1.1. A polycyclic sequence $\mathcal{M} = (M_1, \ldots, M_l)$ for a finitely generated group $H \leq \mathrm{Tr}_1(n, \mathbb{Q})$ is called *constructive* if there exists a practical algorithm which, given any $h \in H$, determines the normal form $\mathrm{nf}(h)$ with respect to $\mathcal{M}$.

It is well-known how to compute a constructive polycyclic sequence for a given finitely generated group $H \leq \mathrm{Tr}_1(n, \mathbb{Q})$ which is also a Mal'cev basis of H, see for example [39, Chapter 9]. By changing the underlying generating set of the polycyclic presentation of G, we can assume in the following that $\mathcal{G} = (g_1, \ldots, g_l)$ is a polycyclic sequence of G such that $(g_1\beta, \ldots, g_l\beta)$ is constructive polycyclic sequence and Mal'cev basis for $G\beta$. As a basis for $\mathcal{L}(G) \cong \mathbb{Q}\log(G\beta)$ we use the set of matrices $(\log(g_1\beta), \ldots, \log(g_l\beta))$.

The task Logarithm can be solved as follows. Given $g = g_1^{e_1} \cdots g_l^{e_l} \in \hat{G}$, we compute $M = (g_1\beta)^{e_1} \cdots (g_l\beta)^{e_l}$ and then $\log(M)$. Finally, by solving linear equations, we determine $(\alpha_1, \ldots, \alpha_l)$ such that $\sum \alpha_i \log(g_i\beta) = \log(M)$.

For the task Exponential we do the following. Given $x = \sum_{i=1}^{l} \alpha_i \log(g_i\beta)$, we compute $\exp(x)$. Then we can use the constructive polycyclic sequence $(g_1\beta, \ldots, g_l\beta)$ of $G\beta$ to compute the exponent vector $(e_1, \ldots, e_l)$ of $\exp(x)$ as described in [39, Chapter 9].

Example 4.1.2. Let $G = F_{2,2}$ be the free nilpotent of class two group on two generators g_1, g_2. Then the derived group is cyclic and generated by $g_3 = [g_2, g_1]$. It follows that

$$G = \langle g_1, g_2, g_3 | g_2^{(g_1^{\pm 1})} = g_2 g_3^{\pm 1} \rangle$$

is a polycyclic presentation for G and $\mathcal{G} = (g_1, g_2, g_3)$ is a Mal'cev basis. The embedding $\beta \to \mathrm{Tr}_1(3, \mathbb{Q})$, as computed by the algorithm in [13], is given by

$$g_1\beta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, g_2\beta = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, g_3\beta = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In this example $(g_1\beta, g_2\beta, g_3\beta)$ is a constructive polycyclic sequence for $G\beta$ and therefore we do not have to change the underlying generating set of G

and set $\mathcal{M} = (g_1\beta, g_2\beta, g_3\beta)$. The corresponding basis of $\mathbb{Q}\log(G\beta) \cong \mathcal{L}(G)$ consists of

$$\log(M_1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \log(M_2) = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \log(M_3) = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Let $x = \log(M_1) + \log(M_2) + \log(M_3) \in \mathcal{L}(G)$ and assume that we want to compute the exponent vector of $\mathrm{Exp}(x)$, i.e. the vector (e_1, e_2, e_3) such that

$$(g_1^{e_1} g_2^{e_2} g_3^{e_3})\beta = \exp(x).$$

First we compute

$$\exp(x) = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} = \begin{pmatrix} 1 & -1 & -3/2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Now we can read off from the first subdiagonal that $e_1 = 1$ and $e_2 = 1$. Next we divide off and compute

$$(g_3\beta)^{e_3} = ((g_1\beta)^{e_1}(g_2\beta)^{e_2})^{-1} \exp(x) = \begin{pmatrix} 1 & 0 & -3/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

This implies $e_3 = 3/2$.

4.2 Via the Baker–Campbell–Hausdorff formula

This method uses an abstract vector space and a structure constant table with respect to the basis $\mathcal{B}$ to present $\mathcal{L}(G)$ on a computer.

For the computation of the structure constants, we use equation (3.7) from Lemma 3.3.9; it expresses $[\mathrm{Log}(g), \mathrm{Log}(h)]$, where $g, h \in G$, in terms of a linear combination of logarithms of group commutators.

The proof of Lemma 3.3.9 is constructive and therefore can be used to compute the terms of (3.7); it makes use of the equation (3.6) from Lemma 3.3.7, whose terms can be determined using the method in [36, §8] and the Dynkin bracket operator defined in [44, Chapter 2].

Example 4.2.1. With $[\mathrm{Log}(g), \mathrm{Log}(h)]$ as left hand side, equation (3.6) is

$$\begin{aligned} [\mathrm{Log}(g), \mathrm{Log}(h)] &= \mathrm{Log}([g,h]) + \frac{1}{2}[\mathrm{Log}(h), \mathrm{Log}(g), \mathrm{Log}(h)] \\ &\quad + \frac{1}{2}[\mathrm{Log}(h), \mathrm{Log}(g), \mathrm{Log}(g)] + \ldots \end{aligned} \tag{4.1}$$

where only terms up to length 3 are displayed. Now we replace all Lie brackets of length ≥ 3 by the according right hand side of (3.6) and get

$$[\mathrm{Log}(g), \mathrm{Log}(h)] = \mathrm{Log}([g,h]) + \frac{1}{2}\mathrm{Log}([h,g,h]) + \frac{1}{2}\mathrm{Log}([h,g,g]) + \dots$$

This is equation (3.7) with $[\mathrm{Log}(g), \mathrm{Log}(h)]$ as left hand side, where only terms up to length 3 are displayed.

Note that the terms of equation (3.6) and equation (3.7) do not depend on G. Therefore they can be precomputed up to a given length. The number of terms grows exponentially in the length.

Let $\mathcal{G} = (g_1, \dots, g_l)$ be a Mal'cev basis of G and denote by $\mathcal{B}$ the corresponding basis $\{\mathrm{Log}(g_1), \dots, \mathrm{Log}(g_l)\}$ of $\mathcal{L}(G)$. We show by induction on l that it is possible to compute the structure constant table of the Lie algebra $\mathcal{L}(G)$ with respect to $\mathcal{B}$.

If $l = 1$, then $\mathcal{L}(G)$ is abelian and so the structure constant table of $\mathcal{B}$ is known. If $l > 1$, then we can assume by induction that the structure constant table of $\{\mathrm{Log}(g_2), \dots, \mathrm{Log}(g_l)\}$ is already known and that we can compute $\mathrm{Log}(g)$ for g in the $\mathbb{Q}$-powered hull of $\langle g_2, \dots, g_l\rangle$. Then using (3.7), we can express $[\mathrm{Log}(g_1), \mathrm{Log}(g_i)]$ as a linear combination of logarithms of repeated group commutators $\kappa(g_1, g_i)$. Since $\kappa(g_1, g_i)$ is in $\langle g_2, \dots, g_l\rangle$ for any commutator κ, we can compute $\mathrm{Log}(\kappa(g_1, g_i))$ and therefore determine the coefficients of $[\mathrm{Log}(g_1), \mathrm{Log}(g_i)]$ with respect to $\mathcal{B}$. Thus we can compute the structure constant table of $\mathcal{B}$.

For the computation of Logarithms we use the fact that $\mathrm{Log}(gh) = \mathrm{Log}(g) * \mathrm{Log}(h)$. Thus for given $g = g_1^{e_1} \cdots g_l^{e_l} \in \hat{G}$ we have that $\mathrm{Log}(g) = (e_1 \mathrm{Log}(g_1)) * \cdots * (e_l \mathrm{Log}(g_l))$, and therefore the coefficients of $\mathrm{Log}(g) = \sum \alpha_i \mathrm{Log}(g_i)$ can be be computed by using the Baker–Campbell–Hausdorff formula and the structure constant table of $\mathcal{B}$.

It remains to solve the task Exponential. For a given element $x = \sum_{i=1}^{l} \alpha_i \mathrm{Log}(g_i) \in \mathcal{L}(G)$ we have that $x = (\alpha_1 \mathrm{Log}(g_1)) * (-\alpha_1 \mathrm{Log}(g_1)) * x$ and $y = (-\alpha_1 \mathrm{Log}(g_1)) * x \in \langle \mathrm{Log}(g_2), \dots, \mathrm{Log}(g_l)\rangle_{\mathbb{Q}}$. Thus $e_1 = \alpha_1$. Using the structure constant table of $\mathcal{B}$ and the BCH-formula we can compute $y := (-\alpha_1 \mathrm{Log}(g_1)) * x$. By induction on l, we can assume that we can determine $f_2, \dots, f_l$ such that $g_2^{f_2} \cdots g_l^{f_l} = \mathrm{Exp}(y)$. Since $\mathrm{Exp}(x) = g_1^{\alpha_1} \mathrm{Exp}(y)$ we deduce that $(e_1, \dots, e_l) = (\alpha_1, f_2, \dots, f_l)$.

Example 4.2.2. Let $G = F_{2,2}$ be given as in Example 4.1.2. We have that $[\mathrm{Log}(e), \mathrm{Log}(f)] = \mathrm{Log}([e,f])$ for $e, f \in G$. Therefore $[\mathrm{Log}(g_2), \mathrm{Log}(g_1)] = \mathrm{Log}(g_3)$ and $[\mathrm{Log}(g_3), \mathrm{Log}(g_1)] = [\mathrm{Log}(g_3), \mathrm{Log}(g_2)] = 0$.

Let $g = g_1 g_2^5$ and suppose that we want to compute the coefficients of $\mathrm{Log}(g)$. We have that $\mathrm{Log}(g) = \mathrm{Log}(g_1) * \mathrm{Log}(g_2^5) = \mathrm{Log}(g_1) + 5\,\mathrm{Log}(g_2) + \frac{1}{2}[\mathrm{Log}(g_1), 5\,\mathrm{Log}(g_2)]$. Thus $\mathrm{Log}(g) = \mathrm{Log}(g_1) + 5\,\mathrm{Log}(g_2) - \frac{5}{2}\mathrm{Log}(g_3)$.

4.3 Symbolic Log and Exp

It is well-known that, in the context of $\mathcal{T}$-groups, Log and Exp can be described by polynomial functions [22, Chapter 6]. In this section we show how to compute these functions and apply them for the computations of logarithms and exponentials.

Lemma 4.3.1. *Let G be a $\mathcal{T}$-group with Mal'cev basis $\mathcal{G} = (g_1, \ldots, g_l)$.*

(i) Define l functions $\overline{\alpha}_1, \ldots, \overline{\alpha}_l$ in l rational variables $e_1, \ldots, e_l$ such that

$$\sum_{i=1}^{l} \overline{\alpha}_i \operatorname{Log}(g_i) = \operatorname{Log}(g_1^{e_1} \cdots g_l^{e_l}).$$

Then $\overline{\alpha}_i$ is a polynomial in $e_1, \ldots, e_l$ for $i = 1, \ldots, l$.

(ii) Define l functions $\overline{e}_1, \ldots, \overline{e}_l$ in l rational variables $\alpha_1, \ldots, \alpha_l$ such that

$$g_1^{\overline{e}_1} \cdots g_l^{\overline{e}_l} = \operatorname{Exp}\left(\sum_{i=1}^{l} \alpha_i \operatorname{Log}(g_i)\right).$$

Then $\overline{e}_i$ is a polynomial in $\alpha_1, \ldots, \alpha_l$ for $i = 1, \ldots, l$.

Proof. (i): Let $x = \sum_{i=1}^{l} r_i \operatorname{Log}(g_i)$, $y = \sum_{i=1}^{l} s_i \operatorname{Log}(g_i) \in \mathcal{L}(G)$. By the properties of the Baker–Campbell–Hausdorff formula the coefficients of $x * y$ with respect to the basis $\{\operatorname{Log}(g_1), \ldots, \operatorname{Log}(g_l)\}$ are polynomials in $r_1, \ldots, r_l, s_1, \ldots, s_l$.

For $g = g_1^{e_1} \cdots g_l^{e_l}$ we have that $\operatorname{Log}(g) = (e_1 \operatorname{Log}(g_1)) * \cdots * (e_l \operatorname{Log}(g_l))$. By the repeated application of the argument from above we see that $\overline{\alpha}_i$ is a polynomial in $e_1, \ldots, e_l$ for $i = 1, \ldots, l$.

(ii): Let $x = \sum_{i=1}^{l} \alpha_i \operatorname{Log}(g_i) \in \mathcal{L}(G)$. If $l = 1$, then $\overline{e}_1(\alpha_1) = \alpha_1$ and thus $\overline{e}_1$ is a polynomial. Now assume that $l > 1$. In §4.2 we saw that $(\overline{e}_1, \ldots \overline{e}_l) = (\alpha_1, f_2, \ldots, f_l)$ where $g_2^{f_2} \cdots g_l^{f_l} = \operatorname{Exp}(y)$ with $y = (-\alpha_1 \operatorname{Log}(g_1)) * x$. Let $y = \sum_{i=2}^{l} \beta_i \operatorname{Log}(g_i)$. By the properties of the Baker–Campbell–Hausdorff formula β_i is a polynomial in $\alpha_1, \ldots, \alpha_l$. Further by induction we can assume that $f_2, \ldots, f_l$ are polynomials in $\beta_2, \ldots, \beta_l$. Thus $\overline{e}_i$ is a polynomial in $\alpha_1, \ldots, \alpha_l$ for $i = 1, \ldots, l$. □

Example 4.3.2. Let $G = F_{2,2}$ be the group already studied in Example 4.1.2 and 4.2.2. We have that

$$\operatorname{Log}(g_1^{e_1} g_2^{e_2} g_3^{e_3}) = (e_1 \operatorname{Log}(g_1)) * (e_2 \operatorname{Log}(g_2)) * (e_3 \operatorname{Log}(g_3))$$

which is equal to $e_1 \operatorname{Log}(g_1) + e_2 \operatorname{Log}(g_2) + e_3 \operatorname{Log}(g_3) + \frac{1}{2}[e_1 \operatorname{Log}(g_1), e_2 \operatorname{Log}(g_2)]$. Therefore we have $\overline{\alpha}_1 = e_1$, $\overline{\alpha}_2 = e_2$ and $\overline{\alpha}_3 = -\frac{1}{2} e_1 e_2 + e_3$.

The proof of the Lemma 4.3.1 is constructive and can be used to compute the polynomials $\overline{\alpha}_i$ and $\overline{e}_j$ if the structure constant table of the Lie algebra $\mathcal{L}(G)$ is known. See §4.4 for comments on the implementation and runtimes.

The functions $\overline{\alpha}_i$, $\overline{e}_j$ can be applied for the computation of logarithms and exponentials. In §4.4 we will see that this is yields a considerable speed up in comparison with the methods described in §4.1 and 4.2.

4.4 Runtimes and comparison

The approaches described in §4.1, §4.2 and §4.3 to realizing the Mal'cev correspondence have been implemented in GAP [41] as a part of the package Guarana [2]. In this section we make comments on their implementation, indicate runtimes and compare them.

Implementation

For the method of §4.1, we used the algorithm and implementation of Nickel [33] to compute the faithful matrix representations of the given $\mathcal{T}$-group G. It is much more efficient than previous methods [13, 26].

For the method of §4.2, we use a *weight function* that can be associated to every Mal'cev basis $\mathcal{G} = (g_1, \ldots, g_l)$; this is a function $w : \mathcal{G} \to \mathbb{N}\backslash\{0\}$ such that for all g_k showing up in the normal form of $[g_i, g_j]$ we have that $w(g_k) \geq w(g_i)+w(g_j)$. If $\bar{w} = \max w(\mathcal{G})$ then $[g_i, g_j] = 1$ if $w(g_i)+w(g_j) > \bar{w}$; more generally a group commutator in g_i and g_j with α occurrences of g_i and β occurrences of g_j is equal to 1 if $\alpha w(g_i) + \beta w(g_j) > \bar{w}$. The equivalent fact holds in the corresponding Lie algebra. A Lie commutator in $\mathrm{Log}(g_i)$ and $\mathrm{Log}(g_j)$ with α occurrences of $\mathrm{Log}(g_i)$ and β occurrences of $\mathrm{Log}(g_j)$ is equal to zero if $\alpha w(g_i) + \beta w(g_j) > \bar{w}$. This can be used to reduce the number of commutators which have to be evaluated during the computation of $\mathrm{Log}(g_i) * \mathrm{Log}(g_j)$.

For the method of §4.3 for computing Log and Exp, we use the structure constant table of the Lie algebra $\mathcal{L}(G)$ as computed by the method of §4.2. Further the terms of the BCH-formula are determined using [36, §8] and the Dynkin bracket operator defined in [44, Chapter 2].

Example groups

We use the following two classes of examples of polycyclically presented $\mathcal{T}$-groups to test our implementations. For background on algebraic number theory see Appendix A.

1. Let $\mathbb{Q}(\theta)$ be an algebraic extension of $\mathbb{Q}$ and $\mathcal{O}$ its maximal order. Then we denote by $\mathrm{Tr}_1(\mathcal{O})$ the group of upper-unitriangular matrices in $\mathrm{GL}(n,\mathcal{O})$. In a similar way to $\mathrm{Tr}_1(n,\mathbb{Q})$, we can compute a constructive polycyclic sequence for $\mathrm{Tr}_1(n,\mathcal{O})$, which then yields a polycyclic presentation for $\mathrm{Tr}_1(n,\mathcal{O})$. We use the irreducible polynomials $p_1(x) = x^2 - 3$ and $p_2(x) = x^3 - x^2 + 4$ for our examples. By $\mathcal{O}_i$ we denote the maximal order of $\mathbb{Q}(\theta_i)$ where θ_i is a zero of the polynomial p_i.

2. Let F_n be the free group on n generators $f_1, \ldots, f_n$. Then $F_{n,c} = F_n/\gamma_{c+1}(F_n)$, where γ_i denotes i-th term of the lower central series, is the free nilpotent of class c group on n generators. It is a $\mathcal{T}$-group and we use the nilpotent quotient algorithm in the GAP package NQ [32] to compute a polycyclic presentation for it.

Runtimes

In Table 4.1 we indicate the time that is needed to set up the Mal'cev correspondence for several examples of $\mathcal{T}$-groups. Further the time that is needed to compute the polynomials $\overline{\alpha}_i$, $\overline{e}_j$, defined in §4.3, are displayed.

All computations were carried out in GAP Version 4.4.7 on a 3 gigahertz Pentium 4 processor.

Discussion of the results

The runtimes displayed in Table 4.1 show that setting up the Mal'cev correspondence with the help of the BCH-formula, as described in §4.2, is more efficient than using matrix representations, as described in §4.1. For the tested examples the BCH-method is usually 100 to 1000 times faster than the matrix method.

Our experiments also show that the average time needed for computing Log and Exp using the method from §4.2 is faster than using the method from §4.1. Further the symbolic approach of §4.3 yields an additional speed up for computing Log and Exp. For example for the group $F_{2,8}$ for random elements of range 1024 (i.e. elements of the form $g = g_1^{e_1} \cdots g_k^{e_k}$, where e_i is a randomly chosen integer in $[-1024, \ldots, 1024]$) computing Log takes only 10 milliseconds (symbolic) instead of 106 milliseconds (BCH) or 1979 milliseconds (Matrix approach). However a considerable amount of time is needed to compute the polynomials used for the symbolic method. In the case of the group $F_{3,6}$ our implementation takes 15 seconds to compute these polynomials. Thus the setup of the symbolic method from §4.3 is time consuming but it only has to be done once. Then it yields a considerable speed up for the computation of Log and Exp.

Group	Hl	Class	Matrix	BCH	Pols
$F_{2,2}$	3	2	16	4	8
$F_{2,3}$	5	3	36	6	20
$F_{2,4}$	8	4	80	8	36
$F_{2,5}$	14	5	380	20	120
$F_{2,6}$	23	6	1756	44	380
$F_{2,7}$	41	7	14825	196	1684
$F_{2,8}$	71	8	154000	776	6536
$F_{3,2}$	6	2	32	4	20
$F_{3,3}$	14	3	292	20	68
$F_{3,4}$	32	4	3256	80	256
$F_{3,5}$	80	5	97239	820	1940
$F_{3,6}$	196	6	3504971	8681	14833
$G(\mathrm{Tr}_1(2,\mathcal{O}_1))$	2	1	8	1	1
$G(\mathrm{Tr}_1(3,\mathcal{O}_1))$	6	2	48	4	12
$G(\mathrm{Tr}_1(4,\mathcal{O}_1))$	12	3	180	16	48
$G(\mathrm{Tr}_1(5,\mathcal{O}_1))$	20	4	1048	68	108
$G(\mathrm{Tr}_1(6,\mathcal{O}_1))$	30	5	5784	232	324
$G(\mathrm{Tr}_1(7,\mathcal{O}_1))$	42	6	47116	772	880
$G(\mathrm{Tr}_1(8,\mathcal{O}_1))$	56	7	393325	1988	2813
$G(\mathrm{Tr}_1(2,\mathcal{O}_2))$	3	1	12	4	4
$G(\mathrm{Tr}_1(3,\mathcal{O}_2))$	9	2	64	8	32
$G(\mathrm{Tr}_1(4,\mathcal{O}_2))$	18	3	631	48	132
$G(\mathrm{Tr}_1(5,\mathcal{O}_2))$	30	4	4969	188	432
$G(\mathrm{Tr}_1(6,\mathcal{O}_2))$	45	5	32630	664	2184
$G(\mathrm{Tr}_1(7,\mathcal{O}_2))$	63	6	363484	2069	11972

Table 4.1: Setup of the Mal'cev correspondence and symbolic Log and Exp: The second and third column indicate the Hirsch Length and the class of the given example group. In the fourth, respectively fifth, column we display the time in milliseconds that is needed to set up the Mal'cev correspondence via the matrix approach (see §4.1), respectively the BCH approach (see §4.2). In the sixth column we see the time in milliseconds that is needed to compute the polynomials describing Log and Exp (see §4.3).

Chapter 5

Mal'cev collection

In this chapter we describe a collection algorithm, which we call Mal'cev collection, for polycyclically presented groups. The Mal'cev correspondence, nomen est omen, plays a central role in this algorithm.

This work is motivated by a paper of du Sautoy [16]. He uses the Mal'cev correspondence to investigate the nature of functions that describe the collection process in splittable polycyclic groups. This so-called symbolic collection will be discussed in more detail in Chapter 6.

In §5.1 we recall some well-known facts about collection in polycyclically presented groups; in particular we mention collection from the left and Deep Thought. Then we describe the Mal'cev collection in §5.2. The underlying methods for computations with automorphisms of $\mathcal{T}$-groups are give in §5.3 and §5.4. Finally, in §5.5 we report on our implementation of the Mal'cev collection in GAP and compare it with collection from the left.

5.1 Classical collection

Let G be a polycyclic group given by a polycyclic presentation $\mathcal{P}$ with respect to a polycyclic sequence $\mathcal{G} = (g_1, \ldots, g_n)$.

Definition 5.1.1. Let $w = w(g_1, \ldots, g_n)$ be a word in $g_1, \ldots, g_n$. A method for computing the normal form of w with respect to $\mathcal{G}$ is called a *collection algorithm*. The word w is called *collected* if it is in normal form.

The performance of algorithms for computations in polycyclically presented groups depends very considerably on the ability to do collection efficiently. Typically the word w is the product of two elements given in normal form, i.e. $w = \mathrm{nf}(g)\mathrm{nf}(h)$ for some $g, h \in G$.

Several strategies for collection in polycyclic groups have been studied intensively, see for example [18, 24, 43]. The current state of the art is

"collection from the left (Cftl)". The following algorithm is a simple version of Cftl.

CollectionFromTheLeft($\mathcal{P}$**,** w **)**

1: **while** w is not collected **do**
2: let x be the leftmost uncollected subterm in w of the form $g_i^{r_i}$ or $g_i g_j^{\pm 1}$ where $j < i$.
3: replace x by a collected subterm according to the relations of $\mathcal{P}$.
4: **end while**

Cftl was successfully used for structural explorations of finite polycyclically presented groups of very large order, see for example [34]. In finite polycyclically presented groups Cftl benefits from the fact that all relative orders are finite, and thus the exponents of generators arising in the collection process are bounded. However in infinite polycyclic groups this is not the case. Thus, one of the main challenges of implementations of Cftl for infinite polycyclic groups is dealing with large exponents.

To our knowledge the fastest current implementation of collection from the left is part of MAGMA [11]. A description of this implementation can be found in [18]; it uses methods related to repeated squaring to handle big exponents.

The complexity of collection from the left is known to be exponential in the number n of generators [24].

For collection with respect to certain "nice" polycyclic sequences, much better methods are known. For $\mathcal{T}$-groups, i.e. finitely generated torsion-free nilpotent groups, we have the following result due to Hall [19]. Recall that for a polycyclic group G with polycyclic sequence $(g_1, \ldots, g_n)$ and $x \in \mathbb{Z}^n$ we denote $g^x = g_1^{x_1} \cdots g_n^{x_n}$.

Theorem 5.1.2. *Let G be a $\mathcal{T}$-group with Mal'cev basis $\mathcal{G} = (g_1, \ldots, g_n)$. Let $\zeta = (\zeta_i) : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{Z}^n$ be the collection function of G with respect to $\mathcal{G}$ so that*

$$g^x g^y = g^{\zeta(x,y)}$$

and let $\omega = (\omega_i) : \mathbb{Z}^n \times \mathbb{Z} \to \mathbb{Z}^n$ the powering function of G so that

$$(g^x)^k = g^{\omega(x,k)}.$$

Then ζ and ω are given by polynomials over $\mathbb{Q}$.

Proof. By Section 3.3 we have that

$$g^{\zeta(x,y)} = \mathrm{Exp}(\mathrm{Log}(g^x) * \mathrm{Log}(g^y))$$

and $g^\omega = \mathrm{Exp}(k \,\mathrm{Log}(g_1^{x_1} \cdots g_n^{x_n}))$. By Lemma 4.3.1 and the fact that $x * y$ has only finitely many nonzero terms for $x, y \in \mathcal{L}(G)$, we deduce that ζ_i and ω_j are rational polynomials. □

This result can be used for computational applications. Leedham-Green and Soicher developed the algorithm "Deep Thought" [25], which computes these polynomials and uses them for collection in $\mathcal{T}$-groups. An implementation of Deep Thought by Merkwitz [29] is part of the GAP system. Deep Thought yields a big speed up compared to collection from the left for the multiplication of two random elements of a $\mathcal{T}$-group.

5.2 Collection using the Mal'cev correspondence

Let G be an infinite polycyclic group. In this section we show that, with respect to a carefully chosen polycyclic sequence $\mathcal{G}$ of G, collection in G can be reduced to the following 3 subtasks:

(1) Collection and powering in a $\mathcal{T}$-subgroup of G.

(2) Computations with powers of automorphisms and consecutive powers of automorphisms of a normal $\mathcal{T}$-subgroup of G.

(3) Computations with coset representatives of a subgroup of finite index of G.

As discussed in §5.1, (1) is well understood and we can apply standard methods such as Deep Thought to it. For (2) we will use the Mal'cev correspondence as explained in §5.3 and §5.4. For (3) we use collection from the left.

5.2.1 Choosing the polycyclic sequence $\mathcal{G}$

Let G be an infinite polycyclic group. Now we explain how to choose the polycyclic sequence $\mathcal{G}$ mentioned at the beginning of §5.2.

Recall that by §2.5 we can compute a normal $\mathcal{T}$-group N and a $\mathcal{T}$-group C such that $H = CN$ is normal of finite index in G and H/N is free abelian of finite rank. By computing the upper central series of N as described in [17, Chapter 9] we can obtain a Mal'cev basis of N.

Let $\mathcal{N} = (n_1, \ldots, n_l)$ be a Mal'cev basis of N and let $(c_1 N, \ldots, c_k N)$ be a basis for the free abelian group CN/N. Then $\mathcal{H} = (c_1, \ldots, c_k, n_1, \ldots, n_l)$ is

a polycyclic sequence for $H = CN$. Further there exist $f_1, \ldots, f_j \in G$ such that $(f_1H, \ldots, f_jH)$ is a polycyclic sequence for G/H. Now we set

$$\mathcal{G} = (f_1, \ldots, f_j, c_1, \ldots, c_k, n_1, \ldots, n_l)$$

which is a polycyclic sequence for G.

Lemma 5.2.1. *The list $(c_1, \ldots, c_k)$ can be extended to a Mal'cev basis*

$$\mathcal{C} = (c_1, \ldots, c_k, c_{k+1}, \ldots, c_{k+m})$$

of the $\mathcal{T}$-group C.

Proof. The upper central series of $C \cap N$ has torsion-free factors and is invariant under the action of $(c_1, \ldots, c_k)$.

This series can be refined to a central series with torsion-free factors which are centralized by $(c_1, \ldots, c_k)$. To see this, let M be one of the torsion-free factors. Denote by A the centraliser of $(c_1, \ldots, c_k)$ in M. Then A is non-trivial, since C acts nilpotently on M, and furthermore M/A is torsion-free, since for $m \in M, z \in \mathbb{N}$, the equality $zm = (zm)^{c_i} = z(m^{c_i})$ implies $m^{c_i} = m$. Thus, by induction on the dimension of M, we get a strictly ascending series of submodules of M with torsion-free factors, which are by construction centralized by $(c_1, \ldots, c_k)$.

Finally we set $(c_{k+1}, \ldots, c_{k+m})$ to be a Mal'cev basis of $C \cap N$ which goes through the refined upper central series of $C \cap N$. Then $\mathcal{C}$ is a Mal'cev basis of C. □

5.2.2 Mal'cev collection in $H = CN$

Now we show that in H collection with respect to $\mathcal{H}$ can be reduced to the subtasks (1) and (2). Denote by $c^x n^{\bar{x}}$ the element in H given by the exponent vector $(x_1, \ldots, x_k, \bar{x}_1, \ldots, \bar{x}_l)$ with respect to $\mathcal{H}$. For two elements $c^x n^{\bar{x}}, c^y n^{\bar{y}} \in H$ we have

$$c^x n^{\bar{x}} c^y n^{\bar{y}} = c^x c^y (n^{\bar{x}})^{(c^y)} n^{\bar{y}}.$$

Since H/N is free abelian, the normal form of $c^x c^y$ with respect to $\mathcal{C}$ is of the form $c^{x+y} c_{k+1}^{z_{k+1}} \cdots c_{k+m}^{z_{k+m}}$. The computation of the tail $t = c_{k+1}^{z_{k+1}} \cdots c_{k+m}^{z_{k+m}}$ is a computation entirely in the $\mathcal{T}$-group C and therefore a part of subtask (1).

We can also compute the normal form of the tail $t \in C \cap N$ with respect to $\mathcal{N}$ as part of (1). For this purpose we compute the normal forms of $c_{k+1}, \ldots, c_{k+m}$ with respect to $\mathcal{N}$ as part of the setup. Then computing the

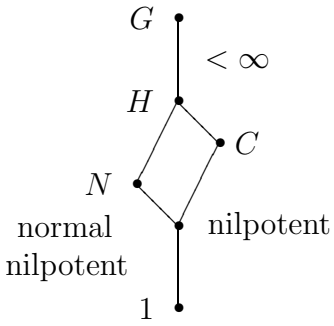


Figure 5.1: Let G be an infinite polycyclic group. There is a normal $\mathcal{T}$-group N and a $\mathcal{T}$-group C such that $H = CN$ is normal of finite index in G and H/N is free abelian of finite rank. In §5.2 we describe an effective collection method with respect to a polycyclic sequence $\mathcal{G}$ going through the normal series $1 \leq N \leq H \leq G$.

normal form of t with respect $\mathcal{N}$ reduces to m powering operations and $m-1$ multiplications in N.

The efficient computation of the normal form of

$$(n^{\bar{x}})^{(c^y)} = (n^{\bar{x}})^{(c_1^{y_1} \cdots c_k^{y_k})}$$

is the crucial step of our method. It is a computation with automorphisms of N and therefore part of (2). Finally, the multiplication of $\mathrm{nf}(t), \mathrm{nf}((n^{\bar{x}})^{(c^y)})$ and $n^{\bar{y}}$ in N can be done again as a part of (1).

5.2.3 Inversion in $H = CN$

Let $c^x n^{\bar{x}} \in H$ as in §5.2.2. The computation of the normal form of $g = (c^x n^{\bar{x}})^{-1}$ can be done as follows. Denoting $c = c^x$ and $n = n^{\bar{x}}$ we have that $g = c^{-1}(n^{-1})^{(c^{-1})}$. Inverting in C and N is part of (1), and thus the normal form of c^{-1} with respect to $\mathcal{C}$ and the normal form of n^{-1} with respect to $\mathcal{N}$ can be determined efficiently. Similarly to §5.2.2, we transform the normal form of c^{-1} with respect to $\mathcal{C}$ to an element of the form $c^y n^{\bar{y}}$. The remaining computation of the normal form of $n^{\bar{y}}(n^{-1})^{(c^{-1})}$ can be done as in §5.2.2 and therefore be reduced to (1),(2).

5.2.4 Powering in $H = CN$

Let $c^x n^{\bar{x}} \in H$ as in §5.2.2. We describe a method to compute the normal form of $(c^x n^{\bar{x}})^q$ where $q \in \mathbb{Z}$.

By inverting $c^x n^{\bar{x}}$ with the method of §5.2.3 if necessary, we can assume that $q \geq 0$. If we denote $c = c^x$ and $n = n^{\bar{x}}$ then we have

$$(cn)^q = c^q \underbrace{n^{(c^{q-1})} \cdots n^c n}_{t}.$$

The group H/N is free abelian. Thus

$$c^q = (c^x)^q = c_1^{qx_1} \cdots c_k^{qx_k} c_{k+1}^{z_{k+1}} \cdots c_{k+m}^{z_{k+m}}$$

for some $z_{k+1}, \ldots, z_{k+m} \in \mathbb{Z}$. The computation of the tail $s = c_{k+1}^{z_{k+1}} \cdots c_{k+m}^{z_{k+m}}$ is a powering computation in the $\mathcal{T}$-group C and therefore a part of subtask (1). We can also compute the normal form of the tail $s \in C \cap N$ with respect to $\mathcal{N}$ as part of (1), as described in §5.2.2. The computation of the normal form of t is a subtask of (2). Finally, the computation of the normal form of st is again a part of (1).

5.2.5 Mal'cev collection in G

Now we describe our collection method with respect to $\mathcal{G}$. Denote by $f^x c^{\bar{x}} n^{\tilde{x}}$ the element in G given by the exponent vector

$$(x_1, \ldots, x_j, \bar{x}_1, \ldots, \bar{x}_k, \tilde{x}_1, \ldots, \tilde{x}_l)$$

with respect to $\mathcal{G}$. For two elements $f^x c^{\bar{x}} n^{\tilde{x}}, f^y c^{\bar{y}} n^{\tilde{y}} \in G$ we have

$$f^x c^{\bar{x}} n^{\tilde{x}} f^y c^{\bar{y}} n^{\tilde{y}} = \underbrace{f^x f^y}_{f^r c^{\bar{r}} n^{\tilde{r}}} \underbrace{(c^{\bar{x}})^{(f^y)}}_{c^{\bar{s}} n^{\tilde{s}}} \underbrace{c^{\bar{y}} (n^{\tilde{x}})^{(f^y c^{\bar{y}})} n^{\tilde{y}}}_{c^{\bar{t}} n^{\tilde{t}}},$$

where $f^r c^{\bar{r}} n^{\tilde{r}}$, $c^{\bar{s}} n^{\tilde{s}}$ and $c^{\bar{t}} n^{\tilde{t}}$ are the normal forms with respect to $\mathcal{G}$ of the corresponding expressions in the brackets above them.

- The computation of $c^{\bar{t}} n^{\tilde{t}}$ can be reduced to the subtasks (1),(2) as explained in §5.2.2.
- For the computation of $c^{\bar{s}} n^{\tilde{s}}$ we use the equality

  $$(c^{\bar{x}})^{(f^y)} = (c_1^{(f^y)})^{\bar{x}_1} \cdots (c_k^{(f^y)})^{\bar{x}_k}.$$

  The normal form of $c_i^{(f^y)} \in H$ can be precomputed for $i = 1, \ldots, k$ and every $f^y H \in G/H$, and therefore can be assumed to be given. Then $\mathrm{nf}((c_i^{(f^y)})^{\bar{x}_i})$ can be computed using the methods for powering in H of §5.2.4. The remaining computation of the normal form of $\mathrm{nf}((c_1^{(f^y)})^{\bar{x}_1}) \cdots \mathrm{nf}((c_k^{(f^y)})^{\bar{x}_k})$ can be done by again using the methods for H.

- The normal form $f^r c^{\bar{r}} n^{\tilde{s}}$ of $f^x f^y$ can be precomputed for all $f^x H, f^y H \in G/H$ and therefore assumed to be given.
- Finally, the computation of the normal form of $c^{\bar{r}} n^{\tilde{r}} c^{\bar{s}} n^{\tilde{s}} c^{\bar{t}} n^{\tilde{t}}$ can be done with the method of §5.2.2.

5.2.6 Inversion in G

Let $g = f^x c^{\bar{x}} n^{\tilde{x}}$ be an element in G given in normal form with respect to $\mathcal{G}$. Then $g^{-1} = (c^{\bar{x}} n^{\tilde{x}})^{-1} (f^x)^{-1}$. Thus, by precomputing the normal forms of all elements $(f^x)^{-1}$, we can reduce the computation of the normal form of g^{-1} to inversion in CN and collection in G.

5.3 Computations with powers of automorphisms of $\mathcal{T}$-groups

Let N be a $\mathcal{T}$-group given by a polycyclic presentation with respect to a Mal'cev basis $\mathcal{N} = (n_1, \ldots, n_l)$. Let φ be an automorphism of N, given by the list $(n_1^\varphi, \ldots, n_l^\varphi)$. In this section we describe an effective method to compute the normal form of $n^{(\varphi^q)}$, where $n \in N$ and $q \in \mathbb{Z}$.

As explained in §3.3, let $\mathcal{L}(N)$ be the Lie algebra corresponding to the radicable hull of N. Then $\{\mathrm{Log}(n_1), \ldots, \mathrm{Log}(n_l)\}$ is a basis for $\mathcal{L}(N)$. We define a $l \times l$ matrix Φ by

$$\mathrm{Log}(n_i^\varphi) = \sum_{j=1}^{l} \Phi_{ij} \mathrm{Log}(n_i).$$

By Theorem 3.3.6 the matrix Φ is a representation of the Lie algebra isomorphism $\mathrm{Exp} \circ \varphi \circ \mathrm{Log}$, with respect to the basis $\{\mathrm{Log}\, n_1, \ldots, \mathrm{Log}\, n_l\}$. This yields the following algorithm.

ApplyPowerOfAutomorphism(n**,** φ**,** q **)**

1: determine $\gamma =$ **Logarithm(** n **)**.
2: compute $\bar{\gamma} = \gamma \cdot \Phi^q$.
3: compute $g =$ **Exponential(** $\bar{\gamma}$ **)**.
4: return g.

For the realization of Step 1 and 3, see Chapter 4. Note that for Step 2 repeated squaring can be used.

If we want to apply several powers of automorphisms $\varphi_1^{q_1} \varphi_2^{q_2} \cdots \varphi_k^{q_k}$, as in §5.2, we switch only once from n to the corresponding element γ in the Lie algebra, then multiply γ with $\Phi_1^{q_1} \cdots \Phi_k^{q_k}$, where Φ_i is the matrix representation

of the Lie algebra isomorphism corresponding to the group automorphism φ_i, and then switch back to the representation with respect to $(n_1, \ldots, n_l)$.

5.4 Computations with consecutive powers of automorphisms of $\mathcal{T}$-groups

Let N be a $\mathcal{T}$-group and $\varphi \in \mathrm{Aut}(N)$ be given as in §5.3. We describe an effective method to compute the normal form of

$$\pi_{q+1} = n^{(\varphi^q)} n^{(\varphi^{q-1})} \cdots n^{\varphi} n$$

where $n \in N$ and $q \in \mathbb{N}$.

As in §5.3 we use the Lie algebra $\mathcal{L}(N)$ and the Lie algebra automorphism $\Phi \in \mathrm{Aut}(\mathcal{L}(N))$ corresponding to φ for this purpose; if we denote $x = \log(n)$ then we are interested in computing the coefficients of the vector

$$\Pi_{q+1} = (x\Phi^q) * (x\Phi^{(q-1)}) * \cdots * (x\Phi) * x$$

because $\mathrm{Exp}(\Pi_{q+1}) = \pi_{q+1}$.

Our method is a variation of repeated squaring; we use a binary representation of q and the identities

$$\begin{aligned} \Pi_{2p} &= (\Pi_p \Phi^p) * \Pi_p \\ \Pi_{2p+1} &= (\Pi_{2p}\Phi) * x. \end{aligned}$$

This reduces the computations of the coefficients of Π_q to $\log(q)$ matrix and vector multiplications and $\log(q)$ $*$-operations.

5.5 Implementation and runtimes

The Mal'cev collection algorithm has been fully implemented in GAP and is part of the Guarana package [2]. In this section we make comments on our implementation and compare it with collection from the left. All computations have been carried out on a 3 gigahertz Pentium 4 processor. Indications of memory usage will be given later at the appropriate places.

5.5.1 Example groups

Throughout this section we use the following classes of example groups. For background on algebraic number theory see Appendix A.

1. Let $\mathbb{Q}(\theta)$ be an algebraic extension of $\mathbb{Q}$ and $\mathcal{O}$ its maximal order. Let $\mathrm{Tr}_n(\mathcal{O})$ be the group of upper-triangular matrices in $\mathrm{GL}(n,\mathcal{O})$, $\mathrm{Tr}_1(n,\mathcal{O})$ the subgroup of matrices in $\mathrm{Tr}_n(\mathcal{O})$ with 1s on the diagonal and $D_n(\mathcal{O})$ the group of diagonal matrices in $\mathrm{GL}(n,\mathcal{O})$. Every polycyclic group has a subgroup of finite index which can be embedded in some $\mathrm{Tr}_n(\mathcal{O})$ [38, page 132]. Therefore this class of groups is very suitable for testing our collection algorithm.

Let $U(\mathcal{O})$ be the group of units of $\mathcal{O}$. As a consequence of Dirichlet's Units Theorem, $U(\mathcal{O})$ is polycyclic and therefore $D_n(\mathcal{O})$ is polycyclic as well. Using the torsion unit and fundamental units of $U(\mathcal{O})$, it is straightforward to obtain a polycyclic presentation for $D_n(\mathcal{O})$.

As mentioned in §4.4 we can compute a polycyclic presentation for the group $\mathrm{Tr}_1(n,\mathcal{O})$. Since $\mathrm{Tr}_n(\mathcal{O}) = D_n(\mathcal{O}) \ltimes \mathrm{Tr}_1(n,\mathcal{O})$, it is straightforward to obtain a polycyclically presented group $G(\mathrm{Tr}_n(\mathcal{O}))$ being isomorphic to $\mathrm{Tr}_n(\mathcal{O})$.

We use the irreducible polynomials $p_1(x) = x^2 - 3$ and $p_2(x) = x^3 - x^2 + 4$ for our examples. By $\mathcal{O}_i$ we denote the maximal order of $\mathbb{Q}(\theta_i)$ where θ_i is a zero of the polynomial p_i.

2. Let $F_{n,c}$ be the free nilpotent of class c group on n generators. As explained in §4.4 we can compute a polycyclic presentation for $F_{n,c}$. An automorphism φ of the free group F_n naturally induces an automorphism $\bar{\varphi}$ of $F_{n,c}$.

We use the automorphism φ_1 of F_2 which maps f_1 to f_2^{-1} and f_2 to $f_1 f_2^3$ and the automorphism φ_2 of F_3 mapping f_1 to f_2^{-1}, f_2 to f_3^{-1} and f_3 to $f_2^{-3} f_1^{-1}$ for our examples.

Using an automorphism ψ of $F_{n,c}$ we can construct a polycyclically presented group $G(\langle\psi\rangle \ltimes F_{n,c})$ which is isomorphic to $\langle\psi\rangle \ltimes F_{n,c}$.

Note that all example groups in this section are extensions of the groups from §4.4.

5.5.2 Runtimes setup

In Table 5.1 we display the time that is needed for the complete setup of the Mal'cev collector. We assume that the input group is given by a polycyclic presentation with respect to a nice polycyclic sequence in the sense of §5.2.1 and that the subgroup C is given by a polycyclic presentation with respect to a Mal'cev basis. The setup of the Mal'cev collector includes the setup of the Mal'cev correspondence for the normal subgroup N as described in §4.2, the computation of the polynomials describing Log and Exp as described in 4.3, the computation of the multiplication table of G/CN, the computation of the Deep Thought collector for C and N and all other information that

is needed to do Mal'cev collection. All computations have been carried out with 80 MB of memory for GAP.

5.5.3 Mal'cev collection versus collection from the left

In Table 5.2 we display the average runtime for the multiplication of two random elements in our example groups. The compared methods are collection from the left as implemented in MAGMA V2.12-14 and Mal'cev collection. For a group G with polycyclic sequence $(g_1, \ldots, g_k)$ we say that a random element $g \in G$ is of *range* $r \in \mathbb{N}$ if is of the form $g = g_1^{e_1} \cdots g_k^{e_k}$, where e_i is a randomly chosen integer in $[-r, \ldots, r]$. The Mal'cev collector uses the implementation of Deep Thought in GAP as the collection method in the $\mathcal{T}$-groups C and N.

In Table 5.2 we see that Cftl is more efficient than Mal'cev collection for random elements of very small range such as 1. This is not surprising. For elements g, h of small range Cftl needs to do very few replacements to yield the normal form of gh. We note that the runtime of Cftl can differ considerably because the size of the exponents of the normal form of gh varies a lot for random elements g, h of same range.

For random elements of bigger range Mal'cev collection dramatically outperforms Cftl. It is much faster and also less memory consuming. The multiplication of random elements of big range such as 1000 was not possible with Cftl with 1 GB of memory, while Mal'cev needed at most 85 MB of memory.

Therefore it depends very much on the context which method should be applied. For computations where typically elements with sparse and small exponent vectors are multiplied, Cftl is preferable. For example for the computation of a polycyclic sequence for the derived subgroup it might better to choose Cftl, since we mainly compute normal forms of commutators in the original generators. Computations that involve slightly more complicated elements are better done with Mal'cev collection. For those elements Mal'cev is much faster even if we include the cost of the setup of Mal'cev collector. Further the multiplication of elements of big range is often not possible with Cftl since we run out of memory.

It might be a good idea to use a hybrid collector that combines both methods. By looking at the exponent vectors of the input elements this collector could make an estimate whether Cftl or Mal'cev should be chosen. Alternatively the hybrid collector could run Cftl and Mal'cev in parallel on the same input and return the result of the method that finished first.

Group	Hl	Setup time
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{22})$	4	40
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{23})$	6	48
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{24})$	9	48
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{25})$	15	160
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{26})$	24	508
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{27})$	42	2232
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{28})$	72	8447
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{32})$	7	30
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{33})$	15	96
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{34})$	33	543
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{35})$	81	4201
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{36})$	197	41376
$G(\mathrm{Tr}_2(\mathcal{O}_1))$	4	16
$G(\mathrm{Tr}_3(\mathcal{O}_1))$	9	48
$G(\mathrm{Tr}_4(\mathcal{O}_1))$	16	148
$G(\mathrm{Tr}_5(\mathcal{O}_1))$	25	428
$G(\mathrm{Tr}_6(\mathcal{O}_1))$	36	1272
$G(\mathrm{Tr}_7(\mathcal{O}_1))$	49	3864
$G(\mathrm{Tr}_8(\mathcal{O}_1))$	64	12757
$G(\mathrm{Tr}_2(\mathcal{O}_2))$	5	28
$G(\mathrm{Tr}_3(\mathcal{O}_2))$	12	76
$G(\mathrm{Tr}_4(\mathcal{O}_2))$	22	292
$G(\mathrm{Tr}_5(\mathcal{O}_2))$	35	1036
$G(\mathrm{Tr}_6(\mathcal{O}_2))$	51	3949
$G(\mathrm{Tr}_7(\mathcal{O}_2))$	70	17197

Table 5.1: Setup of the Mal'cev collector: In the second column we indicate the Hirsch length of the given example group. In the third column we specify the time in milliseconds that is needed for the complete setup of the Mal'cev collector.

Group	range = 1		range = 10		range = 100		range = 1000	
	Cftl	Mal'cev	Cftl	Mal'cev	Cftl	Mal'cev	Cftl	Mal'cev
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{22})$	1	1	3	1	2481	1	*	1
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{23})$	1	1	27	1	*	2	*	3
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{24})$	1	2	108	3	*	4	*	11
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{25})$	1	5	5627	7	*	14	*	151
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{26})$	1	11	21276	19	*	57	*	810
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{27})$	2	52	485532	120	*	501	*	9269
$G(\langle\bar{\varphi}_1\rangle \ltimes F_{28})$	17	150	*	522	*	2442	*	58497
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{32})$	1	1	7	1	*	2	*	3
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{33})$	1	3	252	4	*	7	*	40
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{34})$	1	9	3160	13	*	63	*	568
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{35})$	1	62	325312	158	*	1185	*	17473
$G(\langle\bar{\varphi}_2\rangle \ltimes F_{36})$	64	434	*	2268	*	22377	*	310518
$G(\mathrm{Tr}_2(\mathcal{O}_1))$	1	1	1	2	4	3	67	3
$G(\mathrm{Tr}_3(\mathcal{O}_1))$	1	5	2	7	20	9	308	11
$G(\mathrm{Tr}_4(\mathcal{O}_1))$	1	9	19	14	978	16	*	28
$G(\mathrm{Tr}_5(\mathcal{O}_1))$	1	20	140	28	9068	35	*	81
$G(\mathrm{Tr}_6(\mathcal{O}_1))$	5	56	549	83	27510	112	*	391
$G(\mathrm{Tr}_7(\mathcal{O}_1))$	28	127	1899	190	150030	276	*	1043
$G(\mathrm{Tr}_8(\mathcal{O}_1))$	80	496	5884	785	*	1381	*	5450
$G(\mathrm{Tr}_2(\mathcal{O}_2))$	1	2	1	3	9	3	187	4
$G(\mathrm{Tr}_3(\mathcal{O}_2))$	1	5	3	8	47	10	732	15
$G(\mathrm{Tr}_4(\mathcal{O}_2))$	1	15	85	23	6705	28	*	72
$G(\mathrm{Tr}_5(\mathcal{O}_2))$	4	43	829	59	34520	77	*	216
$G(\mathrm{Tr}_6(\mathcal{O}_2))$	25	157	3233	225	*	318	*	1238
$G(\mathrm{Tr}_7(\mathcal{O}_2))$	114	501	12118	700	*	1008	*	3870

Table 5.2: Runtimes for the multiplication of two random elements: This table specifies the average runtime in milliseconds of 1000 computations of the normal form of gh where g, h are randomly chosen group elements of range r, i.e. elements of the form $g = g_1^{e_1} \cdots g_k^{e_k}$, where e_i is a randomly chosen integer in $[-r, \ldots, r]$. The two compared methods are Cftl, i.e. collection from the left as implemented in MAGMA V2.12-14 and Mal'cev collection as implemented in the GAP package Guarana. The symbol '*' indicates that the computation of the average runtime was aborted because it needed more than 1 GB of memory.

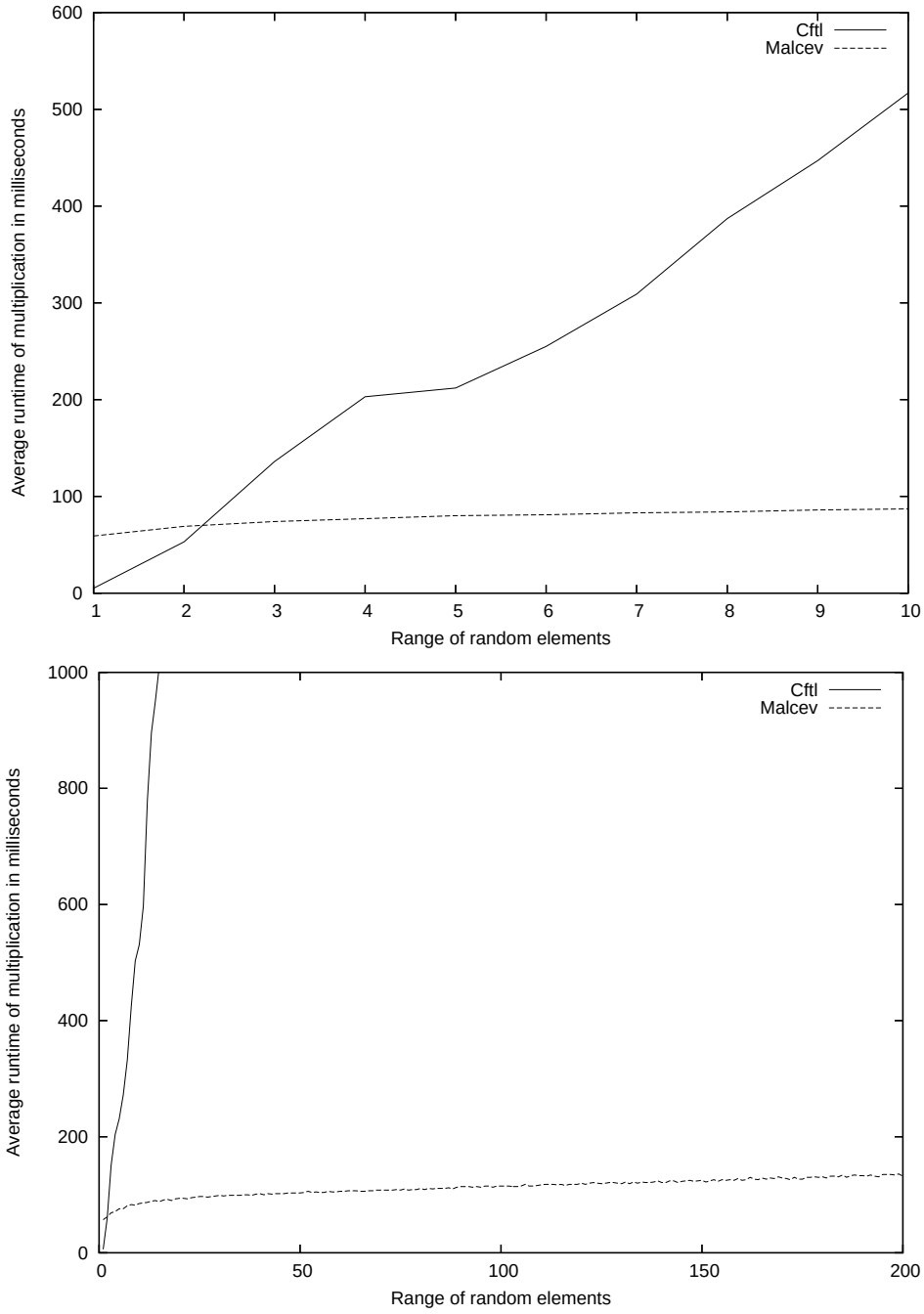


Figure 5.2: For the group $G(\mathrm{Tr}_6(\mathcal{O}_1))$ the average runtime of the collection algorithm Mal'cev and Cftl as a function in the range of the multiplied random elements is displayed.

5.5.4 Concluding remarks

We have seen in §5.5.3 that Mal'cev collection can be used to speed up collection in polycyclically groups considerably. However this collection method only works if the group is given by a polycyclic presentation with respect to a nice polycyclic sequence in the sense of §5.2.1.

All example groups in this chapter have been constructed around such a polycyclic sequence. This is a natural way to construct infinite polycyclic groups.

As mentioned in §2.5 the methods in [17, Chapter 9] can be used to compute such a nice polycyclic sequence for an arbitrary polycyclically presented group. Therefore it would be desirable to have a very efficient implementation of these methods. So far a prototype implementation exists. Since such a polycyclic sequence which goes through a nilpotent-by-abelian-by finite normal series of the group is a natural starting point for further investigations, it is desirable to compute it anyway.

Chapter 6

Symbolic collection

Let G be a polycyclic group with a basis $\mathcal{G} = (g_1, \ldots, g_n)$, i.e. $\mathcal{G}$ is a polycyclic sequence of G such that all relative orders are infinite. Then each element $g \in G$ has unique normal form $g_1^{e_1} \cdots g_n^{e_n}$ where $e_i \in \mathbb{Z}$.

With respect to the basis $\mathcal{G}$ we can define collection functions $\zeta_1, \ldots, \zeta_n$ in $2n$ integer variables $x_1, \ldots, x_n, y_1, \ldots, y_n$ by

$$g_1^{x_1} \cdots g_n^{x_n} g_1^{y_1} \cdots g_n^{y_n} = g_1^{\zeta_1} \cdots g_n^{\zeta_n}$$

where the right hand side is the normal form of the left hand side of this equation.

As already mentioned in §5.1 a result due to Hall [19] states that, if $\mathcal{G}$ is a Mal'cev basis, then the functions ζ_i are rational polynomials.

Hall's result was generalised by du Sautoy to the class of splittable polycyclic groups [16]. He showed that in those groups the functions ζ_i with respect to a carefully chosen basis are polynomials over a number field $\mathbb{F}$ in $x_1, \ldots, x_n, y_1, \ldots, y_n$ and a finite number of expressions of the form $\omega_{ij}^{y_i}$ where $\omega_{ij} \in \mathbb{F}$. Note that every polycyclic group has a splittable subgroup of finite index [38, Chapter 7].

In this chapter we describe an algorithm that computes the collection functions for splittable polycyclic groups. It is based on the partially constructive proof of du Sautoy in [16] and the constructive methods in Chapter 4. This algorithm can be seen as an extension of the algorithm "Deep Thought" by Leedham-Green and Soicher, which computes Hall polynomials.

In §6.1 and §6.2 we give some necessary background material on the multiplicative Jordan decomposition and splittable polycyclic groups. Then in §6.3 we outline the algorithm for the computation of collection functions in splittable polycyclic groups. Finally, in §6.4 we comment on applications of this algorithm.

6.1 Jordan decomposition

In this section we recall some of the well-known properties of the multiplicative Jordan decomposition.

Definition 6.1.1. We call an element $g \in \mathrm{GL}(d,\mathbb{Q})$ *diagonalizable* if it is conjugate in $\mathrm{GL}(d,\mathbb{C})$ to a matrix in diagonal form. Further $u \in \mathrm{GL}(d,\mathbb{Q})$ is said to be *unipotent* if it is conjugate in $\mathrm{GL}(d,\mathbb{Q})$ to a matrix in upper unitriangular form.

Lemma 6.1.2. *Let $g \in \mathrm{GL}(d,\mathbb{Q})$. Then there exist unique $g_u, g_s \in \mathrm{GL}(d,\mathbb{C})$ such that g_u is unipotent, g_s is diagonalizable, and $g = g_u g_s = g_s g_u$. Moreover g_u and g_s both lie in $\mathrm{GL}(d,\mathbb{Q})$.*

Proof. See [38, Chapter 7]. □

Definition 6.1.3. Let $g \in \mathrm{GL}(d,\mathbb{Q})$. The decomposition $g = g_s g_u$ from Lemma 6.1.2 is called the *multiplicative Jordan decomposition* of g. We call g_s the *semisimple part* and g_u the *unipotent part* of g.

For given $g \in \mathrm{GL}(d,\mathbb{Q})$ we can compute g_s with the method in [7]. This also yields g_u by calculating $g_u = g_s^{-1} g$.

For a group $G \leq \mathrm{GL}(d,\mathbb{Q})$ we denote $G_s = \{g_s | g \in G\}$ and $G_u = \{g_u | g \in G\}$. The following theorem yields that if G is nilpotent then G_u and G_s are subgroups of $\mathrm{GL}(d,\mathbb{Q})$ that commute element-wise.

Theorem 6.1.4. *Let G be a nilpotent subgroup of $\mathrm{GL}(d,\mathbb{Q})$. Then G_u and G_s are subgroups of $\mathrm{GL}(d,\mathbb{Q})$, $G \leq G_u \times G_s$ and the maps $(\ )_s : G \to G_s$ and $(\ )_u : G \to G_u$ are homomorphisms.*

Proof. See [38, Chapter 7]. □

6.2 Splittable polycyclic groups

The key concept used by du Sautoy [16] to study collection functions in polycyclic groups is that of a splittable polycyclic group as introduced by Segal [38]. It will be explained in the following.

Let G be a polycyclic group with Fitting subgroup $\mathrm{Fitt}(G) = N$, i.e. N is the largest nilpotent normal subgroup of G. If N is a $\mathcal{T}$-group, then we can associate a Lie algebra $\mathcal{L}(N)$ to it, as outlined in Chapter 3. For $g \in G$ we denote by $\Phi(g)$ the Lie automorphism of $\mathcal{L}(N)$ that corresponds to the conjugation action of g on N. We identify in the following $\Phi(g)$ with its matrix representation with respect to an arbitrary fixed basis of $\mathcal{L}(N)$. Recall from §6.1 that we denote by $\Phi(g)_s$ the semisimple part of $\Phi(g)$.

Definition 6.2.1. Let G be a polycyclic group with Fitting subgroup N. Then G is said to be *splittable* if

(1) N is a $\mathcal{T}$-group and G/N is free abelian.
(2) $G = CN$ for some nilpotent $\mathcal{T}$-group $C \leq G$.
(3) $\mathrm{Log}(N)\Phi(C)_s = \mathrm{Log}(N)$, i.e. $\Phi(C)_s$ stabilises $\mathrm{Log}(N)$ as a set.

If G only satisfies (1) and (2) then G is called *almost splittable*.

Theorem 6.2.2. *Let G be a polycyclic group. Then G has a splittable subgroup of finite index.*

Proof. See [38, Chapter 7]. □

Definition 6.2.3. Let G be an almost splittable polycyclic group and $G = CN$ as in Definition 6.2.1. Let $(g_1N, \ldots, g_rN)$ be a basis for CN/N with $g_i \in C$ for $i = 1, \ldots, r$, and let $\mathcal{N} = (g_{r+1}, \ldots, g_s)$ be a Mal'cev basis of N. Then

$$\mathcal{G} = (g_1, \ldots, g_r, g_{r+1}, \ldots, g_s)$$

is a basis for $G = CN$; we call $\mathcal{G}$ a *canonical basis* for G.

Without loss of generality we can assume that $\Phi(C)_u$ stabilises the central series of N associated to $\mathcal{N}$ [16]. Also note that the list $(g_1, \ldots, g_r)$ can be extended to a Mal'cev basis of C, by Lemma 5.2.1.

Let H be a polycyclic group given by a polycyclic presentation. Eick describes in [17, Chapter 9] a practical algorithm to compute an almost splittable subgroup G of finite index in H. Her algorithm also yields a canonical basis for G.

As mentioned in the introduction of this chapter, du Sautoy [16] proved the following result about collection functions in splittable polycyclic groups. For complex valued functions $f_1, \ldots, f_z$ a *monomial* in $f_1, \ldots, f_z$ is a function of the form $\prod_{i=1}^{z} f_i^{a_i}$ where $a_i \in \mathbb{Z}$.

Theorem 6.2.4. *Let G be a splittable polycyclic group with a canonical basis $\mathcal{G} = (g_1, \ldots, g_r, g_{r+1}, \ldots, g_s)$. Let $\zeta = (\zeta_i) : \mathbb{Z}^s \times \mathbb{Z}^s \to \mathbb{Z}^s$ be the collection function of G with respect to $\mathcal{G}$ so that*

$$g^x g^y = g^{\zeta(x,y)}.$$

Then there exist a number field $\mathbb{F}$ and $w_{ij} \in \mathbb{F}$ where $1 \leq i \leq r$ and $1 \leq j \leq s - r$ such that ζ is given by a $\mathbb{F}$-linear combination of monomials in $x_1, \ldots, x_s, y_1, \ldots, y_s$ and $w_{ij}^{y_i}$ where $1 \leq i \leq r$ and $1 \leq j \leq s - r$.

Remark 6.2.5. du Sautoy calls the function ζ from the last theorem a polynomial over $\mathbb{F}$ in x, y and $w_{ij}^{y_i}$.

6.3 Computing collection functions

Let G be an almost splittable polycyclic group as defined in §6.2. Assume that G is given by a polycyclic presentation with respect to a canonical basis $\mathcal{G} = (g_1, \ldots, g_r, g_{r+1}, \ldots, g_s)$ as in Definition 6.2.3. In this section we show how to compute the collection functions $\zeta_1, \ldots, \zeta_s$ in $2s$ integer variables $x_1, \ldots, x_s, y_1, \ldots, y_s$ such that $g^x g^y = g^\zeta$.

By definition $\mathcal{N} = (g_{r+1}, \ldots, g_s)$ is a Mal'cev basis for the normal $\mathcal{T}$-group N in G. Further there is a $\mathcal{T}$-group C with $g_1, \ldots, g_r \in C$ such that $G = CN$ and $G/N = CN/N$ is a free lattice of rank r with basis $g_1N, \ldots, g_rN$. Throughout this section we will use the notation $c^x = g_1^{x_1} \cdots g_r^{x_r} \in C$ and $n^x = g_{r+1}^{x_{r+1}} \cdots g_s^{x_s} \in N$.

Example 6.3.1. Let $N = F_{2,2}$ be the free nilpotent of class two group on two generators. Then

$$N = \langle g_2, g_3, g_4 | g_3^{(g_2^{\pm 1})} = g_3 g_4^{\pm 1} \rangle$$

is a polycyclic presentation for N and $\mathcal{N} = (g_2, g_3, g_4)$ is a Mal'cev basis of N.

An automorphism of the free group F_2 naturally induces an automorphism of N. Let g_1 act on N like the automorphism of $F_2 = \langle f_1, f_2 \rangle$ that maps f_1 to f_2^{-1} and f_2 to $f_1 f_2^3$. This yields a polycyclically presented almost splittable group $G \cong \langle g_1 \rangle \ltimes N$ with canonical basis $\mathcal{G} = (g_1, g_2, \ldots, g_4)$; we have $C = \langle g_1 \rangle \cong \mathbb{Z}$ and $G/N = \langle g_1 N \rangle \cong \mathbb{Z}$.

6.3.1 The action of C on N

In this section we show how to compute symbolically the normal form of

$$(n^x)^{(c^y)}$$

with respect to $\mathcal{N}$.

Let $\mathcal{L}(N)$ be the Lie algebra associated to N. Recall that

$$\mathcal{B} = \{\mathrm{Log}(g_{r+1}), \ldots, \mathrm{Log}(g_s)\}$$

is a basis for $\mathcal{L}(N)$ and that for $g \in G$ we denote by $\Phi(g)$ the Lie automorphism of $\mathcal{L}(N)$ that corresponds to the conjugation action of g on N. In the following we identify $\Phi(g)$ with its matrix representation with respect to $\mathcal{B}$.

According to Theorem 3.3.5 we have that

$$(n^x)^{(c^y)} = \mathrm{Exp}(\mathrm{Log}(n^x)\Phi(c^y)).$$

By Lemma 4.3.1 the functions Log and Exp can be described by polynomials, and we can therefore compute symbolically with them; these polynomials can be computed as described in §4.3. It remains to get a symbolic expression for $\Phi(c^y)$ in the integer variables $y_1, \ldots, y_s$. For this purpose, we will use the multiplicative Jordan decomposition $\Phi(c^y) = \Phi(c^y)_s \Phi(c^y)_u$ as defined in §6.1.

Example 6.3.2. Let G be as in Example 6.3.1. Let $l_i = \mathrm{Log}(g_i)$ for $i = 2, 3, 4$ and $\overline{\alpha}_2, \overline{\alpha}_3, \overline{\alpha}_4$ be defined as in Lemma 4.3.1 so that

$$\mathrm{Log}(g_2^{e_2} g_3^{e_3} g_4^{e_4}) = \overline{\alpha}_2 l_2 + \overline{\alpha}_3 l_3 + \overline{\alpha}_4 l_4.$$

A straightforward computation, as shown in Chapter 4, yields

$$\mathcal{L}(N) = \langle l_2, l_3, l_4 | [l_2, l_3] = l_4 \rangle,$$

and $\overline{\alpha}_2 = e_2$, $\overline{\alpha}_3 = e_3$ and $\overline{\alpha}_4 = -\frac{1}{2} e_2 e_3 + e_4$.

Further we have that

$$\Phi(g_1) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 3 & -3/2 \\ 0 & 0 & 1 \end{pmatrix}.$$

Semisimple action

Theorem 6.3.3. *The entries of $\Phi(c^y)_s$ are $\mathbb{F}$-linear combinations of a finite number of expressions $\omega_{ij}^{y_i}$, where $\mathbb{F}$ is some number field and $\omega_{ij} \in \mathbb{F}$.*

Proof. We have that $\Phi(c^y)_s = \Phi(g_1)_s{}^{y_1} \cdots \Phi(g_r)_s{}^{y_r}$, because $(\ )_s : C \to C_s$ is a homomorphism. By definition every $\Phi(g_i)_s$ is diagonalizable. Furthermore, since $\Phi(C)_s$ is abelian by the proof of [38, Chapter 7, Theorem 1], the elements $\Phi(g_1)_s, \ldots, \Phi(g_r)_s$ are simultaneously diagonalizable.

Let $\mathbb{F}$ be the smallest number field that contains all eigenvalues of the matrices $\Phi(g_1)_s, \ldots, \Phi(g_r)_s$. Denote by T the base change matrix over $\mathbb{F}$ so that $\Phi(g_i)_s^T = T^{-1} \Phi(g_i)_s T$ is in diagonal form with diagonal entries $\omega_{i1}, \ldots, \omega_{il} \in \mathbb{F}$ for $i = 1, \ldots, r$, where l is the dimension of $\mathcal{L}(N)$. Since

$$\Phi(g_i)_s{}^{y_i} = ((\Phi(g_i)_s{}^T)^{y_i})^{T^{-1}} \tag{6.1}$$

the entries of $\Phi(g_i)_s{}^{y_i}$ are $\mathbb{F}$-linear combinations of $\omega_{i1}^{y_i}, \ldots, \omega_{il}^{y_i}$. We deduce that the entries of $\Phi(c^y)_s$ are $\mathbb{F}$-linear combinations of the $\omega_{ij}^{y_i}$ where $i = 1, \ldots, r$ and $j = 1, \ldots, l$. □

Remark 6.3.4. A matrix T such that ${\Phi(g_i)_s}^T$ is in diagonal form for $i = 1, \ldots, r$ can be computed as follows: by [17, Lemma 5.11] we have that a random rational linear combination of the elements of $\mathcal{A} = \{\Phi(g_1)_s, \ldots, \Phi(g_r)_s\}$ yields a primitive element α of the $\mathbb{Q}$-algebra $\mathbb{Q}[\mathcal{A}]$ with high probability. Let $\mathbb{F}$ be the splitting field of the minimal polynomial of α. Standard linear algebra now yields a matrix T over $\mathbb{F}$ such that α^T is in diagonal form. As a consequence ${\Phi(g_i)_s}^T$ is in diagonal form for $i = 1, \ldots, r$.

Example 6.3.5. We continue Example 6.3.2. The matrix $\Phi(g_1)$ is diagonalizable and thus we have $\Phi(g_1)_s = \Phi(g_1)$ and $\Phi(g_1)_u = 1$. The minimal polynomial of $\Phi(g_1)$ is $x^3 - 4x^2 + 4x - 1 = (x-1)(x^2 - 3x + 1)$ and thus we take $\mathbb{F} = \mathbb{Q}(\theta)$ where θ is a root of $x^2 - 3x + 1$. Computing the eigenspaces yields the base change matrix

$$T = \begin{pmatrix} -3/2 & 6/5 - 3/10\,\theta & 3/10 + 3/10\,\theta \\ 3/2 & -3/10 - 3/10\,\theta & -6/5 + 3/10\,\theta \\ 1 & 0 & 0 \end{pmatrix}.$$

Using equation (6.1) we deduce that $\Phi(g_1)^{y_1} = {\Phi(g_1)_s}^{y_1}$ is equal to

$$\begin{pmatrix} \alpha_1{\omega_{12}}^{y_1} + \alpha_2{\omega_{13}}^{y_1} & \alpha_3{\omega_{12}}^{y_1} - \alpha_3{\omega_{13}}^{y_1} & \alpha_4{\omega_{12}}^{y_1} + \alpha_5{\omega_{13}}^{y_1} + 3/2 \\ -\alpha_3{\omega_{12}}^{y_1} + \alpha_3{\omega_{13}}^{y_1} & \alpha_3{\omega_{12}}^{y_1} + \alpha_1{\omega_{13}}^{y_1} & -\alpha_5{\omega_{12}}^{y_1} - \alpha_4{\omega_{13}}^{y_1} + 3/2 \\ 0 & 0 & 1 \end{pmatrix}$$

where $\omega_{12} = \theta$, $\omega_{13} = 3 - \theta$, $\alpha_1 = 7/5 - 3/5\,\theta$, $\alpha_2 = -2/5 + 3/5\,\theta$, $\alpha_3 = 3/5 - 2/5\,\theta$, $\alpha_4 = 6/5 - 3/10\,\theta$ and $\alpha_5 = 3/10 + 3/10\,\theta$.

Theorem 6.3.6. *The elements $\omega_{ij} \in \mathbb{F}$ from Theorem 6.3.3 are contained in the group of units of the maximal order of $\mathbb{F}$.*

Proof. By [38, Chapter 6] the additive group $\mathbb{Z}\,\mathrm{Log}(N)$ is free abelian of finite rank and spans $\mathcal{L}(N)$ over $\mathbb{Q}$. Thus there exists a $\mathbb{Z}$-basis $\mathcal{B}$ of $\mathbb{Z}\,\mathrm{Log}(N)$ which is also a $\mathbb{Q}$-basis for $\mathcal{L}(N)$. By Theorem 3.3.6, the lattice $\mathbb{Z}\,\mathrm{Log}(N)$ is invariant under $\Phi(C)$. Thus with respect to the basis $\mathcal{B}$ we have that $\Phi(C) \leq \mathrm{GL}(d, \mathbb{Z})$.

Recall that $(g_1N, \ldots, g_rN)$ is a basis for CN/N with $g_i \in C$ for $i = 1, \ldots, r$. We denote by $\chi_{g_i}, \chi_{g_i^{-1}}$ the minimal polynomial of $\Phi(g_i), \Phi(g_i^{-1})$ over $\mathbb{Q}$. Since $\Phi(g_i)$ is conjugate to an element in $\mathrm{GL}(d, \mathbb{Z})$ we deduce that $\chi_{g_i}, \chi_{g_i^{-1}} \in \mathbb{Z}[x]$. Since ω_{ij} is by definition an eigenvalue of $\Phi(g_i)$ it follows that ω_{ij} is in the maximal order $\mathcal{O}$ of $\mathbb{F}$; further ω_{ij}^{-1} is an eigenvalue of $\Phi(g_i^{-1})$ and thus $\omega_{ij}^{-1} \in \mathcal{O}$. Therefore ω_{ij} lies in the group of units of $\mathcal{O}$. □

Unipotent action

Lemma 6.3.7. *Let $u \in \mathrm{GL}(d, \mathbb{Q})$ be unipotent and α be a rational variable. Then the entries of u^α are rational polynomials in α and the entries of u.*

Proof. By §3.2 we have that $u^\alpha = \exp(\alpha \log(v))$ where log and exp are given by

$$\begin{aligned} \log &: g \mapsto (g-1) - \frac{1}{2}(g-1)^2 + \cdots + \frac{(-1)^d}{(d-1)}(g-1)^{d-1} \\ \exp &: x \mapsto 1 + x + \frac{1}{2}x^2 + \cdots + \frac{1}{(d-1)!}x^{d-1}. \end{aligned}$$

Thus the claim follows. □

Since $(\ )_u : C \to C_u$ is a homomorphism, we deduce that $\Phi(c^y)_u = \Phi(g_1)_u{}^{y_1} \cdots \Phi(g_r)_u{}^{y_r}$. The matrix $\Phi(g_i)_u$ is unipotent and therefore the entries of $\Phi(g_i)_u^{y_i}$ are polynomials in the entries of $\Phi(g_i)_u$ and y_i. Thus the entries of $\Phi(c^y)_u$ are polynomials in the entries of $\Phi(g_1)_u, \ldots, \Phi(g_r)_u$ and $y_1, \ldots, y_r$. Using the constructive proof of Lemma 6.3.7, we can compute the entries of $\Phi(c^y)_u$.

Remark 6.3.8. As a consequence of §6.3.1 and §6.3.1 the entries of the matrix $\Phi(c^y)$ are polynomials over $\mathbb{F}$ in x, y and $\omega_{ij}^{y_i}$ where $\mathbb{F}$ and ω_{ij} are as in Theorem 6.3.3. Since Log and Exp can be described by polynomials we deduce that the exponents of $\mathrm{nf}_\mathcal{N}((n^x)^{(c^y)})$ are polynomials over $\mathbb{F}$ in x, y and $\omega_{ij}^{y_i}$.

6.3.2 Converting tails

In this section we describe how to compute symbolically $\mathrm{nf}_\mathcal{G}(c^x c^y)$, i.e. the normal form of $c^x c^y$ with respect to $\mathcal{G}$.

Since CN/N is free abelian we have $c^x c^y = c^{x+y} t$ for some tail $t \in C \cap N$. Thus we only need to compute the exponent vector of t with respect to the Mal'cev basis $\mathcal{N}$ of N.

According to §6.2 there is a Mal'cev basis $\mathcal{C} = (g_1, \ldots, g_r, c_{r+1}, \ldots, c_k)$ of C. Using Deep Thought we can compute symbolically the exponents $a_{r+1}, \ldots, a_k$ of the normal form of $t = c_{r+1}^{a_{r+1}} \cdots c_k^{a_k}$ with respect to $\mathcal{C}$.

By computing $\mathrm{nf}_\mathcal{N}(c_{r+1}), \ldots, \mathrm{nf}_\mathcal{N}(c_k)$ and using symbolic collection for multiplication and powering with respect to $\mathcal{N}$, we can compute the normal form of t with respect to $\mathcal{N}$. If $t = g_{r+1}^{b_{r+1}} \cdots g_s^{b_s}$, then the exponents $b_{r+1}, \ldots, b_s$ are rational polynomials in the variables x_i, y_j where $i, j = 1, \ldots, r$, because we only used symbolic collection in $\mathcal{T}$-groups for their computation.

6.3.3 The algorithm

We now summarise the different steps to compute the collection functions $\zeta_1, \ldots, \zeta_s$. Since

$$\begin{aligned} g^x g^y &= c^x n^x c^y n^y \\ &= c^x c^y (n^x)^{(c^y)} n^y \end{aligned}$$

we have the following algorithm.

ComputeCollectionFunctions($\mathcal{G}$)

1: compute symbolically $c^{x+y} n^\alpha := \mathrm{nf}_{\mathcal{G}}(c^x c^y)$ (§6.3.2).
2: compute symbolically $n^\beta := \mathrm{nf}_{\mathcal{N}}((n^x)^{(c^y)})$ (§6.3.1).
3: compute symbolically $n^\gamma := \mathrm{nf}_{\mathcal{N}}(n^\alpha n^\beta n^y)$ (Deep Thought).
4: return $\zeta_i = x_i + y_i$ for $i = 1, \ldots, r$ and $\zeta_{r+j} = \gamma_j$ for $j = 1, \ldots, s - r$.

Remark 6.3.9. As a consequence of the last algorithm we can reprove du Sautoy's Theorem 6.2.4 and see that the collection functions ζ_i are polynomials over some number field $\mathbb{F}$ in x, y and a finite number of expression $\omega_{ij}^{y_i}$ where $\omega_{ij} \in \mathbb{F}$; in addition we show that the ω_{ij} are units in the maximal order of $\mathbb{F}$.

Proof of Theorem 6.2.4. In §6.3.2 we saw that the exponents of n^α are polynomials in x and y. By Remark 6.3.8, the exponents of n^β are polynomials over some number field $\mathbb{F}$ in x, y and a finite number of expressions $\omega_{ij}^{y_i}$ where $\omega_{ij} \in \mathbb{F}$. Since multiplication with respect to the Mal'cev basis $\mathcal{N}$ can be described by polynomials we see that exponents of n^γ are polynomials over $\mathbb{F}$ in x, y and $\omega_{ij}^{y_i}$. Thus the statement of Theorem 6.2.4 follows.

By Theorem 6.3.6, we see that the ω_{ij} are contained in the group of units of the maximal order of $\mathbb{F}$. □

Example 6.3.10. We continue Example 6.3.5. Recall that $\mathbb{F} = \mathbb{Q}(\theta)$ where θ is a root of $x^2 - 3x + 1$ and that we denote $\omega_{12} = \theta$, $\omega_{13} = 3 - \theta$. Using the algorithm ComputeCollectionFunctions we deduce that with respect to the

canonical basis $\mathcal{G}$ of G the collection functions are of the form

$$\begin{aligned}
\zeta_1 &= x_1 + y_1 \\
\zeta_2 &= (\alpha_1 x_2 - \alpha_3 x_3)\omega_{12}{}^{y_1} + (\alpha_2 x_2 + \alpha_3 x_3)\omega_{13}{}^{y_1} + y_2 \\
\zeta_3 &= (\alpha_3 x_2 + \alpha_2 x_3)\omega_{12}{}^{y_1} - (\alpha_3 x_2 - \alpha_1 x_3)\omega_{13}{}^{y_1} + y_3 \\
\zeta_4 &= (\alpha_4 x_2^2 - 1/5\, x_2 x_3 + 1/10\, \theta x_3^2)\omega_{12}{}^{2y_1} \\
&\quad +(1/10\, \theta x_2^2 - 1/5\, x_2 x_3 + \alpha_4 x_3^2)\omega_{13}{}^{2y_1} \\
&\quad +(\alpha_3 x_2 y_2 + \alpha_2 y_2 x_3 + \alpha_5 x_2 + \alpha_6 x_3)\omega_{12}{}^{y_1} \\
&\quad +(-\alpha_3 x_2 y_2 + \alpha_1 y_2 x_3 + -\alpha_6 x_2 + -\alpha_5 x_3)\omega_{13}{}^{y_1} \\
&\quad -3/10\, x_2^2 + 4/10\, x_2 x_3 - 3/10\, x_3^2 - 3/2\, x_2 + 3/2\, x_3 + x_4 + y_4
\end{aligned}$$

where $\alpha_1 = 7/5 - 3/5\,\theta$, $\alpha_2 = -2/5 + 3/5\,\theta$, $\alpha_3 = 3/5 - 2/5\,\theta$, $\alpha_4 = 3/10 - 1/10\,\theta$, $\alpha_5 = 6/5 - 3/10\,\theta$ and $\alpha_6 = -3/10 - 3/10\,\theta$.

Remark 6.3.11. The function $\zeta : \mathbb{Z}_s \times \mathbb{Z}_s \to \mathbb{Z}_s$ computed by the algorithm ComputeCollectionFunctions can be used to compute the inversion function $i : \mathbb{Z}_s \to \mathbb{Z}_s$ that describes inversion in G, i.e. $g^{i(x)} = (g^x)^{-1}$.

Let $g^x = c^x n^x \in G$. We have $(n^x)^{-1} = \mathrm{Exp}(-\mathrm{Log}(n^x))$ and thus the exponents of $\mathrm{nf}_{\mathcal{N}}((n^x)^{-1})$ are polynomials in x. The same holds for the exponents of $\mathrm{nf}_{\mathcal{C}}((c^x)^{-1})$ and thus for $\mathrm{nf}_{\mathcal{G}}((c^x)^{-1})$. Since $(c^x n^x)^{-1} = (n^x)^{-1}(c^x)^{-1}$ we can use the functions $\zeta_1, \ldots, \zeta_s$ to combine these normal forms to compute the normal form of $(c^x n^x)^{-1}$.

Remark 6.3.12. The algorithm ComputeCollectionFunctions has been implemented in the computer algebra system GAP [41]. The code is publicly available as part of the GAP-package Guarana [2]. Let G be an almost splittable polycyclic group and denote hl the Hirsch length of G and c the nilpotency class of the Fitting subgroup of G. Using the implementation in Guarana we were able to compute collection functions of examples of almost splittable polycyclic groups with $hl \leq 36$ and $c \leq 6$. For example for the group $G(\mathrm{Tr}_6(\mathcal{O}_1)$ from §5.5, which has $hl = 36$ and $c = 5$, it took 100 seconds to compute the collection functions on a 3 gigahertz Pentium 4 processor; all groups were given by a polycyclic presentation with respect to a canonical basis.

6.4 Applications

There are several ways in which the collection functions $\zeta_1, \ldots, \zeta_s$ computed by the algorithm ComputeCollectionFunctions of §6.3.3 can be used.

6.4.1 Collection

A natural application is to use $\zeta_1, \ldots, \zeta_s$ for collection in almost splittable polycyclic groups. Once these functions are computed, the computations of the normal form of $g^x g^y$ can be reduced to the evaluation of $\zeta(x, y)$. Experiments with our examples showed that this yields a method that is faster then standard methods such as "Collection from the left" if the exponents x, y are big, i.e. if they lie in the hundreds. However the method Mal'cev collection, as described in Chapter 5, which has a cheaper set up, is faster and therefore preferable.

6.4.2 pro-p-completions

The *pro-p-completion* of a polycyclic group G is the inverse limit of the system of finite quotients of G which have p-power order. It is well-known that in a $\mathcal{T}$-group N the polynomials defining the group operations with respect to a Mal'cev basis extend to the p-adic integers $\mathbb{Z}_p$ to define the pro-p-completion of N. In [16] du Sautoy showed that every splittable polycyclic group G has a normal subgroup $G(p)$ of finite index, whose pro-p-completion can be obtained in the same way.

In this section we give a brief summary of du Sautoy's result. Then we show how to compute $G(p)$; together with the algorithm from Section 6.3.3, this allows us to construct the pro-p-completion of $G(p)$ algorithmically.

Let G be a splittable polycyclic group with Fitting subgroup N and let $L = \mathbb{Z}\,\mathrm{Log}(N)$ be the $\mathbb{Z}$-lattice generated by $\mathrm{Log}(N)$. For each prime p we define the *p-canonical subgroup* $G(p)$ of G to be HN where H is the centraliser of the induced action of G on L/pL. By [16] $G(p)$ is a normal splittable polycyclic subgroup of finite index in G.

Let $\mathcal{G} = (g_1, \ldots, g_s)$ be a canonical basis of $G(p)$ and let $\zeta : \mathbb{Z}^s \times \mathbb{Z}^s \to \mathbb{Z}^s$ and $i : \mathbb{Z}^s \to \mathbb{Z}^s$ be the functions defining multiplication and inversion in $G(p)$ with respect to $\mathcal{G}$. By [16, Theorem 3.5], these function are extendible to p-adic valued, locally analytic functions $\zeta : \mathbb{Z}_p^s \times \mathbb{Z}_p^s \to \mathbb{Z}_p^s$; $i : \mathbb{Z}_p^s \to \mathbb{Z}_p^s$. We define $G(p)^{\mathbb{Z}_p}$ to be the set of formal products $\{g^x | x \in \mathbb{Z}_p^s\}$, where $g^x = g_1^{x_1} \cdots g_s^{x_s}$, and define multiplication and inversion on $G(p)^{\mathbb{Z}_p}$ using ζ and i. For a proof of the following result see [16, Theorem 3.7].

Theorem 6.4.1. *$G(p)^{\mathbb{Z}_p}$ is isomorphic to the pro-p-completion of $G(p)$; in particular $G(p)^{\mathbb{Z}_p}$ does not depend on a choice of canonical basis of $G(p)$.*

Computing $G(p)$

Once a $\mathbb{Z}$-basis for $L = \mathbb{Z}\operatorname{Log}(N)$ is given, it is straightforward to obtain an integral matrix representation of the induced action of G on L. Together with the algorithm in [4, Section 4], this yields a method to compute a canonical basis for the p-canonical subgroup $G(p)$. It remains to describe a method to compute L.

Computing a $\mathbb{Z}$-basis for $\mathbb{Z}\operatorname{Log}(N)$

Let N be a $\mathcal{T}$-group given by a polycyclic presentation with respect to a Mal'cev basis $\mathcal{N} = (n_1, \ldots, n_l)$. In the following we describe an algorithm to compute a $\mathbb{Z}$-basis for $\mathbb{Z}\operatorname{Log}(N)$. We then comment on the different steps and give a proof of correctness.

For given $(e_1, \ldots, e_l) \in \mathbb{Z}^l$ we denote $n^e = n_1^{e_1} \cdots n_l^{e_l}$. Recall that

$$\{\operatorname{Log}(n_1), \ldots, \operatorname{Log}(n_l)\}$$

is basis for the Lie algebra $\mathcal{L}(N)$. In this section we identify an element $\sum_{i=1}^{l} a_i \operatorname{Log}(n_i) \in \mathcal{L}(N)$ with the coefficient vector $a = (a_1, \ldots, a_l)$.

ComputeZBasis(N **)**

1: let $e = (e_1, \ldots, e_l)$ be a vector of integer variables and set $\mathcal{B} = \{\}$.
2: compute polynomials $x_{11}(e), \ldots, x_{1l}(e) \in \mathbb{Q}[e]$ such that $x_1(e) = \operatorname{Log}(n^e)$.
3: **for** $i = 1, \ldots, l$ **do**
4: write $x_{ii}(e) = q_i(e)/\pi_i$ where $q_i \in \mathbb{Z}[e]$ and $\pi_i \in \mathbb{N}$.
5: determine $\gamma_i \in \mathbb{N}$ such that $\mathbb{Z}\gamma_i = \mathbb{Z}\{q_i(f_j) | f_j \in \mathbb{Z}^l\}$.
6: write $\gamma_i = \sum_j \alpha_j q_i(f_j)$ where $\alpha_j \in \mathbb{Z}$, $f_j \in \mathbb{Z}^l$.
7: set $y_i = \sum_j \alpha_j x_i(f_j) \in \mathbb{Q}^l$.
8: add y_i to $\mathcal{B}$.
9: set $x_{i+1}(e) = x_i(e) - \frac{q_i(e)}{\gamma_i} y_i$.
10: **end for**
11: return $\mathcal{B}$.

Remark 6.4.2. For step 2 we can use the method from §4.3 to compute the polynomials x_{1i}. For step 5 and 6 note that $\mathbb{Z}\{q_i(f_j) | f_j \in \mathbb{Z}^l\}$ is a subgroup of $(\mathbb{Z}, +)$ and thus cyclic. A generator γ_i can be computed using modular arithmetic: let $0 \neq \delta_1 = q_i(f_1)$ for some $f_1 \in \mathbb{Z}^l$. If $q_i(f_k) = 0 \mod \delta_1$ for all $f_k \in \mathbb{Z}^l$, then set $\gamma_i = \delta_1$ (note that testing this condition only involves finitely many f_k); otherwise we can determine $0 \neq \delta_2 = q_i(f_2) < \delta_1$ for some $f_2 \in \mathbb{Z}^l$ and recurse.

Theorem 6.4.3. *The algorithm* ComputeZBasis(N) *returns a* $\mathbb{Z}$*-basis for* $\mathbb{Z}\,\mathrm{Log}(N)$.

Proof. (1): First we prove by induction on i that $x_i(\mathbb{Z}^l) \subseteq \mathbb{Z}\,\mathrm{Log}(N)$ for $i = 1, \ldots, l$; as a consequence $y_i = \sum_j \alpha_j x_i(f_j) \in \mathbb{Z}\,\mathrm{Log}(N)$. For $i = 1$ we have $x_1(f_j) = \mathrm{Log}(n^{f_j}) \in \mathrm{Log}(N)$ and thus $x_1(\mathbb{Z}^l) \subseteq \mathbb{Z}\,\mathrm{Log}(N)$. Now assume that $x_i(\mathbb{Z}^l) \subseteq \mathbb{Z}\,\mathrm{Log}(N)$. Since $x_{i+1}(e) = x_i(e) - \frac{q_i(e)}{\gamma_i} y_i$ and $\frac{q_i(\mathbb{Z}^l)}{\gamma_i} \subseteq \mathbb{Z}$ and $y_i \in \mathbb{Z}\,\mathrm{Log}(N)$ we deduce that $x_{i+1}(\mathbb{Z}^l) \subseteq \mathbb{Z}\,\mathrm{Log}(N)$.

(2): Second we show, again by induction on i, that $x_{ik} = 0$ if $k < i$ for $i = 1, \ldots, l+1$; as a consequence the same holds for y_i. For $i = 1$ this is clearly true. Now assume that the claim is true for x_i. Since $x_{i+1}(e) = x_i(e) - \frac{q_i(e)}{\gamma_i} y_i$ and

$$x_{ii}(e) = \frac{q_i(e)}{\pi_i} = \frac{q_i(e)}{\gamma_i}\frac{\gamma_i}{\pi_i} = \frac{q_i(e)}{\gamma_i} y_{ii}$$

it follows that $x_{i+1i}(e) = 0$.

(3): As a consequence of (1), $\mathcal{B} \subseteq \mathbb{Z}\,\mathrm{Log}(N)$. By (2) we see that $x_{l+1} = 0$. We deduce that $\mathrm{Log}(n^e) = x_1(e) = \sum_{i=1}^{l} \frac{q_i(e)}{\gamma_i} y_i$. Thus every element of $\mathrm{Log}(N)$ is $\mathbb{Z}$-linear combination of the y_i and so $\mathcal{B} = \{y_1, \ldots, y_l\}$ is a generating set for $\mathbb{Z}\,\mathrm{Log}(N)$. Since l is the dimension of $\mathcal{L}(N) = \mathbb{Q}\,\mathrm{Log}(N)$ the elements of $\mathcal{B}$ are linearly independent. Thus $\mathcal{B}$ is a $\mathbb{Z}$-basis for $\mathbb{Z}\,\mathrm{Log}(N)$. □

Concluding remark

Let G be a splittable polycyclic group. We described a method to compute a canonical basis $\mathcal{G}$ for the the p-canonical subgroup $G(p)$. Using the algorithm in §6.3.3 we can compute the functions ζ and i which describe multiplication and inversion in $G(p)$ with respect to $\mathcal{G}$. This allows us to do computations in $G(p)^{\mathbb{Z}_p}$ which is isomorphic to the pro-p completion of $G(p)$ by Theorem 6.4.1.

Chapter 7

Alternatives beyond the Tits alternative

Let G be a finitely generated subgroup of $\mathrm{GL}(d, \mathbb{Q})$. A famous theorem, due to Tits, states that G is either virtually soluble or contains a non-abelian free subgroup [42]; it is called the Tits alternative. Beals [10] and Ostheimer [35] describe algorithms to decide the Tits alternative.

It is well-known that membership in G is in general undecidable; the reason for this lies precisely in the possible occurrence of a non-abelian free subgroup [30].

However if G happens to be virtually soluble then it seems to be more open to algorithmic explorations. For example if G is virtually polycyclic then it is well known how to test membership in G [4, 35]. Thus the Tits alternative tells us, in a way, to what extent G is suitable for further algorithmic investigations.

In this chapter we describe an algorithm for checking whether G is polycyclic. Such a method has not been available so far. More precisely we describe algorithms for

(1) testing whether G is polycyclic (or virtually polycyclic),
(2) testing whether G is nilpotent (or virtually nilpotent).

Our methods for (1) and (2) rely on an algorithm for testing whether a virtually polycyclic group $G \leq \mathrm{GL}(d, \mathbb{Q})$ is conjugate to a subgroup of $\mathrm{GL}(d, \mathbb{Z})$. We describe such an algorithm in Section 7.3. An alternative method for this purpose can be found in [8].

Our solutions for the algorithms in (1) and (2) are closely related to each other. They heavily rely on an application of the Mal'cev correspondence for upper unitriangular matrix groups. Based on that, they reduce to some simple applications of linear algebra methods.

We implemented our algorithm for testing polycyclicity using the computer algebra system GAP [41]. In §7.9 we report and comment on this implementation and give runtimes for some example groups.

Our algorithms also apply to finitely generated subgroups of $\mathrm{GL}(d, K)$, where K is an algebraic number field, since such matrix groups can be considered as subgroups of $\mathrm{GL}(d[K : \mathbb{Q}], \mathbb{Q})$.

Remark 7.0.4. Since the content of this chapter is joined work with Bettina Eick [5], I want to clarify what my contribution to this project was.

At a conference in Warwick in August 2005, Bettina Eick gave a talk in which she showed that she could test polycyclicity of a finitely generated subgroup G of $\mathrm{GL}(d, \mathbb{Q})$ if she could test conjugacy into $\mathrm{GL}(d, \mathbb{Z})$ of certain induced actions of G.

After the conference I joined the project and showed, using hints of Bettina Eick and Gabriele Nebe, how to check whether a virtually polycyclic subgroup of $\mathrm{GL}(d, \mathbb{Q})$ conjugates into $\mathrm{GL}(d, \mathbb{Z})$. Later we found out that alternative methods for this purpose were described in [8].

Further I found out how the Mal'cev correspondence could be used to reprove Eick's result. This has led to an efficiency improvement and to a simplification of the mathematical outline of the algorithm. We then discovered jointly that minor modifications of the polycyclicity algorithm could be used to test (virtual) nilpotency.

7.1 Deciding the Tits' alternative

Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated and write $V = \mathbb{Q}^d$. In this section we briefly recall the method of [4] for testing whether G is soluble or virtually soluble, since we need various parts of it later.

7.1.1 Computing a semisimple series

Definition 7.1.1. The $\mathbb{Q}$-*algebra* $\mathbb{Q}[G]$ is the vector space spanned by G over $\mathbb{Q}$. A vector subspace $W \leq \mathbb{Q}^d$ is said to be a $\mathbb{Q}[G]$-*module* or a G-*module* if G acts via $\mathbb{Q}$-linear automorphism on V.

The vector space $V = \mathbb{Q}^d$ has naturally the structure of a G-module.

Definition 7.1.2. A G-module W is said to be *irreducible* if its only G-submodules are $\{0\}$ and W. Further W is said to be *semisimple* if it is the direct product of irreducible G-modules. A series

$$W = W_1 > \ldots > W_l > W_{l+1} = \{0\}$$

of G-submodules of W is called *semisimple* if W_i/W_{i+1} is semisimple as a G-module for $1 \leq i \leq l$.

In this subsection we briefly recall a method to determine a semisimple series for G.

Definition 7.1.3. The radical $Rad_G(V)$ is defined as the intersection of all maximal G-submodules in V.

Theorem 7.1.4. *$Rad_G(V)$ is the smallest G-submodule in V with a semisimple factor module. Further $Rad_G(V) = V Rad_G(\mathbb{Q}[G])$.*

Proof. See [31, Chapter 7]. □

Thus the determination of a semisimple series can be reduced to an iterated computation of radicals.

A method to compute the radical $Rad_G(V)$ has been introduced by L.E. Dickson in [15]. It uses the fact that $Rad_G(V) = V Rad_G(\mathbb{Q}[G])$.

7.1.2 The p-congruence subgroup

Since G is finitely generated, there exists a finite set π of primes such that $G \leq \mathrm{GL}(d, \mathbb{Q}_\pi)$, where $\mathbb{Q}_\pi$ is the set of all rational numbers $\frac{a}{b}$ with b divisible by primes in π only. Let $p > 2$ be a prime with $p \notin \pi$. We say that p is a *suitable prime* for G. Then the natural homomorphism $\psi_p : \mathbb{Q}_\pi \to \mathbb{F}_p$ extends to a homomorphism

$$\varphi_p : G \to \mathrm{GL}(d, \mathbb{F}_p)$$

defined by applying ψ_p to every entry in a matrix element of G. The kernel H of φ_p is called the *p-congruence subgroup* and the image I of φ_p is the *p-congruence image* of G. By construction, the group H has finite index in G. As G is finitely generated, this implies that H is finitely generated. Generators for H can be computed from generators for G using an orbit-stabilizer algorithm, since $H = Stab_G(B)$, where B is a basis of $\mathbb{F}_p^d$ and G acts via φ_p on $\mathbb{F}_p^d$. However, the resulting generating set for H is often too large to allow efficient computations. A usually significantly smaller set of normal subgroup generators for H can be determined from generators for G as described in [4].

7.1.3 Testing (virtual) solvability

The following theorem provides a characterisation of the finitely generated soluble or virtually soluble subgroups of $\mathrm{GL}(d,\mathbb{Q})$. This characterisation can be checked easily with available computational tools and thus it yields an algorithm for checking solvability and virtual solvability. If the group G acts on a module W, then $G_W \leq \mathrm{GL}(W)$ denotes the group induced by the action of G on W. A proof of the following theorem can be found in [4].

Theorem 7.1.5. *Let $G \leq \mathrm{GL}(d,\mathbb{Q})$ be finitely generated with p-congruence subgroup H and p-congruence image I where p is a suitable prime for G. Let $V = V_1 > \ldots > V_l > V_{l+1} = \{0\}$ be a semisimple series for G.*

a) G is virtually soluble if and only if $H_{V_i/V_{i+1}}$ is abelian for $1 \leq i \leq l$.

b) G is soluble if and only if G is virtually soluble and I is soluble.

7.1.4 Comparing classes of groups

Recall that by Corollary 2.4.5 a group G is polycyclic if and only if G is soluble and every subgroup of G is finitely generated.

Not every finitely generated soluble subgroup of $\mathrm{GL}(d,\mathbb{Q})$ is polycyclic, as the following example shows:

$$G = \langle \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \rangle.$$

The group G contains the normal subgroup $U = \{ \left(\begin{smallmatrix} 1 & \frac{a}{2^e} \\ 0 & 1 \end{smallmatrix}\right) \mid a \in \mathbb{Z}, e \in \mathbb{N}_0 \}$. The quotient G/U is infinite cyclic and U is abelian; thus G is soluble. However U is not finitely generated and hence G is not polycyclic.

7.2 The Mal'cev correspondence and finite generation

Let $U \leq \mathrm{Tr}_1(d,\mathbb{Q})$. In this section we explore the relationship between finite generation of U and certain properties of the Lie algebra $\mathbb{Q}\log(U)$.

Recall that $\mathrm{Aut}(U)$ acts on $\mathbb{Q}\log(U)$ by Theorem 3.3.6. Let $\phi_{\mathcal{B}} : \mathrm{Aut}(U) \to \mathrm{GL}(e,\mathbb{Q})$ describe this action with respect to a basis $\mathcal{B}$ of $\mathbb{Q}\log(U)$ and let $\phi = \phi_{\mathcal{B}}$ for some arbitrary, fixed basis $\mathcal{B}$.

Theorem 7.2.1. *Let $U \leq \mathrm{Tr}_1(d,\mathbb{Q})$ and $H \leq \mathrm{Aut}(U)$ such that*

$$U = \langle u_1, \ldots, u_l \rangle^H$$

for certain elements $u_1, \dots, u_l \in U$. *Let* $W \leq \mathrm{Tr}_0(d, \mathbb{Q})$ *be the Lie algebra generated by* $\log(u_1), \dots, \log(u_l)$. *Then* $\mathbb{Q}\log(U) = W^{\phi(H)}$.

Proof. Let $S = \langle u_1, \dots, u_l \rangle$. Since W is a Lie algebra, we have that $x * y \in W$ for all $x, y \in W$. Since $\log : \mathrm{Tr}_1(d, \mathbb{Q}) \to (\mathrm{Tr}_0(d, \mathbb{Q}), *)$ is an isomorphism, it follows that $\log(S) \subseteq W$ and therefore $\mathbb{Q}\log(S) \subseteq W$. On the other hand, $\log(u_i) \in \mathbb{Q}\log(S)$ for all i and so $W \subseteq \mathbb{Q}\log(S)$. This yields that $\mathbb{Q}\log(S) = W$.

An element $g \in S^H$ is of the form $g = u_{i_1}^{h_{i_1}} \cdots u_{i_l}^{h_{i_l}}$ for certain $h_{i_j} \in H$. Therefore $\log(g) = \log(u_{i_1})^{\phi(h_{i_1})} * \cdots * \log(u_{i_l})^{\phi(h_{i_l})}$ which is contained in $\mathbb{Q}\log(S)^{\phi(H)}$. It follows that $\mathbb{Q}\log(U) = \mathbb{Q}\log(S^H) \subseteq \mathbb{Q}\log(S)^{\phi(H)} \subseteq \mathbb{Q}\log(U)^{\phi(H)} = \mathbb{Q}\log(U)$. Thus $\mathbb{Q}\log(S)^{\phi(H)} = \mathbb{Q}\log(U)$. □

Theorem 7.2.1 yields that a basis for $\mathbb{Q}\log(U)$ can be computed if $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ is given as $U = \langle u_1, \dots, u_l \rangle^H$ for a finitely generated group $H \leq \mathrm{Aut}(U)$. For this purpose we determine $\log(u_1), \dots, \log(u_l)$ and compute iteratively a basis for the smallest vector space that contains these elements and is closed under taking Lie brackets and acting with the generators of H; this method is called a spinning algorithm. This yields a Lie algebra which is finite dimensional, since it is a subalgebra of the finite dimensional algebra $\mathrm{Tr}_0(d, \mathbb{Q})$, and hence the spinning algorithm terminates.

A similar approach could be considered for computing a generating set for U. However, the group U might not be finitely generated, even if it is finitely generated as an H-module, and in this case the spinning algorithm would not terminate.

Definition 7.2.2. A subgroup $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ is a *lattice group* if $\log(U)$ is closed under addition in $\mathrm{Tr}_0(d, \mathbb{Q})$. The group $\mathrm{Tr}_1(d, \mathbb{Q})$ is a lattice group, since $\log(\mathrm{Tr}_1(d, \mathbb{Q})) = \mathrm{Tr}_0(d, \mathbb{Q})$. For a subgroup $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ we define the *lattice hull* U^{lat} as the intersection of all lattice groups in $\mathrm{Tr}_1(d, \mathbb{Q})$ containing U.

Lemma 7.2.3.

a) If $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ *is finitely generated, then the additive group* $\mathbb{Z}\log(U)$ *is free abelian of finite rank and spans* $\mathbb{Q}\log(U)$ *over* $\mathbb{Q}$. *Furthermore,* U *has finite index in the lattice hull* U^{lat}.

b) If M *is a finitely generated subgroup of the additive group* $\mathrm{Tr}_0(d, \mathbb{Q})$, *then* $\langle \exp(M) \rangle$ *is a finitely generated subgroup of* $\mathrm{Tr}_1(d, \mathbb{Q})$.

Proof. For a) see [38, Chapter 6].
b) The group $\mathrm{Tr}_0(d, \mathbb{Q})$ is torsion-free. Thus the additive group M has a $\mathbb{Z}$-basis $\log(u_1), \dots, \log(u_l)$ for certain $u_1, \dots, u_l \in \mathrm{Tr}_1(d, \mathbb{Q})$. Let $U =$

$\langle u_1,\dots,u_l\rangle$. Since U has finite index in U^{lat}, it follows that U^{lat} is finitely generated and thus polycyclic. Since $\log(u_i)\in\log(U^{\mathrm{lat}})$ for $i=1,\dots,l$, we find that M is contained in the lattice $\log(U^{\mathrm{lat}})$. Therefore, $\exp(M)\subseteq U^{\mathrm{lat}}$ and thus $\langle\exp(M)\rangle$ is finitely generated. □

7.3 Checking conjugacy into $\mathrm{GL}(d,\mathbb{Z})$

Let $G\leq\mathrm{GL}(d,\mathbb{Q})$ be a virtually polycyclic group. In this section we exhibit an effective test to check whether G can be conjugated into $\mathrm{GL}(d,\mathbb{Z})$; that is, whether there exists an element $h\in\mathrm{GL}(d,\mathbb{Q})$ such that $G^h\leq\mathrm{GL}(d,\mathbb{Z})$. Note that not every polycyclic subgroup of $\mathrm{GL}(d,\mathbb{Q})$ conjugates into $\mathrm{GL}(d,\mathbb{Z})$ as the example $G=\langle(\frac{1}{2})\rangle$ shows.

As a first step towards this aim, we recall two well-known characterisations of groups which conjugate into $\mathrm{GL}(d,\mathbb{Z})$. For a subset M of a vector space V we denote by $\langle M\rangle_{\mathbb{Q}}$ and $\langle M\rangle_{\mathbb{Z}}$ its $\mathbb{Q}$-span and its $\mathbb{Z}$-span, respectively. We call a finitely generated abelian group $L\subseteq V$ a lattice. If $\langle L\rangle_{\mathbb{Q}}=V$ then L is said to be a full lattice in V. Further, we denote by $\mathbb{Z}[G]$ the subring of $M_d(\mathbb{Q})$ which is generated by the matrices in G. By a $\mathbb{Z}$-order in the matrix algebra $\mathbb{Q}[G]$ we mean a subring of $\mathbb{Q}[G]$ that is finitely generated as a $\mathbb{Z}$-module, contains the same identity as $\mathbb{Q}[G]$ and spans $\mathbb{Q}[G]$ over $\mathbb{Q}$. Therefore $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order in $\mathbb{Q}[G]$ if and only if $\mathbb{Z}[G]$ is finitely generated as an additive group.

Lemma 7.3.1. *The following properties are equivalent:*

a) G is conjugate to a subgroup of $\mathrm{GL}(d,\mathbb{Z})$.
b) There exists a full G-invariant lattice $L\leq V=\mathbb{Q}^d$ with $\langle L\rangle_{\mathbb{Q}}=V$.
c) $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order.

Proof. *a)* $\Rightarrow$ *c)*: Suppose that G is conjugate to a subgroup H of $\mathrm{GL}(d,\mathbb{Z})$. Since $M_d(\mathbb{Z})$ is finitely generated as an additive group we deduce that $\mathbb{Z}[H]$ is finitely generated as an additive group. Thus $\mathbb{Z}[G]$ is finitely generated as an additive group and therefore $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order.

c) $\Rightarrow$ *b)*: Suppose that $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order and let $B=\{b_1,\dots,b_s\}$ be a $\mathbb{Z}$-basis for $\mathbb{Z}[G]$. Then for all $g\in G$ there exist $a_{g,k}\in\mathbb{Z}$ such that $b_ig=\sum_{k=1}^s a_{g,k}b_k$ for $1\leq i\leq s$. Let $r_{i,j}$ be the j-th row of b_i. Then $r_{i,j}g=\sum_{k=1}^s a_{g,k}r_{k,j}$ follows. Thus $L=\langle r_{i,j}\mid 1\leq i\leq s,1\leq j\leq d\rangle_{\mathbb{Z}}$ is a G-invariant lattice. Further $\langle L\rangle_{\mathbb{Q}}=V$, because the rows of $1\in\mathbb{Z}[G]$ are linearly independent.

b) $\Rightarrow$ *a)*: Let L be a full G-invariant lattice. Then G acts on L as a subgroup of $\mathrm{GL}(d,\mathbb{Z})$. Since a basis for L is also a basis for V, it follows that G is conjugate to a subgroup of $\mathrm{GL}(d,\mathbb{Z})$. □

As a next step, we introduce an effective method to check whether $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order. We first consider the special case of a cyclic group. For $g \in \mathrm{GL}(d,\mathbb{Q})$ denote by χ_g the minimal polynomial of g.

Lemma 7.3.2. *Let $g \in \mathrm{GL}(d,\mathbb{Q})$ and $U = \langle g \rangle$. Then the following are equivalent:*

a) $\mathbb{Z}[U]$ is a $\mathbb{Z}$-order.
b) $\chi_g, \chi_{g^{-1}} \in \mathbb{Z}[x]$.
c) $\chi_g \in \mathbb{Z}[x]$ and χ_g has constant term ± 1.

Proof. *b)* $\Leftrightarrow$ *c)*: Let $\chi_g = x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0$. Then $\chi_{g^{-1}} = x^n + \alpha_1/\alpha_0\, x^{n-1} + \cdots + \alpha_{n-1}/\alpha_0\, x + 1/\alpha_0$. Thus $\chi_g, \chi_{g^{-1}} \in \mathbb{Z}[x]$ if and only if $\alpha_0 = \pm 1$.

a) $\Rightarrow$ *b)*: If $\mathbb{Z}[U]$ is a $\mathbb{Z}$-order, then U conjugates into $\mathrm{GL}(d,\mathbb{Z})$. As the minimal polynomial is invariant under conjugation, it follows that $\chi_g, \chi_{g^{-1}} \in \mathbb{Z}[x]$.

b) $\Rightarrow$ *a)*: Let $n = \deg \chi_g = \deg \chi_{g^{-1}}$. As χ_g and $\chi_{g^{-1}}$ are monic polynomials over $\mathbb{Z}$, it follows that $\{g^{-n+1}, \ldots, g^{-1}, 1, g, \ldots, g^{n-1}\}$ generates $\mathbb{Z}[U]$ as an additive group and hence $\mathbb{Z}[U]$ is a $\mathbb{Z}$-order. □

The following theorem yields a reduction to the case of cyclic groups.

Theorem 7.3.3. *Let $\{g_1, \ldots, g_n\}$ be a generating set of $G \leq \mathrm{GL}(d,\mathbb{Q})$ such that every element of g can be written as a collected word $g = g_1^{e_1} \ldots g_n^{e_n}$ with $e_1, \ldots, e_n \in \mathbb{Z}$. Then $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order if and only if $\mathbb{Z}[\langle g_i \rangle]$ is a $\mathbb{Z}$-order for $1 \leq i \leq n$.*

Proof. Write $U_i = \langle g_i \rangle$. If $\mathbb{Z}[G]$ is a $\mathbb{Z}$-order, then $\mathbb{Z}[U_i]$ is a $\mathbb{Z}$-order because a subgroup of a finitely generated abelian group is finitely generated. Thus it suffices to show the converse. Let $a_{i,1}, \ldots, a_{i,l_i}$ be a $\mathbb{Z}$-basis for $\mathbb{Z}[U_i]$. Then for every $g \in G$ there exist $\alpha_{ik} \in \mathbb{Z}$ where $1 \leq k \leq l_i$ with

$$\begin{aligned} g &= g_1^{e_1} \cdots g_n^{e_n} \\ &= \left(\sum_{j_1=1}^{l_1} \alpha_{1j_1} a_{1j_1} \right) \cdots \left(\sum_{j_n=1}^{l_n} \alpha_{nj_n} a_{nj_n} \right) \\ &= \sum_{j_1=1}^{l_1} \cdots \sum_{j_n=1}^{l_n} \alpha_{1j_1} \cdots \alpha_{nj_n} a_{1j_1} \cdots a_{nj_n}. \end{aligned}$$

Thus

$$S = \{a_{1j_1} \cdots a_{nj_n} | 1 \leq j_i \leq l_i\}$$

is a finite generating set for $\mathbb{Z}[G]$ as an additive group. □

As G is virtually polycyclic, there exists a polycyclic normal subgroup N of finite index in G. Let T be a transversal for N in G and let $R = (r_1, \ldots, r_l)$ be a polycyclic sequence for N. Then every element g in G can be written as $g = tr_1^{e_1} \cdots r_l^{e_l}$ for some $t \in T$ and $e_1, \ldots, e_l \in \mathbb{Z}$. Thus $T \cup R$ is a generating set for G which satisfies the hypothesis of Theorem 7.3.3. Hence we obtain the following corollary to Theorem 7.3.3 which provides an effective check whether G conjugates into $\mathrm{GL}(d, \mathbb{Z})$.

Corollary 7.3.4. *Let G be virtually polycyclic with normal polycyclic subgroup N of finite index. Let T be a transversal for N in G and let R be a polycyclic sequence for N. Then G is conjugate to a subgroup of $\mathrm{GL}(d, \mathbb{Z})$ if and only if $\chi_g, \chi_{g^{-1}} \in \mathbb{Z}[x]$ for every $g \in T \cup R$.*

7.4 Testing polycyclicity

In this section we introduce an effective method to test whether a finitely generated subgroup G of $\mathrm{GL}(d, \mathbb{Q})$ is polycyclic. Every polycyclic group is soluble. Thus as a first step in our method, we use the algorithm of Section 7.1.3 to check whether G is soluble. We then assume throughout that the considered group G is soluble.

Definition 7.4.1. Let $V = \mathbb{Q}^d$ and let $V = V_1 > \ldots > V_l > V_{l+1} = \{0\}$ be an arbitrary, fixed semisimple series of G. Then the centraliser of this series $U = \cap_{i=1}^{l} C_G(V_i/V_{i+1})$ is called the *unipotent radical* of G.

Remark 7.4.2. The unipotent radical of Definition 7.4.1 is the maximal normal unipotent subgroup of G. This is because every normal unipotent subgroup of G must centralise the semisimple factors of $V = V_1 > \ldots > V_l > V_{l+1} = \{0\}$.

Note that by choosing a basis for V exhibiting the semisimple series, we can assume that $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$.

The following lemma summarises some information of the structure of G and U which will be used throughout.

Lemma 7.4.3. *Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated and soluble and let U be the unipotent radical of G. Then U is nilpotent and G/U is polycyclic.*

Proof. As $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$, it follows that U is nilpotent. Let $V = V_1 > \ldots > V_l > V_{l+1} = \{0\}$ be a semisimple series underlying U. Then the factor G/U embeds into the direct product $G_{V_1/V_2} \times \ldots \times G_{V_l/V_{l+1}}$. Let H be a p-congruence subgroup of G as defined in §7.1.2 for some suitable prime p; recall that $[G : H] < \infty$. It follows from Theorem 7.1.5 that $H_{V_i/V_{i+1}}$ is

abelian. Thus $G_{V_i/V_{i+1}}$ is (finitely generated abelian)-by-(finite soluble) and hence G/U is polycyclic. □

The following theorem provides a characterisation for polycyclic rational matrix groups.

Theorem 7.4.4. *Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated and soluble and let U be the unipotent radical of G. Then G is polycyclic if and only if U is finitely generated.*

Proof. If G is polycyclic, then every subgroup of G is finitely generated and hence U is finitely generated. Conversely, if U is finitely generated, then U is polycyclic, because U is nilpotent by Lemma 7.4.3. As G/U is also polycyclic by Lemma 7.4.3, the result follows. □

As described in [4], we can compute a polycyclic presentation for G/U. By evaluating the relators of such a presentation, we obtain a finite set of normal subgroup generators for U, that is $U = \langle u_1, \ldots, u_l \rangle^G$ for certain $u_1, \ldots, u_l \in U$. By Theorem 7.4.4, it remains to check whether U is finitely generated.

We employ the Lie algebra $\mathbb{Q}\log(U)$ for this purpose. First, we note that a basis for the finite dimensional vector space $\mathbb{Q}\log(U)$ can be computed using the comment following Theorem 7.2.1. Let $e \in \mathbb{N}$ be the dimension of $\mathbb{Q}\log(U)$. The conjugation action of G on U induces a subgroup $H \leq \mathrm{Aut}(U)$. In turn, this subgroup H acts on $\mathbb{Q}\log(U)$ by Theorem 3.3.6. Let $\phi_{\mathcal{B}} : \mathrm{Aut}(U) \to \mathrm{GL}(e, \mathbb{Q})$ describe this action with respect to the basis $\mathcal{B}$ of $\mathbb{Q}\log(U)$ and let $\phi = \phi_{\mathcal{B}}$ for some arbitrary, fixed basis $\mathcal{B}$.

Our aim in the following is to show that we can read off from the action of G on $\mathbb{Q}\log(U)$ whether U is finitely generated. The following theorem is a first step in that direction.

Theorem 7.4.5. *Let $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ and $H \leq \mathrm{Aut}(U)$ such that we have $U = \langle u_1, \ldots, u_l \rangle^H$ for certain $u_1, \ldots, u_l \in U$. Then U is finitely generated if and only if $\phi(H)$ can be conjugated into $\mathrm{GL}(e, \mathbb{Z})$.*

Proof. Assume that U is finitely generated. By Lemma 7.2.3, the additive group $\mathbb{Z}\log(U)$ is free abelian of finite rank and spans $\mathbb{Q}\log(U)$ over $\mathbb{Q}$. Thus there exists a $\mathbb{Z}$-basis $\mathcal{B}$ for $\mathbb{Z}\log(U)$ which is also a $\mathbb{Q}$-basis for $\mathbb{Q}\log(U)$. By Theorem 3.3.6, the lattice $\mathbb{Z}\log(U)$ is invariant under the action of H. Hence $\phi_{\mathcal{B}}(H) \leq \mathrm{GL}(e, \mathbb{Z})$ and $\phi(H)$ can be conjugated into $\mathrm{GL}(e, \mathbb{Z})$.

Now assume that $\phi(H)$ can be conjugated into $\mathrm{GL}(e, \mathbb{Z})$. Let $\mathcal{B}$ be a basis of $\mathbb{Q}\log(U)$ such that $\phi_{\mathcal{B}}(H) \leq \mathrm{GL}(e, \mathbb{Z})$ and let L be the $\mathbb{Z}$-span of $\mathcal{B}$. Denote $W = \langle u_1, \ldots, u_l \rangle$. Then $\mathbb{Z}\log(W)$ is finitely generated by

Lemma 7.2.3 and hence there exists $z \in \mathbb{N}$ such that $\mathbb{Z}\log(W) \subseteq M := \frac{1}{z}L$. As $\phi_{\mathcal{B}}(H) \leq \mathrm{GL}(e,\mathbb{Z})$, the lattice M is invariant under the action of H. Therefore for all $u \in W$ and $h \in H$, it follows that $\log(u^h) = \log(u)^h \in M$. Thus $U = W^H \subseteq \langle \exp(M) \rangle$. By Lemma 7.2.3, the group $\langle \exp(M) \rangle$ is finitely generated. Hence U is finitely generated because subgroups of polycyclic groups are polycyclic. □

Let $\varphi : G \to \mathrm{GL}(e,\mathbb{Q})$ denote the action of G on $\mathbb{Q}\log(U)$ with respect to an arbitrary, fixed basis $\mathcal{B}$ of $\mathbb{Q}\log(U)$. Then Theorem 7.4.5 yields that the group U is finitely generated if and only if $\varphi(G)$ can be conjugated into $\mathrm{GL}(e,\mathbb{Z})$. Section 7.3 contains a method to check whether a polycyclic subgroup of $\mathrm{GL}(e,\mathbb{Q})$ conjugates into $\mathrm{GL}(e,\mathbb{Z})$. However, this method does not apply directly, as $\varphi(G)$ might not be polycyclic. The next theorem shows that the method of Section 7.3 generalises to the case considered here. For $g \in G$ we denote by $\chi_{\varphi(g)} \in \mathbb{Q}[x]$ the minimal polynomial of $\varphi(g)$.

Theorem 7.4.6. *Let $G \leq \mathrm{GL}(d,\mathbb{Q})$ be finitely generated and soluble and let U be the unipotent radical of G. Let $(g_1U,\ldots,g_nU)$ be a polycyclic sequence for G/U. Then G is polycyclic if and only if $\chi_{\varphi(g_i)}, \chi_{\varphi(g_i^{-1})} \in \mathbb{Z}[x]$ for $1 \leq i \leq n$.*

Proof. Suppose that G is polycyclic. Then U is finitely generated and thus $\varphi(G)$ can be conjugated into $\mathrm{GL}(e,\mathbb{Z})$ by Theorem 7.4.5. Thus $\chi_{\varphi(g_i)}$ and $\chi_{\varphi(g_i^{-1})}$ are contained in $\mathbb{Z}[x]$.

Conversely, suppose that $\chi_{\varphi(g_i)}, \chi_{\varphi(g_i^{-1})} \in \mathbb{Z}[x]$ for $1 \leq i \leq n$. Let $\mathbb{Q}\log(U) = L_1 > \cdots > L_{l+1} = \{0\}$ be a refinement of the upper central series of $\mathbb{Q}\log(U)$ to a $\mathbb{Q}[G]$-composition series. We use induction on l to show that $\varphi(G)$ can be conjugated into $\mathrm{GL}(e,\mathbb{Z})$. As U is finitely generated as a G-normal subgroup, this yields by Theorem 7.4.5 and Theorem 7.4.4 that G is polycyclic.

First consider the case $l = 1$. Then U acts trivially on $\mathbb{Q}\log(U)$ and thus $\varphi(G)$ is polycyclic with polycyclic sequence $(\varphi(g_1),\ldots,\varphi(g_n))$. Corollary 7.3.4 now yields that $\varphi(G)$ can be conjugated into $\mathrm{GL}(e,\mathbb{Z})$.

Now let $l > 1$. We assume by induction that there exists a basis $\mathcal{B}$ of $\mathbb{Q}\log(U)$ which exhibits $L_1 > L_2 > \{0\}$ and with respect to which G_{L_1/L_2} and G_{L_2} have integral matrix representations and thus are polycyclic. With respect to $\mathcal{B}$ every element $\varphi(g)$ is represented by a matrix of the form

$$\begin{pmatrix} \alpha(g) & \gamma(g) \\ & \beta(g) \end{pmatrix}$$

where $\alpha(g)$, respectively $\beta(g)$, are the representations of the action of g on L_1/L_2, respectively L_2. For $g,h \in G$, it follows that $\gamma(gh) = \alpha(g)\gamma(h) +$

IsPolycyclic(G **)**

1: test whether G is soluble and return false if this is not the case.
2: compute a pc-sequence $(g_1U, \ldots, g_nU)$ for G/U where U is the unipotent radical.
3: compute normal subgroup generators for U.
4: compute a basis $\mathcal{B}$ for the Lie algebra $\mathbb{Q}\log(U)$.
5: compute the induced action $\varphi(g_i)$ with respect to $\mathcal{B}$ for $1 \leq i \leq n$.
6: let $\chi_{\varphi(g_i)}$ be the minimal polynomial of $\varphi(g_i)$ for $1 \leq i \leq n$.
7: **if** $\chi_{\varphi(g_i)} \in \mathbb{Z}[x]$ and has constant term ± 1 for $1 \leq i \leq n$, **then**
8: return true
9: **else**
10: return false
11: **end if**

Table 7.1: As a result of §7.4 we get the above algorithm to test polycyclicity. The input G is a finitely generated subgroup of $\mathrm{GL}(d, \mathbb{Q})$.

$\gamma(g)\beta(h)$. Thus, since $\alpha(G), \beta(G)$ are integral matrix groups and by assumption G is finitely generated, we deduce that the denominators of the entries of $\gamma(G)$ are bounded. Since $\alpha(G)$ and $\beta(G)$ are polycyclic, it follows that $\varphi(G)$ is polycyclic. Therefore there exists a polycyclic sequence $(\varphi(g_1), \ldots, \varphi(g_n), \varphi(u_1), \ldots, \varphi(u_l))$ of $\varphi(G)$, where $u_1, \ldots, u_l \in U$. Since U is unipotent, the minimal polynomial of $\varphi(u_j^{\pm 1})$ is of the form $(x-1)^{m_j}$ for some $m_j \in \mathbb{N}$ and $1 \leq j \leq l$. By Corollary 7.3.4, $\varphi(G)$ can be conjugated into $\mathrm{GL}(e, \mathbb{Z})$. □

The results of this section yield an algorithm to test polycyclicity; it is displayed in Table 7.1.

7.5 Testing virtual polycyclicity

A variation of the method to test polycyclicity yields a method to determine whether a finitely generated subgroup of $\mathrm{GL}(d, \mathbb{Q})$ is virtually polycyclic. The following theorem characterises the virtually polycyclic groups in a computationally useful form.

Theorem 7.5.1. *Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated and virtually soluble, and let H be a p-congruence subgroup of G for some suitable prime p. Then G is virtually polycyclic if and only if H is polycyclic.*

Proof. If H is polycyclic, then G is virtually polycyclic, because $[G : H] < \infty$. If G is virtually polycyclic, then there exists a normal polycyclic subgroup K with $[G : K] < \infty$. Being a subgroup of K, the group $H \cap K$ is polycyclic. We have that $H/H \cap K \cong KH/K \leq G/K$ and thus $H/H \cap K$ is finite. Since H is soluble by Theorem 7.1.5, $H/H \cap K$ is soluble. Thus $H/H \cap K$ is polycyclic. Therefore H is polycyclic. □

Generators for a p-congruence subgroup H of G can be computed from a generating set of G as discussed in Section 7.1.2. Thus the method of Section 7.4 extends to testing virtual polycyclicity.

7.6 Testing nilpotency

Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated. In this section we describe a method to test whether G is nilpotent.

Methods for this purpose have been developed by Detinko and Flannery. In [14] they describe an algorithm for testing nilpotency of $G \leq \mathrm{GL}(d, K)$ where K is a finite field. Their algorithm for the case $K = \mathbb{Q}$ has not been published yet.

In the following we outline an alternative approach. This alternative shows that testing nilpotency is closely related to testing polycyclicity as in Section 7.4 and, further, the alternative extends to testing virtual nilpotency as shown in Section 7.7 below.

Every finitely generated nilpotent group is polycyclic. Hence as a first step to our algorithm we check whether the given group G is polycyclic using the method of Section 7.4. Thus we can assume in the following that G is polycyclic.

We characterise the nilpotent matrix groups among the polycyclic matrix groups. For this purpose we use the following notation.

Definition 7.6.1. If H is a group which acts by automorphisms on a group U, then H *acts nilpotently* on U if there exists a series of H-invariant normal subgroups of U such that H centralises every factor of the series.

If H acts by automorphisms on a Lie algebra L, then H *acts nilpotently* on L if there exists a series of H-invariant Lie subalgebras with H-central factors.

Lemma 7.6.2. *Let G be a polycyclic subgroup of $\mathrm{GL}(d, \mathbb{Q})$ and let U be the unipotent radical of G. Then G is nilpotent if and only if G/U is nilpotent and G acts nilpotently on U.*

Proof. If G is nilpotent then G/U is nilpotent. By intersecting a central series of G with U we see that G acts nilpotently on U.

Conversely assume that G/U is nilpotent and that G acts nilpotently on U. Thus there exists a G-central series of U. Since G/U is nilpotent we can extend this series to a central series of G and thus G is nilpotent. □

As side-results of the algorithm IsPolycyclic of Section 7.4, we have given normal subgroup generators for the unipotent radical U of G and a polycyclic sequence $\mathcal{G} = (g_1U, \ldots, g_kU)$ of G/U. We can use this to determine a polycyclic presentation for G/U and, based on that, we can test whether G/U is nilpotent with the methods in [17].

It remains to find a criterion which decides whether G acts nilpotently on U. As in the test for polycyclicity, one can use the action of G on the Lie algebra $\mathbb{Q}\log(U)$ for this purpose. The following theorem provides a first step towards proving this.

Theorem 7.6.3. *Let $U \leq \mathrm{Tr}_1(d, \mathbb{Q})$ and let $H \leq \mathrm{Aut}(U)$. Then H acts nilpotently on U if and only if H acts nilpotently on $\mathbb{Q}\log(U)$.*

Proof. Assume that H acts nilpotently on U. Then H acts nilpotently on any subfactor of U. Thus there exists an H-invariant central series $1 = U_k < \cdots < U_1 = U$ of U with H-central factors (take for example the upper central series of U and refine it to a series with H-central factors). Define a chain of Lie subalgebras of $\mathbb{Q}\log(U)$ by $\mathcal{L}_i = \mathbb{Q}\log(U_i)$. First we show that $\{0\} = \mathcal{L}_k < \cdots < \mathcal{L}_1 = \mathbb{Q}\log(U)$ is a central series of $\mathbb{Q}\log(U)$. Let $l_i \in \mathcal{L}_i$, $l \in \mathbb{Q}\log(U)$. Then $\exp(l) \in \hat{U}$, the $\mathbb{Q}$-powered hull of U, and thus there exists $z \in \mathbb{N}$ such that $n = \exp(l)^z \in U$. Thus $l = \frac{1}{z}\log(n)$, and with the same argument there exist $z_i \in \mathbb{N}$ and $n_i \in U_i$ such that $l_i = \frac{1}{z_i}\log(n_i)$. Now

$$[l_i, l] = \frac{1}{z_i z}[\log(n_i), \log(n)] = \frac{1}{z_i z}\log(y)$$

where, by Theorem 3.2.11, y is a product of rational powers of group commutators in n_i and n. Thus, y lies in the $\mathbb{Q}$-powered hull of U_{i+1}, and therefore $[l_i, l] \in \mathcal{L}_{i+1}$. Second we show that the factors of $\{0\} = \mathcal{L}_k < \cdots < \mathcal{L}_1 = \mathbb{Q}\log(U)$ are H-central. Let $h \in H$ and $l_i \in \mathcal{L}_i$. Then there exist $z_i \in \mathbb{N}$, $n_i \in U_i$ such that $l_i = \frac{1}{z_i}\log(n_i)$. Let $\varphi(H)$ be the induced action of H on $\mathbb{Q}\log(U)$. Then

$$l_i^{\varphi(h)} = \left(\frac{1}{z_i}\log(n_i)\right)^{\varphi(h)} = \frac{1}{z_i}\log(n_i^h) = \frac{1}{z_i}\log(n_i n_{i+1})$$

where $n_{i+1} \in U_{i+1}$. By the Baker–Campbell–Hausdorff formula,

$$\log(n_i n_{i+1}) = \log(n_i) + \log(n_{i+1}) + y,$$

where y is a $\mathbb{Q}$-linear combination of Lie commutators in $\log(n_i), \log(n_{i+1})$. Thus $y \in \mathcal{L}_{i+1}$, because $\mathcal{L}_i/\mathcal{L}_{i+1}$ is centralised by $\mathbb{Q}\log(U)$, and therefore l_i is centralised by H modulo $\mathcal{L}_{i+1}$.

Assume conversely that H acts nilpotently on $\mathbb{Q}\log(U)$. Then $\mathbb{Q}\log(U)$ has a central series $0 = \mathcal{L}_k < \cdots < \mathcal{L}_1 = \mathbb{Q}\log(U)$ with H-central factors. Define a descending chain of subgroups of U by $U_i = \exp(\mathcal{L}_i) \cap U$. For $n_i \in U$, $n \in U$, we see by Lemma 3.3.7 that $\log([n_i, n])$ is $\mathbb{Q}$-linear combination of Lie commutators in $\log(n_i)$ and $\log(n)$. Thus $\log([n_i, n]) \in \mathcal{L}_{i+1}$ and therefore $[n_i, n] \in U_{i+1}$. This implies that $U_k < \cdots < U_1$ is a central series of U. Further, for $n_i \in U_i$, $h \in H$ we have $\log(n_i^h) = \log(n_i)^{\varphi(h)} = \log(n_i) + y$, where $y \in \mathcal{L}_{i+1}$. By the inverse Baker–Campbell–Hausdorff formula h_1, $\log(n_i) + y = \log(n_i \exp(y)z)$ where z is a product of $\mathbb{Q}$-powers of group commutators in n_i, $\exp(y)$ and so $z \in \widehat{U_{i+1}}$, the $\mathbb{Q}$-powered hull of U_{i+1}. Thus ${n_i}^{-1} n_i^h = \exp(y)z \in \widehat{U_{i+1}}$ and so $n_i^h = n_i n_{i+1}$ for some $n_{i+1} \in U_{i+1}$. Therefore U_i/U_{i+1} is centralised by H. □

Let $\varphi : G \to \mathrm{GL}(e, \mathbb{Q})$ denote the action of the polycyclic group G on the Lie algebra $\mathbb{Q}\log(U)$ of the unipotent radical of G with respect to an arbitrary, fixed basis of $\mathbb{Q}\log(U)$. The following theorem shows how the nilpotency of G can be read off from the action φ in a similar way to the way polycyclicity is read off.

Definition 7.6.4. A polycyclic sequence $(g_1, \ldots, g_k)$ is called a *nilpotent sequence* if its corresponding polycyclic series $G_i = \langle g_i, \ldots, g_k \rangle$ is a central series (and hence the underlying group is nilpotent).

Theorem 7.6.5. *Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be polycyclic and let U be the unipotent radical of G. Let G/U be nilpotent with nilpotent sequence $(g_1U, \ldots, g_nU)$. Then G is nilpotent if and only if $\chi_{\varphi(g_i)}(x) = (x-1)^{m_i}$ for certain $m_i \in \mathbb{Z}$ and for $1 \leq i \leq n$.*

Proof. Assume that G is nilpotent. Then G acts nilpotently on U by Lemma 7.6.2 and thus on $\mathbb{Q}\log(U)$ by Theorem 7.6.3. Hence $\chi_{\varphi(g_i)}(x) = (x-1)^{m_i}$ for certain m_i and $1 \leq i \leq n$ follows.

Conversely, assume that $\chi_{\varphi(g_i)}(x) = (x-1)^{m_i}$ for all i. Denote by $\zeta_k(\mathbb{Q}\log(U))$ the k-th term of the upper central series of $\mathbb{Q}\log(U)$. Note that $\zeta_i(\mathbb{Q}\log(U))$ is an ideal of $\mathbb{Q}\log(U)$ and so in particular a Lie subalgebra. Further $\zeta_i(\mathbb{Q}\log(U))$ is invariant under automorphisms of $\mathbb{Q}\log(U)$ and therefore invariant under the action of G.

We show that G acts nilpotently on the factors

$$F_k = \zeta_k(\mathbb{Q}\log(U))/\zeta_{k+1}(\mathbb{Q}\log(U)).$$

IsNilpotent(G **)**

1: test whether G is polycyclic and return false if this is not the case.
2: as side-results of step 1, obtain a polycyclic sequence $\mathcal{G}$ of G/U and a basis $\mathcal{B}$ of $\mathbb{Q}\log(U)$ for the unipotent radical U of G,
3: using $\mathcal{G}$, test whether G/U is nilpotent and return false if this is not the case.
4: compute a nilpotent sequence $(g_1U, \ldots, g_nU)$ for G/U.
5: compute the induced action $\varphi(g_i)$ with respect to $\mathcal{B}$ for $1 \leq i \leq n$.
6: compute the minimal polynomial $\chi_{\varphi(g_i)}(x)$ of $\varphi(g_i)$ for $1 \leq i \leq n$.
7: **if** $\chi_{\varphi(g_i)}(x) = (x-1)^{m_i}$ for $1 \leq i \leq n$, **then**
8: return true
9: **else**
10: return false
11: **end if**

Table 7.2: The results of §7.6 yield the above algorithm to test nilpotency. The input G is a finitely generated subgroup of $\mathrm{GL}(d, \mathbb{Q})$.

This implies that G acts nilpotently on $\mathbb{Q}\log(U)$ and thus on U by Theorem 7.6.3. In turn, this yields the desired result by Lemma 7.6.2.

Let $k \in \mathbb{N}$ and let $\varphi_k(G)$ denote the action induced by G on F_k. Using a similar argumentation as in the proof of Theorem 7.6.3 we deduce that U acts trivially on F_k. Thus the sequence $(\varphi_k(g_1), \ldots, \varphi_k(g_n))$ is a polycyclic sequence of $\varphi_k(G)$. Let $G_i = \langle g_i, \ldots, g_n \rangle$. Then the groups $\varphi_k(G_i)$ for $1 \leq i \leq n$ form a central series of $\varphi_k(G)$. Let $l \in \{1, \ldots, n\}$ be maximal such that $\varphi_k(g_l) \neq 1$. Let W be the eigenspace of $\varphi_k(g_l)$. Then $F_k > W > \{0\}$, since $\varphi_k(g_l)$ is non-trivial and satisfies $(x-1)^{m_l} = 0$. By the choice of l, the element $\varphi_k(g_l)$ is contained in the center of $\varphi_k(G)$. This implies that W is a G-invariant subspace of F_k. The actions induced by G on F_k/W and W satisfy the assumption of the theorem. Thus by induction on the dimension, we can assume that G acts nilpotently on F_k/W and W. Thus G acts nilpotently on F_k. □

The results of this section yield an algorithm to test nilpotency. It is displayed in Table 7.2.

7.7 Testing virtual nilpotency

A modification of the nilpotency testing algorithm yields a method for testing virtual nilpotency. Let $G \leq \mathrm{GL}(d, \mathbb{Q})$ be finitely generated. As a first step,

we check whether G is virtually polycyclic with the method of Section 7.5. As a side-result of this algorithm, we obtain normal subgroup generators for the unipotent radical U of G and a polycyclic sequence for H/U where H is a p-congruence subgroup of G.

Note that H/U is free abelian; see [4]. Since $[G : H] < \infty$ and two subgroups of finite index intersect in a subgroup of finite index, G is virtually nilpotent if and only if H is virtually nilpotent. The latter condition can be checked with the following theorem. Recall that for $h \in \mathrm{GL}(e, \mathbb{Q})$ we denote by $\chi_h(x)$ the minimal polynomial of h.

Theorem 7.7.1. *Let $H \leq \mathrm{GL}(d, \mathbb{Q})$ and let U be the unipotent radical of H. Suppose that H/U is finitely generated abelian with polycyclic sequence $(g_1U, \ldots, g_nU)$. Then H is virtually nilpotent if and only if all roots of $\chi_{\varphi(g_i)}(x) = 0$ are roots of unity for $1 \leq i \leq n$.*

Proof. Assume that H is virtually nilpotent. Let $K \leq H$ with $s = [H : K] < \infty$ and K nilpotent. Then K acts nilpotently on $U \cap K$ and therefore, by Theorem 7.6.3, K acts nilpotently on $\mathbb{Q}\log(U \cap K)$. Since $[U : U \cap K] = [KU : K] \leq [H : K] < \infty$, $\mathbb{Q}\log(U \cap K) = \mathbb{Q}\log(U)$ and thus K acts nilpotently on $\mathbb{Q}\log(U)$. Therefore, since $g_i^s \in K$, the minimal polynomial of $\varphi(g_i^s) = \varphi(g_i)^s$ is $(x-1)^{m_i}$ for some $m_i \in \mathbb{N}$. Thus $\chi_{\varphi(g_i)}(x)$ divides $(x^s-1)^{m_i}$. This implies that all roots of $\chi_{\varphi(g_i)}(x) = 0$ are roots of unity.

Assume conversely that $E \subseteq \mathbb{C}$, the set of all eigenvalues of the matrices $\varphi(g_1), \ldots, \varphi(g_n)$, contains only roots of unity. Let $l \in \mathbb{N}$ such that $\lambda^l = 1$ for all $\lambda \in E$. Define $K = \langle g_1^l, \ldots, g_n^l, U\rangle$. Then $[H : K] = l^n$. In order to show that K acts nilpotently on U, by Theorem 7.6.5, it is sufficient to show that $\chi_{\varphi(g_i{}^l)} = (x-1)^{m_i}$ for some $m_i \in \mathbb{N}$ for $1 \leq i \leq n$. Let θ be an eigenvalue of $\varphi(g_i^l)$. From the Jordan normal form of $\varphi(g_i)$ it can be read off that $\theta = \lambda^l$ for some eigenvalue λ of $\varphi(g_i)$. Since $\lambda \in E$, $\theta = 1$ follows and thus $\chi_{\varphi(g_i{}^l)} = (x-1)^{m_i}$ for some $m_i \in \mathbb{N}$. □

7.8 Summary

Let G be a finitely generated virtually soluble subgroup of $\mathrm{GL}(d, \mathbb{Q})$. In Table 7.3 we summarise the criteria for testing (virtual) polycyclicity and (virtual) nilpotency of G.

	G polycyclic	G virtual polycyclic
condition on factor	G/U polycyclic	
action on Lie algebra	$\varphi(G)$ conjugate to a subgroup of $\mathrm{GL}(e,\mathbb{Z})$	$\varphi(H)$ conjugate to a subgroup of $\mathrm{GL}(e,\mathbb{Z})$

	G nilpotent	G virtual nilpotent
condition on factor	G/U nilpotent	
action on Lie algebra	$\varphi(G)$ conjugate to a unitriangular subgroup of $\mathrm{GL}(e,\mathbb{Z})$	$\varphi(H)$ conjugate to a triangular subgroup of $\mathrm{GL}(e,\mathbb{Z})$ with roots of unity on the diagonal

Table 7.3: In this table we summarise the criteria for testing (virtual) polycyclicity and (virtual) nilpotency of a finitely generated virtual soluble group $G \leq \mathrm{GL}(d,\mathbb{Q})$. Let H be a p-congruence subgroup of G and U the unipotent radical of G where H/U is a free-abelian group. Recall that we denote by $\varphi(G) \leq \mathrm{GL}(e,\mathbb{Q})$ the induced action of G on the Lie algebra $\mathcal{L}(U)$ with respect to an arbitrary fixed basis.

7.9 Implementation and examples

We illustrate our algorithms on the simple example group G, already mentioned in Section 7.1.4, which is generated by

$$g = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad h = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

The series $V = \mathbb{Q}^2 > W = \langle (0,1) \rangle_{\mathbb{Q}} > \{0\}$ is a semisimple series for G. The induced actions $G_{V/W}$ and G_W are both polycyclic, and thus G is soluble. Let U be the centraliser of the semisimple series; thus U is the unipotent radical for G. Then G/U is an infinite cyclic group and (gU) is a polycyclic sequence for G/U. Further $U = \langle h \rangle^G$. It follows that

$$\mathbb{Q}\log(U) = \mathbb{Q}\log(\langle h \rangle^G) = \mathbb{Q}\log(\langle h \rangle)^{\varphi(G)} = \left(\langle \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \rangle_{\mathbb{Q}} \right)^{\varphi(G)} = \langle \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \rangle_{\mathbb{Q}}.$$

It can be read off that the induced action of g on $\mathbb{Q}\log(U)$ with respect to the basis $\mathcal{B} = \{ \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right) \}$ is given by the matrix $\varphi(g) = (\frac{1}{2})$. The minimal polynomial of $\varphi(g)$ is not in $\mathbb{Z}[X]$ and thus G is not polycyclic.

7.9.1 Runtimes

The algorithm IsPolycyclic of Section 7.4 was implemented in GAP [41] as a part of the Polenta package [3]. Instead of computing minimal polynomials in step (6) of the algorithm we determine characteristic polynomials because this is more efficient and because the minimal polynomial of a rational matrix is integral if and only if the characteristic polynomial is integral.

Alternatively the method in [8], which is also implemented in GAP, could be used to test whether the induced action of G to $\mathbb{Q}\log(U)$ conjugates into $\mathrm{GL}(e, \mathbb{Z})$. Therefore this method could replace the steps (6) to (11) of our algorithm "IsPolycyclic". We compared this variation with our method and did not notice any difference in the runtimes for our example groups.

A method for testing virtual polycyclicity has not yet been implemented. To handle this case it is necessary to compute short finite presentations of finite non-soluble matrix groups.

In Table 7.4 we display runtimes for some example matrix groups and we also summarise some of the properties of the considered groups. All example groups considered in Table 7.4 are soluble and not contained in $\mathrm{GL}(d, \mathbb{Z})$. The groups G_1, G_2 are unipotent, G_3, G_4 are almost crystallographic groups. The group G_5 was constructed using the Kronecker product of generators of an almost crystallographic group. The group G_6 is a randomly generated

subgroup of the direct product of a unipotent and a free-abelian-by-finite group. The group G_7 is the group G from the beginning of Section 7.9. The groups G_8, G_9 are randomly generated upper-block-triangular matrix groups.

Every example group G_i is available in the package Polenta via the function "SolvableMatGroupExams(i)". A group $G \leq \mathrm{GL}(d, \mathbb{Q})$ given by generators can be tested to be polycyclic using "IsPolycyclicMatGroup(G)". All computations where carried out in GAP Version 4.4.7. on a 3.2 gigahertz Pentium 4 processor and 90 MB of memory for GAP.

Group	Degree	No. gens	Rank	Dim $\mathbb{Q}\log(U)$	Runtime
G_1	4	2	4	4	47
G_2	5	2	6	6	109
G_3	5	5	4	4	822
G_4	5	5	4	4	568
G_5	16	5	3	3	557
G_6	20	11	7	4	373083
G_7	2	2	-	1	150
G_8	6	4	-	10	15966
G_9	8	4	-	13	14379

Table 7.4: Testing polycyclicity: The columns display the degree d, the number of generators, the rank (or Hirsch length) and the dimension of $\mathbb{Q}\log(U)$ for every of the considered examples $G_1, \ldots, G_9$. If no rank is given, then the example group is not polycyclic. The last column contains the time in milliseconds which is needed by the algorithm IsPolycyclic of Section 7.4.

Appendix A

Algebraic number theory

In this appendix we recall some basic facts from algebraic number theory. For more background we refer to [40].

Definition A.0.1. Let $\mathbb{F}$ be a subfield of $\mathbb{C}$ such that $[\mathbb{F} : \mathbb{Q}]$ is finite. Then $\mathbb{F}$ is called a *number field* or an *algebraic extension* of $\mathbb{Q}$. Let $\theta \in \mathbb{C}$. By $\mathbb{Q}(\theta)$ we denote the smallest subfield of $\mathbb{C}$ which contains θ.

Lemma A.0.2. *Let F be a number field. Then there exists an element $\theta \in \mathbb{C}$ such that $\mathbb{F} = \mathbb{Q}(\theta)$.*

Definition A.0.3. A complex number θ is said to be an *algebraic integer*, if the minimal polynomial of θ is in $\mathbb{Z}[X]$.

Definition A.0.4. Let $\mathbb{F}$ be a number field. The set of algebraic integers in $\mathbb{F}$ is called the *maximal order* $\mathcal{O}$ of $\mathbb{F}$. An element $u \in \mathcal{O}$ is called a *unit*, if u^{-1} is also contained in $\mathcal{O}$. The set of all units in $\mathcal{O}$ is called the *unit group* of $\mathcal{O}$ and is denoted by $U(\mathcal{O})$. A ring $R \subseteq \mathcal{O}$ with $1 \in R$ that spans $\mathbb{F}$ over $\mathbb{Q}$ is called an order of $\mathbb{F}$.

Theorem A.0.5 (Dirichlet's Units Theorem)**.** *Let $\mathbb{F}$ be a number field with maximal order $\mathcal{O}$. Then the torsion subgroup T of $U(\mathcal{O})$ is finite and generated by one element $\zeta \in U(\mathcal{O})$ which is called the torsion unit of $U(\mathcal{O})$. Further there exists fundamental units $\varepsilon_1, \ldots, \varepsilon_r \in U(\mathcal{O})$ such that every unit $u \in U(\mathcal{O})$ can be written uniquely as*

$$u = \zeta^{f_0} \cdot \varepsilon_1^{f_1} \cdot \varepsilon_2^{f_2} \cdots \varepsilon_r^{f_r}$$

where $f_i \in \mathbb{Z}$ and $0 \leq f_0 < |T|$.

Definition A.0.6. Let p be a prime. The ring of *p-adic integers* $\mathbb{Z}_p$ is defined to be the inverse limit of the finite quotients $\mathbb{Z}/p^n\mathbb{Z}$.

Bibliography

[1] B. Assmann. Algorithmic use of the Mal'cev correspondence. In C. M. Campbell and E. F. Robertson, editors, *Groups - St. Andrews 2005*.

[2] B. Assmann. *Guarana - Applications of Lie methods in computational group theory*, 2006. A GAP 4 package, see [41].

[3] B. Assmann. *Polenta - Polycyclic presentations for matrix groups*, 2006. A refereed GAP 4 package, see [41].

[4] B. Assmann and B. Eick. Computing polycyclic presentations for polycyclic rational matrix groups. *J. Symb. Comput.*, 40:1269–1284, 2005.

[5] B. Assmann and B. Eick. Testing polycyclicity of finitely generated rational matrix groups. *To appear in Mathematics of Computation*, 2007.

[6] B. Assmann and S. Linton. Using the Mal'cev correspondence for collection in polycyclic groups. *To appear in Journal of Algebra*, 2007.

[7] L. Babai, R. Beals, J. Cai, G. Ivanyos, and E. M. Luks. Multiplicative equations over commuting matrices. In *SODA '96: Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 498–507, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[8] L. Babai, R. Beals, and D. Rockmore. Deciding finiteness of matrix groups in deterministic polynomial time. In *Proc. of International Symposium on Symbolic and Algebraic Computation* ISSAC '93, pages 117–126. (Kiev), ACM Press, 1993.

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

[10] R. Beals. Improved algorithms for the Tits alternative. In W. M. Kantor and A. Seress, editors, *Groups and Computation III*, pages 63 – 77. (DIMACS, 1999), 2001.

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

[12] W. de Graaf. *Lie Algebras: Theory and Algorithms*. North Holland, 2000.

[13] W. de Graaf and W. Nickel. Constructing faithful representations of finitely-generated torsion-free nilpotent groups. *J. Symb. Comput.*, 33:31–41, 2002.

[14] A. Detinko and D. Flannery. Free subgroups in linear groups. *LMS Journal of Computation and Mathematics*, 9:104–134, 2006.

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

[16] M. du Sautoy. Polycyclic groups, analytic groups and algebraic groups. *Proc. London Math. Soc. (3)*, 85:62–92, 2002.

[17] B. Eick. Algorithms for polycyclic groups. Habilitationsschrift, Universität Kassel, 2001.

[18] V. Gebhardt. Efficient collection in infinite polycyclic groups. *J. Symb. Comput.*, 34 (3):213–228, 2002.

[19] P. Hall. Nilpotent groups. In *The collected works of Philip Hall*, pages 415 – 462. Clarendon Press, Oxford, 1988. Notes of lectures given at the Canadian Mathematical Congress 1957 Summer Seminar.

[20] D. F. Holt, B. Eick, and E. A. O'Brien. *Handbook of Computational Group Theory*. Discrete Mathematics and its Applications. CRC Press, 2005.

[21] N. Jacobson. *Lie algebras*. Dover Publications, 1962.

[22] M. Kargapolov and J. Merzljakov. *Fundamentals of the Theory of Groups*. Graduate Texts in Mathematics. Springer, 1979.

[23] E. Khukhro. *p-Automorphisms of Finite p-Groups*, volume 246 of *Lecture Note Series*. London Mathematical Soc., 1998.

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

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

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

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

[28] A. J. Mal'cev. On certain classes of infinite soluble groups. *Amer. Math. Soc. Transl.*, 2 (2):1–21, 1956.

[29] W. Merkwitz. Symbolische Multiplikation in nilpotenten Gruppen mit Deep Thought. Diplomarbeit, RWTH Aachen, 1997.

[30] K. Mihailova. The occurence problem for direct products of groups (in Russian). *Dokl. Akad. Nauk. SSSR*, 119:1103–1105, 1958.

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

[32] W. Nickel. *NQ*, 1998. A refereed GAP 4 package, see [41].

[33] W. Nickel. Matrix representations for torsion-free nilpotent groups by Deep Thought. *J. Algebra*, 300:603–626, 2006.

[34] E. O'Brien and M. Vaughan-Lee. The 2-generator restricted burnside group of exponent 7. *Internat. J. Algebra Comput.*, 12:575–592, 2002.

[35] G. Ostheimer. Practical algorithms for polycyclic matrix groups. *J. Symb. Comput.*, 28:361 – 379, 1999.

[36] M. Reinsch. A simple expression for the terms of the Baker–Campbell–Hausdorff series. *Journal of Mathematical Physics*, 41 (4):2434–2442, 2000.

[37] 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.

[38] D. Segal. *Polycyclic Groups.* Cambridge University Press, Cambridge, 1983.

[39] C. C. Sims. *Computation with finitely presented groups.* Cambridge University Press, Cambridge, 1994.

[40] I. Stewart and D. Tall. *Algebraic number theory and Fermat's last theorem.* A K Peters, 2002.

[41] The GAP Group. *GAP – Groups, Algorithms and Programming*. www.gap-system.org, 2006.

[42] J. Tits. Free subgroups in linear groups. *J. Algebra*, 20:250–270, 1972.

[43] M. Vaughan-Lee. Collection from the left. *J. Symb. Comput.*, 9:725–733, 1990.

[44] M. Vaughan-Lee. *The Restricted Burnside Problem*, volume 5 of *London Math. Soc. Monographs, (N. S.)*. Oxford University Press, Oxford, 1990.