The Word and Conjugacy Problems in Classes of Solvable Groups

Svetla Vassileva

Master of Science

Department of Mathematics and Statistics

McGill University

Montreal,Quebec

August 2009

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science

DEDICATION

To Valentin Prahov – my chess trainer, my grandfather, my friend.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Alexei Miasnikov, for his vision and guidance. I would also like to thank Jeremy Macdonald, for his patience with carefully proof-reading this thesis and for his help with making it readable. Special thanks to my family for their care and support throughout my studies.

ABSTRACT

This thesis is a survey of certain algorithmic problems in group theory and their computational complexities. In particular, it consists of a detailed review of the decidability and complexity of the word and conjugacy problems in several classes of solvable groups, followed by two original results. The first result states that the Conjugacy Problem in wreath products which satisfy certain elementary conditions is decidable in polynomial time. It is largely based on work by Jane Matthews, published in [14]. The second result, based on ideas of Remeslennikov and Sokolov [17], and Myasnikov, Roman'kov, Ushakov and Vershik [16] gives a uniform polynomial time algorithm to decide the Conjugacy Problem in free solvable groups.

ABRÉGÉ

Cette thèse est une synthèse de certains problèmes algorithmiques dans la théorie des groupes et leur complexité computationnelle. Plus particulièrement, elle présente une revue détaillée de la décidabilité et de la complexité des problèmes du mot et de la conjugaison dans plusieurs classes de groupes solubles, suivie de deux nouveaux résultats. Le premier résultat énonce que le problème de la conjugaison dans les produits couronne qui satisfont certaines conditions élémentaires est décidable en temps polynomial. Elle part d'une publication de Jane Matthews [14]. Le deuxième résultat, basé sur des idées de Remeslennikov et Sokolov [17] et de Myasnikov, Roman'kov, Ushakov et Vershik [16], présente un algorithme en temps polynomial uniforme pour décider le problème de conjugaison dans les groupes solubles libres.

TABLE OF CONTENTS

CHAPTER 1
Introduction

The modern theory of solvable groups is full of new results and ideas and it boasts a wide range of applications from computer vision to cryptography. All these require powerful computational tools: to compute various related objects, solve equations, find numerical invariants, run massive computer experiments. To put it shortly, a robust algorithmic theory of solvable groups is needed.

Algorithmic results on solvable groups have been, for a long time, prized in combinatorial group theory, and have revealed remarkable relations with computational commutative algebra and number theory. Recall for example, two polar outstanding results: decidability of the Isomorphism Problem for polycyclic groups (Grunewald and Segal) and the undecidability of the Diophantine problem for free nilpotent and free metabelian groups (Romankov, Repin in [18], [19], [20], [21]).

Algorithmic problems in group theory were considered as early as 1910, when Dehn introduced three now famous problems: the Word Problem, the Conjugacy Problem and the Isomorphism Problem. At the time, people were fascinated with the decidability of these problems and have left a rich heritage in this area. It is known for example that it is impossible to have a unified decidability theory for such problems. Miller constructed a group which has decidable Word Problem and undecidable Conjugacy Problem [15]. Thus, the best one can do is to solve these problems for specific groups or classes of groups. There have been many positive results in this direction. To mention only a few, the decidability of the Word Problem in braid groups [1], the Conjugacy Problem in hyperbolic groups (Gromov, [8]), the Conjugacy Problem in wreath products (Matthews, [14]), the Word and Conjugacy Problem in Grigorchuk groups ([7], [12]), and the Word and Conjugacy Problem in free solvable groups ([17]).

However, the emphasis nowadays falls on the computational efficiency of the solutions to these problems. Indeed, if an algorithm is to be used in practice, it is crucial that it is within the power of technology to execute it in reasonable time. While the available technology constantly evolves and time bounds computed in terms of hours or days might become obsolete, there is an intrinsic notion of how 'fast' an algorithm is that allows one to compare algorithms (and problems) independent of the current state of technology. It is therefore of crucial importance to improve on the ground work done on decidability and focus on efficiency. In this spirit it is worth mentioning the work of Lysenok, Miasnikov and Ushakov who showed that the Conjugacy Problem in Grigorchuk groups is decidable in polynomial time, the work of Gersten and Short on the decidability of the Conjugacy Problem in biautomatic groups, as well as the work of Marshall, Bridson and Haefliger, Epstein and Holt which, through successively improving time bounds, culminates in showing that the Conjugacy Problem in hyperbolic groups is decidable in linear time. Of great importance, both as a powerful result and as an inspiring and promising approach, is the work of Miasnikov, Roman'kov, Ushakov and Vershik, who use a fascinating geometric view-point to show that the Word and Geodesic problems in free solvable groups are decidable in cubic time.

In this thesis, we build on their results to show that the Conjugacy Problem in free solvable groups is decidable in polynomial time. Chapter 5 gives a detailed and complete account of the main techniques used – the Magnus embedding, Fox derivatives and the computation thereof. In Chapter 6.1, we introduce Matthews' result to decide the Conjugacy Problem in wreath products. We make a major modification, which makes the algorithm that she originally presented, which would have unbounded complexity, into one that has polynomial time complexity. Finally, in Chapter 6.2, we show that the Conjugacy Problem in free solvable groups is decidable in polynomial time. The ideas of this result were inspired by the above mentioned work of Miasnikov, Roman'kov, Ushakov and Vershik [16] and the work of Remeslennikov and Sokolov [17]. Chapter 3 and Chapter 4 place these results in context by giving a survey of the field. More precisely, Chapter 4 presents a compilation of results pertaining to the word and conjugacy problems in some important classes of solvable groups. Chapter 3 gives a more general survey of the results on the complexity and decidability of the word and conjugacy problems in several large classes of groups. All the basic notions needed to understand the material in this thesis are briefly introduced in Chapter 2. With this we wish the reader a pleasant and fruitful reading.

CHAPTER 2
Background

In this section are presented the basic notions used in the study of algorithmic group theory and, in particular in this thesis.

2.1 Free Groups, Presentations and Cayley Graphs

Let $X = \{x_1, \dots, x_r\}$ be any set. For each x_i introduce another symbol x_i^{-1} and a bijection on X defined by $x_i \mapsto x_i^{-1}$ and $(x_i^{-1})^{-1}$. Let $X^{\pm} = \{x_1, \dots, x_r, x_1^{-1}, \dots, x_r^{-1}\}$. A *word*, w, over the alphabet $X^{\pm}$ is a finite string $w = x_{i_1} \cdots x_{i_k}$, where each x_{i_j} is in $X^{\pm}$. Sometimes one writes $w(i)$ for the i^{th} character of w. The *word length*, $|w|$, of a word w is the number of literals k in w. Denote by $(X^{\pm})^*$ the set of all words over the alphabet $X^{\pm}$ including the empty word which will be denoted by 1.

Define the relation $\sim$ on $(X^{\pm})^*$ by setting $w_1 \sim w_2$ if and only if one can transform w_1 into w_2 by finitely many applications of the following transformations:

(T1) Deletion of a subword of the form $x_l x_l^{-1}$: for some j such that $x_{i_{j+1}} = x_{i_j}^{-1}$, replace w by $x_{i_1} \cdots x_{i_{j-1}} x_{i_{j+2}} \cdots x_{i_k}$.

(T2) Insertion of a subword of the form $x_l x_l^{-1}$, for any l, at position j: replace w by the word $x_{i_1} \cdots x_{i_j} x_l x_l^{-1} x_{i_{j+1}} \cdots x_{i_k}$.

Lemma 2.1.1. *The relation $\sim$ is an equivalence relation.*

Proof. Let $w, w_1, w_2, w_3 \in (X^{\pm})^*$. Reflexivity is obvious. Symmetry follows from the fact that $(T1)$ and $(T2)$ are inverses of each other, in the sense that applying them one after the other at the same position will have no net effect. To see that $\sim$ is transitive, suppose that $w_1 \sim w_2$ and $w_2 \sim w_3$, i.e., w_2 can be obtained from w_1 by a finite sequence of transformations and w_3 can be obtained from w_2 by another finite sequence of transformations. Concatenating these two sequences, gives a finite sequence of transformations which brings w_1 into w_3. □

Let $F = (X^{\pm})^* / \sim$ and denote the equivalence class of w by $[w]$.

Proposition 2.1.2. *Define a multiplication on F by $[w_1][w_2] = [w_1 w_2]$. Then F is a group, called the* free group on X.

Proof. This multiplication is well-defined. Let $w_1 \sim w_2$ and $w_3 \sim w_4$. Then $w_1 w_3 \sim w_2 w_4$, since $w_1 w_3$ can be transformed in $w_2 w_3$, which in turn can be transformed in $w_2 w_4$. Since concatenation is associative, so is the multiplication of equivalence classes. The group identity is the empty word, ϵ. The inverse of $[w]$ is $[w^{-1}]$. Indeed, let $w = x_{i_1} \cdots x_{i_k}$. Then $[w]^{-1} = [w^{-1}] = [x_{i_k}^{-1} \cdots x_{i_1}^{-1}]$. Then $[w][w]^{-1} = [x_{i_1} \cdots x_{i_k} x_{i_k}^{-1} \cdots x_{i_1}^{-1}]$ which can readily be seen to reduce to ϵ. Similarly, $[w]^{-1}[w] = [\epsilon]$. □

Note that every word over the alphabet $X^{\pm}$ represents an element of the group $F(X)$. However, the same element in $F(X)$ can be represented by different words. For example, 1, xx^{-1}, $x^2 x^{-1} x^{-3} x^2$ all represent the identity.

Free groups are often defined through the *universal property* they satisfy. Namely, given a group G and any map $\phi : X \to G$, it extends to a homomorphism $\pi : F(X) \to G$. If π is surjective, then X is a *set of generators* for the group G. The word w is called a *representative* for the group element $\overline{w} = \pi(w)$. The notation

$G = \langle X \rangle$ is used when X generates G. It is easy to check that the two definitions are equivalent.

It is often convenient to think of group elements as words. This is achieved through a group presentation. In general, for any $N \lhd G$, the epimorphism $f : G \to G/N$, sending each $g \in G$ to its coset $gN \in G/N$, is called the *canonical epimorphism.*

Definition 2.1.3. *Let G be a group and let $R \subseteq G$. Denote by $<< R >>$ the* normal closure *of R in G, i.e., the minimal normal subgroup in G containing R.*

Definition 2.1.4. *A group G is said to have presentation $\langle X \mid R \rangle$, with $R \subseteq F(X)$, if $G = \langle X \rangle$ and*

$$G \simeq F(X) / << R >> .$$

In this case R is called the set of relators *for the presentation.*

A *finite presentation* for a group is one for which $|X| + |R|$ is finite. A group is *finitely presented* if it has a finite presentation. A notion related to this is that of a *finitely generated* group, i.e., one which has a presentation for which the generating set is finite.

Note that every group has a presentation. This can easily be seen by letting every element be a generator and taking the multiplication table as a set of relators. More precisely, given a group G, construct a presentation $X|R$ for it with $X = G$ and $R = \{ghk^{-1} | gh = k, g, h, k \in G\}$. If the group is infinite, this will give an infinite presentation, though a finite presentation might exist. Presentations are useful because they allow one to take a computation-oriented approach, namely to think of group elements as words (or strings) without losing any information about the group.

The Cayley graph of a group, being a geometric representation of a group, is one of the basic tools to introduce the topological view point in group theory.

Definition 2.1.5. *Let $G = \langle X \rangle$ be a group. Define the* Cayley graph $\Gamma(G) = \Gamma(G, X)$ *to be a directed labelled graph with vertex set G. There is an edge from vertex g to vertex h if $h = gx$ for some generator $x \in X$. In this case, this edge is labelled by the generator x.*

Moreover, for every edge (g, h) labelled by x, the edge (h, g) labelled by x^{-1} is also in $\Gamma(G)$ (this corresponds to multiplying by the inverse of a generator). Note that the Cayley graph contains all the information of the group. It is also worth noting that a path of edges $e_{i_1}, \ldots, e_{i_k}$ in $\Gamma(G)$ labelled by $x_{i_1}, \ldots, x_{i_k}$ corresponds to the word $w = x_{i_1} \cdots x_{i_k}$. Conversely, starting at a vertex v there is a unique way to 'read' a word w from there, i.e., there is a unique path starting at v which is labelled by w. Thus, if one starts reading a word w at the vertex corresponding to the group identity and ends back at the identity vertex, then the word w represents the identity, i.e., $w = 1$ in the group. The converse is also true – if $w = 1$ in the group, then the path it represents in the Cayley graph is a loop. Thus Cayley graphs are a powerful tool to solve the *Word Problem* (defined in Section 2.2).

Observe that the Cayley graph of any group G can be turned into a metric space by assigning length one to each edge and taking the distance between two vertices to be the length of the shortest path between them. In other words, for words $u, v, w \in G$, or equivalently, for vertices u, v, w in $\Gamma(G)$, define

$$d(u, v) = min\{|w| \mid u = vw.\}$$

This is called the *word metric* on $\Gamma(G)$. Of course, it depends on the set of generators X, though the dependence is not significant for most theoretical results (it depends on X up to quasi isometry). A word which realises the distance between two vertices is called a *geodesic*. More precisely, w is a geodesic between two vertices u and v if $|w| = d(u,v)$. A geodesic for an element g is a geodesic from 1 to g.

2.2 Decidability and Complexity

The study of algorithmic problems in group theory began in 1910 with Dehn's formulation of the three famous algorithmic problems in groups – the Word Problem, the Conjugacy Problem and the Isomorphism Problem.

Word Problem. *For a group $G = \langle X \rangle$, given two words u, v over $X^{\pm}$, decide whether*

$$u = v$$

as elements of G. Equivalently, given a word $w \in G$, decide, whether

$$w = 1$$

in the group G.

The second statement is equivalent, since given words $u, v \in G$, $u = v$ if and only if $uv^{-1} = 1$.

Conjugacy Problem. *For a group $G = \langle X \rangle$, given words u, v in G, decide whether u and v are conjugate in G, i.e., decide whether there exists $g \in G$ such that*

$$g^{-1}ug = v.$$

Observe that if one can solve the Conjugacy Problem, then one can also solve the Word Problem. For a given word $w \in G$, in order to check whether $w = 1$, it is enough to check whether w is conjugate to the identity.

Isomorphism Problem. *Given two group presentations $G_1 = \langle X_1 \mid R_1 \rangle$ and $G_2 = \langle X_2 \mid R_2 \rangle$, decide whether the groups G_1 and G_2 are isomorphic.*

The above three problems, as formulated by Dehn, all implicitly referred to finitely presented groups. Nowadays, not only are there many results about groups which do not have finite presentations, but also it is sometimes preferable to work with an infinite presentation even when there is a well-known finite one for the given group.

Since Dehn drew interest to algorithmic group theory, many more important decision problems have sprung up. The Bounded Geodesic Length Problem, Power Problem and Membership Problem are but a few of these.

Definition 2.2.1. *Let G be a group with a finite set of generators $X = \{x_1, \ldots, x_n\}$ and let $\mu : F(X) \to G$ be the canonical epimorphism. The* geodesic length $l_X(g)$ *of an element $g \in G$ relative to X is defined by*

$$l_X(g) = \min\{|w| \mid w \in F(X), \mu(w) = g\}.$$

Bounded Geodesic Length Problem. *Let G be a group with a finite generating set X. Given a word $w \in F(X)$ and a natural number k determine if $l_X(w) \leq k$.*

This is a problem which very naturally emerges from topology and, while related to the Word Problem, is not equivalent. More precisely, let $\Gamma(G)$ be the Cayley graph for a group G with generating set X. For a given word w in generators X, $w = 1$ in

G if and only if $l_X(w) = 1$. Hence solving the Bounded Geodesic Length Problem gives a solutions to the Word Problem.

Power Problem. *Let $G = \langle X \rangle$ be a group. Given x, y over $X^{\pm}$, determine whether there is an integer n such that*

$$y = x^n.$$

Equivalently, decide whether

$$y \in \langle x \rangle.$$

A generalisation of the Power Problem is the Membership Problem.

Membership Problem. *Let $G = \langle X \rangle$ be a group. Given words $w_1 \dots, w_n$ and w over $X^{\pm}$, let $H = \langle w_1, \dots, w_m \rangle$ be the subgroup of G generated by those words, decide whether $w \in H$.*

All the famous algorithmic problems being introduced, it is time to say a few words on computation and decidability. First, one needs to have a model of computation in mind. A commonly used model that is easy to reason about and hence is useful for theoretical results, is the Turing machine. Briefly, a Turing machine consists of a one-sided infinite memory tape and a head which can perform several operations – read a symbol under the head, write a symbol under the head, move the head left, move the head right, stay in place. Any such operation is called a *step* in the computation. In this model, an *algorithm* is a program (represented as a Turing machine), prescribing how to execute these operations. Running this program corresponds to performing a sequence of steps. The algorithm is said to *terminate* if after some finite number of steps the head stops performing operations. Some authors require the sequence of operations to terminate in order to call it an algorithm, but in the context of computational group theory, it is conventional to call an algorithm any sequence of steps regardless of consistency and termination. Thus there is a Turing machine associated to every algorithm.

Another model of computation, which is very similar to the Turing machine, is a RAM (Random Access Memory) machine. It differs from a Turing machine in that the head can move to any cell of the tape in constant time. This makes the analysis and formal description of algorithms less tedious, and is more suitable for our purposes. It does not affect decidability, or even polynomial time decidability (to be defined later). However, the differences in complexity generated by using the RAM model of computation rather than a Turing machine will be pointed out throughout.

A *decision problem* is a problem which requires a yes/no answer. Of course not all problems are of this form. The decision context is particularly good to compare the hardness of problems. Using the same underlying computational model and requiring the same type of answer puts all problems on an equal footing, thus concentrating on the inherent hardness of the computation.

Definition 2.2.2. *A decision problem, P, is said to be* decidable *if there is an algorithm, $\mathcal{A}$, which terminates and correctly answers the question posed in P.*

Of course in practice, termination is crucial, but the time it takes to run an algorithm is as important. This raises the question of how to measure 'time'. Clearly, the running time depends on the size of the input. One needs to measure 'time' in a sense that is intrinsic to the algorithm. In order to say that algorithm $\mathcal{A}$ is 'faster' than algorithm $\mathcal{B}$, one needs to make a statement about their behaviour on *all* inputs. This leads to the notion of complexity.

Given an algorithm $\mathcal{A}$, its *time function*, $T_{\mathcal{A}} : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$, gives the number of steps it takes its associated Turing machine to execute algorithm $\mathcal{A}$, given an input of n bits. Technically, the input of the time function is the number of bits of the encoded input. Typically in group theory, one chooses a more meaningful measurement, which varies linearly with the number of bits, e.g., the word length. Thus, the analysis and comparison of algorithms is often done through their time functions. It is therefore necessary to have a way of comparing time functions.

Definition 2.2.3. *Let $f, g : \mathbb{R} \to \mathbb{R}$. Then $f(x) \in O(g(x))$ if there exist $M \in \mathbb{R}^+$ and $x_0 \in \mathbb{R}$ such that*

$$|f(x)| \leq M|g(x)|,$$

for all $x \geq x_0$.

This means that for large enough inputs, $f(x)$ is bounded by $g(x)$. Note that this defines a class of functions ordered by inclusion. Namely, if $f(x) \in O(g(x))$, then $O(f(x)) \subseteq O(g(x))$. This order is transitive, since $f(x) \in O(g(x))$ and $g(x) \in O(h(x))$, means that $f(x)$ is bounded by $g(x)$, which in turn is bounded by $h(x)$. So $f(x) \in O(h(x))$.

Definition 2.2.4. *A decision problem, P, is* polynomial time decidable *(or* decidable in polynomial time*, or* has polynomial complexity*) if there is an algorithm $\mathcal{A}$, with time function $T_{\mathcal{A}}$, which correctly solves P and a polynomial $p(x)$ such that $T_{\mathcal{A}} \in O(p(x))$.*

In other words, a decision problem is polynomial time decidable if there is an algorithm which solves it in time polynomial in the length of the input. An algorithm, $\mathcal{A}$, is *linear* if $T_{\mathcal{A}} \in O(n)$.

Note that if $p(x)$ and $q(x)$ are polynomials, then $p(q(x))$ is also a polynomial. This shows that a problem which is polynomial time decidable in terms of its subroutines is polynomial time decidable, as long as the subroutines can be computed in polynomial time. Also, observe that, by virtue of the definition, $kf(x) \in O(f(x))$ for any constant $k > 0$. In particular, since any pre-computation is constant with respect to the input, it will not contribute to the complexity of the algorithm.

All the decision problems above were defined for particular groups, so usually one restricts oneself to solving the given problem in a fixed group. However, one might try to solve an algorithmic problem for a class of groups. An algorithm is *uniform* if it solves successfully all instances of a given problem in a class of groups. In this case, some information about the group (for example a presentation, or a set of generators) is part of the input. In the sequel, a uniform algorithm for solving the Conjugacy Problem in the class of free solvable groups is presented.

CHAPTER 3
Algorithmic Problems in Group Theory

In this section, the decidability and time complexity of the Word and Conjugacy Problems in several classes of groups will be discussed.

3.1 Free Groups

Free groups are in a sense the simplest example to be considered in the computational context, due to the minimal number of relations they have. Consequently, the algorithms to solve problems there tend to be not only efficient, but also elegant and simple to describe.

Let F be a free group on generators $X = \{x_1, \ldots, x_r\}$ as defined in Section 2.1. A word $w \in F$ is *freely reduced* if it contains no substrings of the form xx^{-1} or $x^{-1}x$ with $x \in X$. It is usual to consider freely reduced words instead of just strings. Hence we first present an efficient algorithm to freely reduce a word. In fact, for rhetorical purposes, we present two algorithms – an elegant one, which has complexity $O(|w| \log |w|)$ in the Turing (as well as in the RAM) model of computation, and one which is linear in the RAM model of computation (but has complexity $O(|w| \log |w|)$ with respect to a Turing machine).

Proposition 3.1.1. *Let w be a word over X. One can find a freely reduced word $\overline{w}$ over X such that $w = \overline{w}$ in $F(X)$ in time $O(|w|)$.*

Proof. Let $w = x_{i_1}^{\epsilon_1} \cdots x_{i_n}^{\epsilon_n}$ be a word over generators $X^{\pm}$ (with $\epsilon_j = \pm 1$). Start reading w, keeping two 'current' pointers, p_1 and p_2. To the first character, x_{i_1}, associate a counter, N_1 containing ϵ_1, set the pointer p_1 to x_{i_2} and p_2 to N_1. Denote by x_{p_1} the character pointed to by p_1 and by $N(p_2)$ the counter (and by x_{p_2} the corresponding generator) pointed to by p_2. At every following step, move the pointer p_1 one character to the right. If $x_{p_1+1} = x_{p_2}$, replace the content of N_{p_2} with $N_{p_2} + \epsilon_{p_1+1}$. Otherwise, create another counter N_{p_2+1}, whose content is ϵ_{p_1+1} and set the pointer p_2 to N_{p_2+1}. If during this process some counter N_{p_2} is set to zero, delete it, set the pointer p_2 to N_{p_2-1} and continue. If it is the first counter, N_1 which is set to zero, then create a new counter N_{p_1} containing ϵ_{p_1}, set p_2 to N_{p_1}, move p_1 one character to the right and continue. This procedure repeats until the end of w is reached. Output $\overline{w} = x_{j_1}^{N_1} \cdots x_{j_k}^{N_k}$, where x_{j_l} is the generator corresponding to N_l. □

The following algorithm illustrates how reduction is typically thought of. It does not have the best possible complexity in the RAM model, but the bound on its complexity is tight with respect to a Turing machine. Moreover, it is more illustrative of how reduction is intuitively thought of.

Proposition 3.1.2. *Let w be a word over X. One can find a freely reduced word $\overline{w}$ over X such that $w = \overline{w}$ in $F(X)$ in time $O(|w| \log(|w|))$.*

Proof. The idea is that since cancellation only happens locally, one can use a divide-and-conquer technique. Let $w \in F$. The algorithm to compute $\overline{w}$ is given recursively as follows:

If $|w| = 1$, then w is freely reduced and $\overline{w} = w$.

If $|w| = n > 1$, let $w = w_1 \cdots w_n$. Split w into two words $w' = w_1 \cdots w_k$ and $w'' = w_{k+1} \cdots w_n$, where $k = \lfloor \frac{n}{2} \rfloor$. Compute recursively $\overline{w'} = w'_1 \cdots w'_{m_1}$ and $\overline{w''} = w''_1 \cdots w''_{m_2}$. If the word $\tilde{w} = \overline{w'w''}$ is not freely reduced, the only place where cancellation can occur is between the end of $\overline{w'}$ and the beginning of $\overline{w''}$. For $i = 1, \cdots, n$, check whether $w'_{m_1 - i} = (w''_{1+i})^{-1}$. Stop as soon as there is an i for which this does not hold. Then set $\overline{w} = w'_1 \cdots w'_{m_1 - i} w''_{i+1} \cdots w''_{m_2}$.

Splitting w produces a tree of height $\log n$. At level k in the tree, any two words $w^{(k_1)}$ and $w^{(k_2)}$ are of length at most $\lfloor \frac{n}{2^k} \rfloor + 1$. So combining them will involve cancelling no more than $\lfloor \frac{n}{2^k} \rfloor + 1$ terms. There are 2^k words to be pairwise concatenated on a given level k. Hence pairwise combining (concatenating and taking care of any free cancellation in the middle) all the words on level k can be done in at most $2^k \left(\lfloor \frac{n}{2^k} \rfloor + 1 \right) = n + 2^k$ steps. There are $\log n$ levels, so the overall complexity of this procedure is

$$n \log n + \sum_{k=1}^{\log n} 2^k = n \log n + 2^{\log n + 1} = n \log n + n \in O(n \log n),$$

where n is the length of the input word. □

Note that, for the Turing model of computation, this is a tight bound, i.e., one cannot do better. To see this intuitively, it is enough to note that even adding two numbers of length n takes $O(\log n)$ steps. Hence even finding normal forms in free abelian groups, which must involve counting powers, cannot be done in less that $O(n \log n)$ steps. Knowing how to compute normal forms renders solving the Word Problem trivial.

Corollary 3.1.3. *The Word Problem in free groups is decidable in time $O(n)$, where n is the length of the input word.*

Proof. Given a word w, compute $\overline{w}$, which by Proposition 3.1.1 can be done in time $O(|w|)$. It is obvious from the definition of F that $\overline{w}$ is the empty word if and only if $w = 1$ in F. □

Deciding the Conjugacy Problem in free groups also depends on computing normal forms, but requires more work. However, the time complexity is the same.

Proposition 3.1.4. *The Conjugacy Problem in free groups can be decided in time $O(n)$, where n is the length of the input word.*

Proof. First one needs to introduce the notion of *cyclically reducing* a freely reduced word. A freely reduced word $u = u_1 \cdots u_n$ is cyclically reduced if $u_1 \neq u_n^{-1}$. Observe that u is said to be *cyclically reduced* if and only if u^2 is freely reduced. Indeed, if $u_1 \neq u_n^{-1}$ and $u_i \neq u_{i+1}$ for all $1 \leq i \leq n$, then the word $u^2 = u_1 \cdots u_n u_1 \cdots u_n$ is freely reduced. Conversely if u^2 is reduced, it is obvious that no cancellation occurs and in particular $u_1 \neq u_n^{-1}$. Using this fact, it is easy to cyclically reduce u. Let $k = \frac{1}{2}\big(2|u| - |\overline{u^2}|\big)$ and let $\tilde{u} = u_{k+1} \cdots u_{n-k}$. This can be done in time $O(n)$. Clearly, $\tilde{u}$ is cyclically reduced, since k is the length of the cancellation in u^2.

Now, note that two cyclically reduced words u and v are conjugate if and only if one is a cyclic permutation of the other. To see this, suppose that $a^{-1}ua = v$ for some freely reduced word $a \in F$. Since v is cyclically reduced, then so is $a^{-1}ua$, so either u starts with a or ends with a^{-1}, i.e., either $u = au_1$, or $u = u_2 a$. In the case where $u = au_1$,

$$v = a^{-1}ua = a^{-1}au_1a = u_1a,$$

which is a cyclic permutation of u. The case where $u = u_2a$ is very similar.

Finally, it remains to check whether v is a cyclic permutation of u. The naive approach of searching for the beginning of u in v and then checking whether the rest of u appears in v will take quadratic time. This is a famous problem, known in the literature as "string matching" and there exists an algorithm which can solve it in time $O(n)$, where $n = |u| + |v|$. □

3.2 Automatic and Bi-automatic Groups

As the name suggests, automatic and bi-automatic groups are related to finite state automata. A quick review of automata will be followed by the definition of automatic groups and the outline of polynomial time algorithms for solving the word and conjugacy problems in them.

Definition 3.2.1. *A* finite state automaton *is a tuple* (S, X, δ, S_A, s_0), *where*

1. *S is a finite set, calle the set of* states*,*
2. *X is a finite set, called the* alphabet*,*
3. *δ is a function $S \times X \to S$, called the* transition function*,*
4. *$S_A \subseteq S$ is the set of* accept states*, and*
5. *$s_0 \in S$ is the* initial state*.*

Given a word over the alphabet X, the automaton starts in state s_0 and reads the first letter x of the word. According to the transition function on that symbol, the automaton moves to another state $\delta(s_0, x)$ and continues reading the word, changing state at every symbol. When the end of the word is reached, if the automaton is in an accept state, the word is said to be *accepted* by the automaton. The notion generalises to *languages* (which are simply sets of words) – a language is accepted by an automaton if every word in the language is accepted by the automaton. Usually, all inaccessible states (i.e., the ones to which there is no directed path starting at s_0) are deleted from the automaton. Further, all dead states (i.e., states which are not in the domain of δ) are combined into one. This is sometimes referred to as a *normalised* automaton. In the sequel, all automata are assumed to be normalised. This is does not affect the complexity of the presented algorithms.

Definition 3.2.2. *A language L is* regular *if it is* recognised *by a finite state automaton, i.e., if there is an automaton which accepts every word in L and does not accept any word in L^C (the complement of L).*

Conversely, denote the language recognised by a finite state automaton, S, by $L(S)$. Thus every regular language has a finite state automaton associated with it. Regular languages are closed under union, intersection and complementation, which one proves by combining automatons. Sometimes, in order to do this sort of construction, one needs some words to have the same length. This is usually achieved through padding.

Definition 3.2.3. *Let $X_1, \dots, X_n$ be alphabets. Choose a character, $\$ \notin X_i$ for all i and add it to each X_i. Let $Y_i = X_i \cup \{\$\}$. The* padded alphabet *$P(X_1 \times \dots \times X_n)$ corresponding to the alphabets $X_1, \dots, X_n$ is*

$$P(X_1 \times \ldots \times X_n) = Y_1 \times \cdots \times Y_n \setminus \{(\$, \ldots, \$)\}.$$

Let $w_1, \ldots, w_n$ be strings over the alphabets $X_1, \ldots, X_n$, respectively and let $n_i = |w_i|$. Denote $k = max_{i=1}^{n}\{n_i\}$. Then $w = (w_1, \ldots, w_n)$ is a padded string *if whenever $n_i < k$, $w_i(n_i + 1), \ldots, w_i(k) = \$$, i.e., whenever w_i is shorter that the longest word, it is padded to the same length.*

For two alphabets X and Y, associate $(X \times Y)^*$ with $X^* \times Y^*$ via componentwise concatenation. In other words given a word $(x_1, y_1), \ldots (x_n, y_n) \in (X \times Y)^*$, associate with it the word $x_1 \cdots x_n y_1 \cdots y_n \in X^* \times Y^*$.

Recall that we say a set X generates a group G if the map $\pi : F(X) \to G$ is surjective. Consider the more general case, where instead of $F(X)$ one considers a regular language $L(X)$ over the alphabet X.

Definition 3.2.4. *Given a regular language $L(X)$, if the homomorphism $\pi : L(X) \to G$, defined in the natural way is surjective, then G is said to be* regularly generated *by $L(X)$.*

Definition 3.2.5. *Let G be a group with a set of generators X. An* automatic structure *on G consists of:*

1. *a finite state automaton W on $X^{\pm}$, called* the word acceptor*, such that the map $\pi : L(W) \to G$ is surjective.*
2. *for each $x \in X^{\pm} \cup \{1\}$ an automaton M_x over $(X^{\pm}, X^{\pm})$, called* multiplier automaton*, such that, for $x \in X^{\pm} \cup \{1\}$, $(w_1, w_2) \in L(M_x)$ if and only if*
   - *(a) $w_1, w_2 \in L(W)$, and*
   - *(b) $\overline{w_1 x} = \overline{w_2}$*

M_1 is called the equality recogniser*. An automatic structure is typically denoted by $\mathcal{S} = (X, L)$, without mention of W or M.*

It is known (see for example, Theorem 2.4.1 in [5]) that if G admits an automatic structure for one set of generators, it admits an automatic structure for any other set of generators. Hence one can give the following definition.

Definition 3.2.6. *A group G is* automatic *if there exists an automatic structure for G.*

Automatic groups satisfy a sort of Lipschitz property. This notion originates in analysis but comes up very naturally in the context of Cayley graphs and the word metric.

Lemma 3.2.7. *Let $\mathcal{S}$ be an automatic structure for G. Then, there exists a constant k such that if $(w_1, w_2) \in L(M_x)$ for some $x \in X^{\pm} \cup \{1\}$, then for any $t \geq 0$*

$$d\big(\overline{w_1(t)}, \overline{w_2(t)}\big) \leq k.$$

The constant k is called a Lipschitz constant *for $\mathcal{S}$.*

A good proof can be found in [5]. It is worth pointing out at this point, that the main reason why automatic, and by the same token bi-automatic (defined later in this section), groups are of particular interest is that one can derive from the structure many kinds of bounds. Of course, having a bound on word length, or on distance, for instance, is helpful in obtaining a bound on complexity.

Lemma 3.2.8 (Bounded length property)**.** *Let $\mathcal{S}$ be an automatic structure for G. There is a constant N such that if*

1. *$w \in L$ is an accepted word, and*
2. *$g \in G$ is a vertex in the Cayley graph of G, with $d(\bar{w}, g) \leq 1$,*

then

1. *there is a* $v \in L$ *such that* $\bar{v} = g$ *and* $|v| \leq |w| + N$*, and*
2. *if there is some* $v' \in L$ *such that* $\bar{v'} = g$ *and* $|v'| > |w| + N$*, then* g *has infinitely many representatives in* L*.*

Again, a proof of this can be found in [5]. In particular, this means that any word w in generators X has a representative of length at most $N|w| + n_0$, where n_0 is the length of a representative of the identity.

Theorem 3.2.9. *Let* $G = \langle X \rangle$ *be an automatic group. Then there is an algorithm that decides the Word Problem in* G *in quadratic time.*

Proof. Let $\mathcal{S} = (X, L)$ be an automatic structure for G. The idea is to get a 'normal form' for the given word and then compare it to the identity. More precisely, let w be a word over $X^{\pm}$. Then one can find a string in L representing w in quadratic time. This can then be compared to the identity using the equality recogniser and thus getting a solution to the Word Problem.

Suppose that $w = \overline{ux}$, where $u \in L$ and $x \in X$ and proceed to find a representative v for w. The pair (u, v) is accepted by M_x, so one can consider all possible transitions in M_x to find v. Ignoring the second element of the labels in M_x, find a path to an accept state labelled by u (in the first component only). Reading the second component, going back along this path gives a representative for v. The crucial point is finding this path in quadratic time, since simply considering all paths labelled by u will give an exponential procedure. For this, one needs to build a data structure as follows.

Let $M_x = (S, X, \delta, S_A, s_0)$. Define inductively $S_0 = \{s_0\}$ and

$$S_i = \{R_i \mid \delta(S_{i-1}, (x_i, y_i)) = R_i, y_i \in X\},$$

where $x_i = u(i)$ if $i \leq |u|$ and $x_i = \$$ otherwise.

Let $n \geq |u|$ be the smallest number such that $S_n \cap S_A \neq \varnothing$ (i.e., S_n contains an accept state). Backtracking along the path from S_0 to an accept state in S_n, read the labels $y_n, \cdots, y_1$. Let $v = y_1 \cdots y_n$ after discarding all \$ at the end of the string. In the case $w = \overline{ux^{-1}}$, a representative can be found using the same procedure replacing (x_i, y_i) by (y_i, x_i)

There are finitely many states and transitions, so the time taken in each induction step above is bounded by a constant, so the time taken to find a representative is proportional to n (as defined above). Now, by Lemma 3.2.8, $n \leq |u| + N$, where N is a constant for the structure. Thus, one can find a representative of length at most $|u| + N$ in time $O(|u|)$.

Now repeat this process to get a representative of w of length at most $N|w| + n_0$, where n_0 is the length of a representative of the identity. This can be done in time

$$\sum_{i=1}^{|w|} \left(iN + n_0\right) \in O\left(|w|^2\right).$$

Finding a representative for the identity can be done using the same procedure. Start with any accepted string $x_{i_1} \cdots x_{i_k}$ and using the above procedure find consecutively representatives for $x_{i_1} \cdots x_{i_{k-1}} x_{i_k} x_{i_k}^{-1}, \ldots, x_{i_1} \cdots x_{i_{k-1}} x_{i_k} x_{i_k}^{-1} \cdots x_{i_1}^{-1}$. This is done as a pre-computation for the algorithm deciding the Word Problem, so computing this is done in constant time.

Finally, observe that, having a representative of the identity, one can obtain a representative for any word $w = w_1 \cdots w_k$ by using the above procedure to find consecutive representatives of any word in the group.

□

Strangely enough, it is still unknown whether automatic groups have decidable Conjugacy Problem. However, it has been shown by Gersten and Short in [6] that bi-automatic groups have decidable Conjugacy Problem.

Definition 3.2.10. *Let $\mathcal{S} = (X, W, \{M_x\}_{x \in X})$ be an automatic structure for the group G. $L(W)$ is the language of a* bi-automatic structure *on G if in addition, for each $x \in X$, the language*

$$L_x = \{(u, v) \in L \times L \mid \overline{u} = \overline{xv}\}$$

is regular over the padded alphabet $P(X \times X)$.

Bi-automatic groups can be thought of as automatic groups which also have left-multipliers. This allows to consider multiplication on both sides, which is all that is necessary to solve the Conjugacy Problem. The following theorems are due to Gersten and Short and full proofs can be found in [6].

Lemma 3.2.11. *Let L be the language of a bi-automatic structure for $G = \langle X \rangle$. Then for any $g, h \in (X^{\pm})^*$, the language*

$$L(g, h) = \{(u, v) \in L \times L \mid \overline{u} = \overline{gvh}\}$$

is regular.

Theorem 3.2.12. *A group G with bi-automatic structure has decidable Conjugacy Problem.*

Proof. Let G be a group with bi-automatic structure $\mathcal{S} = (X, L)$ and let g and h be words over X. By Lemma 3.2.11, the language $L(g, h)$ is regular. Now if the elements $\overline{g}$ and $\overline{h}$ are conjugate in G, there is an element $r \in G$ such that $r^{-1}gr = h$, i.e., $gr = rh$. Hence there is a word $w \in L$ such that $\overline{gw} = \overline{wh}$. The latter is true if and only if $\overline{w} = \overline{g^{-1}wh}$. Equivalently, g and h are conjugate if and only if

$$L(g^{-1}, h) \cap \{(w, w) \mid w \in L\} \neq \varnothing.$$

An algorithm to decide whether the intersection of two regular languages is non-empty is given in [9]. □

Unfortunately, the complexity of the algorithm described above is exponential and no better one is known so far.

3.3 Hyperbolic Groups

Free groups are universally loved, not only because they have a universal property, but also because many interesting algorithmic problems are easily solvable there. The properties that entail this are captured by the geometry of their Cayley graphs. Indeed the Cayley graph of a free group is a tree – a very special kind of graph. Hyperbolic groups are a generalisation of free groups, in the sense that their Cayley graphs "seen from far away" look like trees. Taking the geometric view-point, this means that the Cayley graph of a hyperbolic group is quasi-isometric to a tree. A more precise definition will follow. Thus the guiding idea is that anything that can be done for free groups can, with some modifications and significantly more work, also be done for hyperbolic groups. However, the class of hyperbolic groups is much larger and includes a number of interesting groups such as finite groups, virtually cyclic groups, most surface groups (the ones having negative Euler characteristic), most triangle groups, fundamental groups of compact Riemannian manifolds with strictly negative sectional curvature, and, more generally, groups that act cocompactly and properly discontinuously on a CAT(k) space with $k < 0$.

Let G be group with generating set X and let Γ be its Cayley graph with the usual metric. The basic building block of a tree is a tripod. The next best thing is a triangle. However, while a tripod between three points is unique, a triangle need not be. Hence one needs a more specific notion of a triangle.

Definition 3.3.1. *A triangle is called* geodesic *if each of its sides is a geodesic in the Cayley graph.*

The idea of looking at the graph "from far away" is captured by the notion of *thinness* of a triangle.

Definition 3.3.2. *A triangle ABC is δ-*thin *if given any point P on any side AB of the triangle, there is a point Q lying either on BC or on AC such that*

$$d(P, Q) \leq \delta.$$

In other words, given a point on one of the sides, it must be within distance δ of one of the other sides. Notice that the smaller δ is, the more the graph looks like a tree. In fact, free groups are hyperbolic with $\delta = 0$, since any geodesic triangle there is a tripod.

A group G is called *hyperbolic* if it is δ-hyperbolic for some δ. As stated, the definition of hyperbolicity depends on the set of generators, since it depends on the geometry of the Cayley graph for this particular set of generators. However, it is known that if a group is hyperbolic for some finite set of generators, it is hyperbolic for any other finite set of generators (maybe with a different δ).

As hyperbolic groups can be thought of as a generalisation of free groups, it is not surprisingly the general strategy for attacking algorithmic problems there is very similar to the one in free groups – find suitable normal forms and use them to solve the given problem.

There are various algorithms to solve the Conjugacy Problem in hyperbolic groups. Notably, one that seems to work well in practice is due to Marshall (see [13]). A better algorithm, in terms of theoretical complexity is a 'linear'one presented by Epstein and Holt in [4]. It is indeed linear with respect to a RAM model of computation, but its complexity is $O(n \log n)$, where n is the size of the input, in the Turing model of computation. It should be noted that the decidability of the Conjugacy Problem in hyperbolic groups was already shown by Gromov in [8]. Moreover, since hyperbolic groups are known to be bi-automatic, the result of Gersten and Short discussed in Section 3.2 also implies the decidability of the Conjugacy Problem there. The later both have exponential time complexities. Bridson and Haefliger also give a cubic time algorithm in Section 2.12 of Part III.Γ of [2] which can be improved to a quadratic one. A related result is that of Inna Bumagin (see [3]) gives an algorithm to decide the Conjugacy Problem in relatively hyperbolic groups.

The algorithm due to Epstein and Holt is currently the one with the best time complexity. Not surprisingly, the idea is to mimic the algorithm for free groups. Given two words u and v, first one attempts to fin *normal forms* in the form of *short-lex* geodesics. Then one tries to find a power of u in a power of v. The former is obtained by using what the authors refer to as "Shapiro's algorithm" to find short-lex geodesics in linear time. Both parts are very technical, and involve a careful adjustment of the constants that cannot be meaningfully summarised in a few paragraphs.

3.4 Wreath Products

Let A and B be groups. Denote by $A^{(B)}$ the set $\mathrm{fun}(B, A)$ of functions from B to A with finite support. In other words,

$$A^{(B)} = \{f : B \to A \mid |\mathrm{supp}(f)| < \infty\},$$

where

$$\mathrm{supp}(f) = \{b \in B \mid f(b) \neq 1\}.$$

Then $A^{(B)}$ is a group with multiplication given by

$$(fg)(b) = f(b)g(b).$$

The inverse $f^{-1} \in A^{(B)}$ is given by $f^{-1}(b) = \big(f(b)\big)^{-1}$ and the identity in the group is the map $b \mapsto 1_A$ for every $b \in B$. It is important to emphasise that functions in $A^{(B)}$ need not be homomorphisms.

Definition 3.4.1. *The* restricted wreath product $A\mathrm{wr}B$ *is the group formed by the set of pairs*

$$A\mathrm{wr}B = \{bf \mid b \in B,\ f \in A^{(B)}\},$$

with multiplication defined by

$$bfcg = bcf^{c}g,$$

where $f^c(x) = f(xc^{-1})$ *for* $x \in B$.

Letters $b, c, d, \ldots$ are used for elements of B and letters $f, g, h, \ldots$ for functions in $A^{(B)}$. It is easy to see that $A\mathrm{wr}B$ is a group, with identity $1_{A\mathrm{wr}B} = 1_B 1_{A^{(B)}}$ and inverses $(bf)^{-1} = b^{-1}(f^{-1})^{b^{-1}}$. Indeed,

$$bfb^{-1}(f^{-1})^{b^{-1}} = bb^{-1}f^{b^{-1}}(f^{-1})^{b^{-1}} = 1_B 1_{A^{(B)}},$$

since for any $c \in B$,

$$\left(f^{b^{-1}}(f^{-1})^{b^{-1}}\right)(c) = f^{b^{-1}}(c)(f^{-1})^{b^{-1}}(c) = f(cb)f^{-1}(cb^{-1}) = 1_A.$$

One can also view the restricted wreath product $A\mathrm{wr}B$ as the semi-direct product $A^{(B)} \rtimes B$. However this line of thought is not pursued in the sequel. The *unrestricted wreath product* is defined in the same way as the restricted one, except that the finite support requirement on the functions is removed. This makes it unsuitable to be considered in the algorithmic context, as even writing down most functions would be impossible.

As usual, in order to even formulate an algorithmic problem, one needs to have a set of generators in mind. Let $X = \{x_1, \cdots, x_n\}$ and $Y = \{y_1, \cdots, y_m\}$ be the generating sets for A and B, respectively.

Define functions $f_{a_i, b_i} \in A^{(B)}$ as

$$f_{a_i,b_i}(x) = \begin{cases} a_i & \text{if } x = b_i \\ 1 & \text{otherwise.} \end{cases}$$

For simplicity, we denote $f_{a_i,1}$ by f_{a_i}. Then any function $f \in A^{(B)}$ with $\mathrm{supp}(f) = \{b_1, \cdots, b_k\}$ and $f(b_j) = a_j$ can be written as

$$f = \prod_{b_i \in \mathrm{supp}(f)} f_{a_i,b_i}.$$

Observe that A embeds in $A^{(B)}$ via the map $a \mapsto f_a$. In the sequel, by abuse of notation, A will be identified with its image under this map. Observe further that $b^{-1} f_a b = f_{a,b}$, since

- $(b^{-1} 1_{A^{(B)}})(1 f_a)(b 1_{A^{(B)}}) = (b^{-1} 1_{A^{(B)}})(b f_a^b) = 1_{A^{(B)}} f_a^b$ in $A\mathrm{wr}B$.
- $f_a^b(x) = f_a(b^{-1}x) = f_{a,b}(x)$.

Thus a function $f \in \mathrm{fun}(B, A)$ can be written as

$$f = \prod_{b_i \in \mathrm{supp}(f)} f_{a_i,b_i} = \prod_{b_i \in \mathrm{supp}(f)} f_{a_i}^{b_i} = \prod_{b_i \in \mathrm{supp}(f)} a_i^{b_i}.$$

Hence, $A\mathrm{wr}B$ is generated by X, Y. In particular, that means that if both A and B are finitely generated, so is $A\mathrm{wr}B$.

Note that the above reasoning works regardless whether the action is defined as $f^b(x) = f(b^{-1}x)$ or $f^b(x) = f(xb^{-1})$. In order to conform to other authors' conventions, the former is used in Section 5 and the latter in Section 6.1.

Usually, when dealing with algorithmic problems, the input word is given as a product of generators. However, in the present case, it is not convenient to use this presentation, so the input word is reprocessed in the first part of the algorithm. One can rewrite a word $w = y_1 x_1 \cdots y_k x_k$ in generators X and Y as $w = bf$ in polynomial time. For simplicity, assume that y_i and x_i are single generators. In general, they will be words in X and Y, but the proof is exactly the same. By applying conjugation

$k-1$ times we may obtain

$$\begin{aligned} w &= y_1y_2(y_2^{-1}x_1y_2)x_2\cdots y_kx_k = y_1y_2x_1^{y_2}x_2y_3x_3\cdots y_kx_k \\ &= y_1y_2y_3(y_3^{-1}x_1^{y_2}x_2y_3)x_3\cdots y_kx_k = y_1y_2y_3(x_1^{y_2}x_3)^{y_3}x_4\cdots y_kx_k \\ &= y_1y_2y_3x_1^{y_2y_3}x_2^{y_3}x_4\cdots y_kx_k = \cdots \\ &= y_1\cdots y_kx_1^{y_2\cdots y_k}x_2^{y_3\cdots y_k}\cdots x_{k-1}^{y_k}x_k. \end{aligned}$$

The idea is that here $b = y_1\cdots y_k \in B$ and $f = x_1^{y_2\cdots y_k}x_2^{y_3\cdots y_k}\cdots x_{k-1}^{y_k}x_k \in A^{(B)}$. However, the powers represent the domain elements of the function. In order to have a well-defined function, one needs to make sure none of the powers are secretly equal. Denote $Y_i = y_i\cdots y_k$. For each $2 \le i < j \le k$, check whether $Y_i = Y_j$. This amounts to solving $\binom{k-1}{2}$ Word Problems in B. For each $Y_{i_1} = Y_{i_2} = \cdots = Y_{i_j}$, set $f(Y_{i_1}) = x_{i_1}\cdots x_{i_j}$. This determines f completely and we can change presentations in time $O(|w|^2T_{WB}(|w|))$, where T_{WB} is the time function for the Word Problem in B.

Note that if a word is given as a product of generators, converting it to standard (or pair) form gives an ordering for $\text{supp}(f) = \{Y_i\}_i$ determined by the indices i. More precisely, $Y_i < Y_j$ whenever $i < j$.

With this, it is easy to solve the Word Problem in a wreath product $A\text{wr}B$.

Proposition 3.4.2. *If the word problems in A and B are decidable in polynomial time, so is the Word Problem in $A\text{wr}B$.*

Proof. Let T_{WB} and T_{WA} be the time functions for algorithms which solve the Word Problems in B and in A, respectively. Given a word w in generators X, Y, rewrite it in the form $w = bf$, as described above. Then $w = 1$ if and only if $b = 1$ and $f = 1$. To check the former requires time T_{WB}. To see whether $f = 1$, one needs to check that every image point is trivial, i.e., one needs to solve at most $|w|$ word problems in A, which can be done in time $|w|T_{WA}$. Thus, the Word Problem has complexity

$$O\Big(|w|^2 T_{WB}(|w|) + |w|T_{WA}(|w|)\Big).$$

□

The Conjugacy Problem, however, is not quite so simple. Jane Matthews showed in [14] that the Conjugacy Problem in wreath products is decidable. The complexity, however, was not known until now. The proof Matthews gave is not presented here, but rather in Section 6.1, where it is shown that the Conjugacy Problem is decidable in polynomial time.

In fact the complexity of the Conjugacy Problem is not known in any of the examples presented in Section 4. In the sequel, a polynomial time algorithm is given to solve the Conjugacy Problem in free solvable groups.

CHAPTER 4
Solvable Groups and Algorithmic Problems Therein

Solvable groups are in a sense a generalisation of abelian groups. Thus one would expect them to share many of their 'nice' properties. This is often the case algebraically. However, with regard to computation, solvable groups present much more variety. In this section several classes of solvable groups and the Word and Conjugacy problems in them are discussed. We start with several basic definitions.

Definition 4.0.3. *Let G be a group. The* commutator *of $g, h \in G$ is*

$$[g,h] = g^{-1}h^{-1}gh.$$

The elements g and h commute*, i.e., $gh = hg$, if and only if $[g,h] = 1$.*

Thus commutators, measure how far two elements are from commuting. Given groups G and H, their commutator group is defined as $[G,H] = \langle [g,h] \mid g \in G, h \in H \rangle$.

Definition 4.0.4. *Let G be a group. Define its* derived series *to be the normal series*

$$G \geq G' \geq G^{(2)} \geq \cdots \geq G^{(n)} \geq \cdots$$

where $G' = [G,G]$ and $G^{(n+1)} = [G^{(n)}, G^{(n)}]$.

Definition 4.0.5. *A group G is* solvable *if its derived series reaches the trivial group, i.e., if there exists an n such that*

$$G^{(n)} = 1.$$

The smallest such n is called the degree of solvability *of G.*

Solvable groups present a large class of groups, and it is impossible to solve most interesting algorithmic problems for the whole class. This follows from a result of O. Kharlampovich. In [10] she exhibits a finitely presented solvable group which has unsolvable Word Problem. Now, a solution to the Conjugacy Probem or to the Bounded Geodesic Length Problem, gives a solution to the Word Problem (check whether a word is conjugate to the identity, or check whether a word has a geodesic of length 1). Thus, the undecidability of the Word Problem implies the undecidability of many other interesting problems.

However, a great deal is known about various subclasses of solvable groups. The rest of this section explores the word and conjugacy problems in several of them.

4.1 Finitely Generated Metabelian Groups

Just as a commutator is a measure of how far two elements are from commuting, the length of the derived series determines how far a group is from being abelian. If the series is of length one, the group is abelian. A group which has derived length two is called *metabelian*.

Even this is in a sense the simplest non-trivial example of a class of solvable groups, algorithmic problems here are not easy to solve. Denote by $\mathcal{M}$ the variety of metabelian groups. Finitely generated metabelian groups are finitely presented in this variety (a well-known result of P.Hall).

Moreover, metabelian groups are residually finite, i.e., for any metabelian group M and for any $g \in M$, tehre is a homomorphism $\psi_g : M \to F_g$, where F_g is a finite group, such that $\psi_g(g) \neq 1$. It follows from the latter two facts that the Word

Problem for metabelian groups is decidable. Indeed, let M be a metabelian group and let w be a word over its generators. Run the following two procedures:

$(P1)$ Enumerate all finite quotients of M and for each check wether the image of w is non-trivial.

$(P2)$ Enumerate all consequences of the relations of M and compare each to w (as strings of generators).

Note that no time complexity can be deduced from this algorithm, as there is no way to bound the size of the enumerations in $(P1)$ and $(P2)$.

The idea for solving the Conjugacy Problem is to translate it in terms of module theory. Then the result follows using some powerful tools available there. The proof hinges on computing finite presentations of various submodules.

The notion of a finite presentation in a given variety of groups is a generalisation of the definition of a presentation given in Section 2.1.

Definition 4.1.1. *Let $\mathbb{V}$ be a variety of groups and let G be a finitely generated group in this variety. Then $G \simeq F/R$, where F is a finitely generated free group in $\mathbb{V}$ and R is some normal subgroup. If R is the normal closure in F of some finite subset, then this gives a finite $\mathbb{V}$-presentation of G.*

One of the most common examples is the variety $\mathcal{A}$ of finitely generated abelian groups. Consider the abelian group $\mathbb{Z}_6 \times \mathbb{Z}_7$ with presentation

$$\langle x, y \mid [x, y] = x^6 = y^7 = 1 \rangle.$$

It is also the quotient of the free abelian group $\mathbb{Z} \times \mathbb{Z}$ by the normal closure of the subset $\{(6,1),(1,7)\}$, i.e., $\mathbb{Z}_7 \simeq \mathbb{Z}/ << (6,1),(1,7) >>$. Thus the presentation of

$\mathbb{Z}_6 \times \mathbb{Z}_7$ in the variety of finitely generated abelian groups is

$$\langle x, y \mid x^6 = y^7 = 1 \rangle_{\mathcal{A}}.$$

Finitely generated metabelian groups are finitely presented in the variety of metabelian groups and so lend themselves better to computation, as their presentations can be expressed in terms of matrices and questions can be reduced to equivalent questions in module theory, where more machinery is available.

Proposition 4.1.2. *Let G be a finitely generated metabelian group. Then G' is a (left) $\mathbb{Z}\left(G/G'\right)$-module.*

Proof. Note that since G is metabelian, its derived series is of length two and so G' is abelian. Define a G/G'-action on G' via $g \cdot a = gag^{-1}$. This action is well defined, since if $gG' = hG'$, then $g = ha_1$ for $a_1 \in G'$ and consequently

$$g \cdot a = gag^{-1} = ha_1 a a_1^{-1} h^{-1} = hah^{-1} = h \cdot a.$$

This action clearly extends to a $\mathbb{Z}\left(G/G'\right)$ which defines scalar multiplication for hte module. □

Now that groups have been translated into module language, the next step is to be able to compute with modules.

Definition 4.1.3. *A ring R is* computable *if its elements can be enumerated by a Turing machine.*

Definition 4.1.4. *Let M be a finitely generated R-module, where R is a computable ring satisfying max-n (the maximal condition on normal subgroups) and suppose M*

is given by a finite presentation. Then M is submodule computable *if there exist algorithms to*

1. *find a finite R-presentation for the submodule N generated by a given finite subset of M;*
2. *decide whether an element w over generators of M is in the submodule N.*

The submodule computability of certain modules is shown in [11] and is used in the proof of some crucial lemmas, including the module theoretic version of the Conjugacy Problem.

Lemma 4.1.5. *Let Q be a finitely generated abelian group and let M be a finitely generated $\mathbb{Z}Q$-module. Then there is an algorithm which, given two elements $x, y \in M$, decides whether there is an element $g \in Q$ such that $x = g \cdot y$, where $\cdot$ denotes scalar multiplication in the module (given by the action of Q).*

A proof of this can be found in [11]. They also present a concise proof of Noskov's result. A slightly more detailed version of their proof is given here.

Theorem 4.1.6 (Noskov)**.** *The Conjugacy Problem in finitely generated metabelian groups is decidable.*

Proof. Notice that if x and y are conjugate in G, then for some $g \in G$, $x = g^{-1}yg$ and so

$$xy^{-1} = g^{-1}ygy^{-1} = [g, y^{-1}] \in G'.$$

Check that $xy^{-1} \in G'$. This is equivalent to checking that $xy^{-1} = 1$ in G/G', which is abelian and so has decidable Word Problem. If $xy^{-1} \notin G'$, they are not conjugate, so assume $xy^{-1} \in G'$.

Observe also that x and y are conjugate in G if and only if they are conjugate in $G/[G', x]$. Indeed if x and y are conjugate in $G/[G', x]$, then there is some $g \in G/[G', x]$ and some $a \in G'$, such that

$$y^g = g^{-1}yg = x[x, a] = xx^{-1}a^{-1}xa = a^{-1}xa = x^a.$$

Now, by Proposition 4.1.2, G' is a $\mathbb{Z}(G/G')$ module and a finite $\mathbb{Z}(G/G')$-presentation for it can be found (see for example Lemma 9.5.3 in [11]). Suppose that G' is generated by $\{g_1, \ldots, g_m\}$ as a $\mathbb{Z}(G/G')$-module. Then, since G' is abelian, $[G', x]$ is generated as a $\mathbb{Z}(G/G')$-module by $\{[g_1, x], \ldots, [g_m, x]\}$. Hence a finite presentation of $G/[G', x]$ can be found in the variety of metabelian groups. Thus, assume without loss of generality that $[G', x] = 1$ (otherwise, simply replace G by $G/[G', x]$).

Let $M = \langle x, G' \rangle$. Since G' is normal and for any $g \in G$

$$g^{-1}xg = g^{-1}xgx^{-1}x = [g, x^{-1}]x \in M,$$

so M is a normal subgroup of G. Moreover, G' is abelian and $[G', x] = 1$, so M is also abelian. Note that $y \in M$, since $xy^{-1} \in G'$. Hence M is an abelian normal subgroup of G, containing both x and y. Furthermore, M is a finitely generated $\mathbb{Z}(G/G')$-module and a presentation for it can be obtained by adding all conjugates of x by generators of G (and their inverses) to the presentation of A. Finally, by applying Lemma 4.1.5 to M, one can decide whether x and y are conjugate in G, since G/G' is actually acting by elements of G. □

4.2 Free Solvable Groups

Let F be the free group of rank r, i.e., let $F = F(X)$ for $X = \{x_1, \ldots, x_r\}$. Let N be a normal subgroup of F. Recall that

$$N' = [N, N] = \langle [g, h] \mid g, h \in N \rangle$$

is called the *derived subgroup* (or *commutator subgroup*) of N. Note that N' is itself normal in F. Indeed, for any $f \in F$ and for any $[g, h] \in N'$,

$$\begin{aligned} [g,h]^f &= f^{-1}ghg^{-1}h^{-1}f = f^{-1}gff^{-1}hff^{-1}g^{-1}ff^{-1}h^{-1}f \\ &= g^f h^f (g^{-1})^f (h^{-1})^f \\ &= [g^f, h^f] \in N'. \end{aligned}$$

Consider the special case $N = F$. As shown above F' is normal in F. Thus one can in turn take its derived subgroup. Recall that

$$F^{(d+1)} = [F^{(d)}, F^{(d)}].$$

The normal series

$$F \rhd F' \rhd \cdots \rhd F^{(d)} \rhd \cdots$$

is called the *derived series* of F. Naturally, given a normal series one is tempted to take the quotient of the terms. The group $F/F^{(d)}$ is called the *free solvable group* of rank r and class d.

Remark. *Note that $F/F^{(d)}$ is indeed a solvable group. To see this, observe that taking the quotient of each term by $F^{(d)}$ the derived series of F by $F^{(d)}$ yields the* *normal series*

$$F/F^{(d)} \rhd F'/F^{(d)} \rhd \cdots \rhd F^{(d)}/F^{(d)} = 1.$$

Further, for each $1 \leq i \leq d$, $F^{(i)}/F^{(d)} \simeq \left(F/F^{(d)}\right)^{(i)}$, *since for any* $x, y \in F^{(i)}$,

$$\begin{aligned}
[xF^{(d)}, yF^{(d)}] &= xF^{(d)}yF^{(d)}x^{-1}F^{(d)}y^{-1}F^{(d)} = xF^{(d)}yF^{(d)}x^{-1}F^{(d)}y^{-1}F^{(d)}yy^{-1} \\
&= xF^{(d)}yF^{(d)}x^{-1}F^{(d)}y^{-1} = xF^{(d)}yF^{(d)}x^{-1}y^{-1} \\
&= xF^{(d)}yx^{-1}y^{-1} = F^{(d)}xyx^{-1}y^{-1} \\
&= [x,y]F^{(d)}.
\end{aligned}$$

Here each step uses the normality of $F^{(d)}$ *in* $F^{(i)}$. *It follows that* $F^{(i)}/F^{(d)} \simeq \left(F/F^{(d)}\right)^{(i)}$ *and the normal series becomes*

$$F/F^{(d)} \rhd \left(F/F^{(d)}\right)' \rhd \cdots \rhd \left(F/F^{(d)}\right)^{(d-1)} \rhd 1.$$

The abelian group F/F' is called the *abelianisation* of F and $F/F^{(2)}$ is the *free metabelian group* of rank r.

The Word Problem in free solvable groups was shown to have cubic complexity by Miasnikov, Roman'kov, Ushakov and Vershik in [16] and Remeslennikov and Sokolov showed in [17] that the Conjugacy Problem is decidable, though without mentioning an analysis of time complexity. The former will be discussed in much more detail in Section 5.5.1. It is worth noting that the cubic algorithm of [16] is uniform for the class of free solvable groups. Both results provide an extensive source of ideas for the present work.

CHAPTER 5
Free Solvable Groups and the Magnus Embedding

In this section we introduce the Magnus embedding and Free Differential Calculus as well as the techniques used to compute them efficiently. With this, we solve the Word Problem in free solvable groups in cubic time. The same methods are used in the next chapter to solve the Conjugacy Problem in free solvable groups in polynomial time.

5.1 Group Rings

Let G be a group and let R be a commutative ring with identity.

Definition 5.1.1. *The* group ring *of G over R is the set of all functions of finite support from G to R,*

$$RG = \text{fun}(G, R) = \{f : G \to R \mid |\text{supp}(f)| < \infty\},$$

where $\text{supp}(f) = \{g \in G \mid f(g) \neq 0\}$ *together with addition and multiplication defined as:*

1. $(f + h)(g) = f(g) + h(g)$
2. $(f \cdot h)(g) = \sum\limits_{xy=g} f(x)h(y)$

Note that multiplication is well-defined because the functions are of finite support, so the sums are finite.

Elements $u \in RG$ are often written as $u = \sum_{g\in G} r_g g$ with $r_g = 0$ for all but finitely many $g \in G$. This clearly corresponds to the function $f \in \text{fun}(G, R)$ given by $f(g) = r_g$. Then for $u = \sum\limits_{g\in G} r_g g, v = \sum\limits_{g\in G} s_g g \in RG$:

(1) $u + v = \sum\limits_{g\in G} (r_g + s_g)g$,

(2) $uv = \big(\sum\limits_{g\in G} r_g g\big)\big(\sum\limits_{h\in G} s_h h\big) = \sum\limits_{g,h\in G} r_g s_h gh = \sum\limits_{xy=g} r_x s_y g$.

It is clear from the above that RG is a ring with identity $1_{RG} = 1_R 1_G$.

Lemma 5.1.2. *A group homomorphism $\phi : G \to H$ induces by linearity a ring homomorphism $\phi : RG \to RH$.*

Proof. Given $\phi : G \to H$, define $\phi : RG \to RH$ by

$$\phi\Big(\sum_{g\in G} r_g g\Big) = \sum_{g\in G} r_g \phi(g).$$

It is easy to see that ϕ is a ring homomorphism. □

Definition 5.1.3. *The ring homomorphism $\epsilon : RG \to R$ induced by $\phi : G \to 1$ is called the* trivialisation *homomorphism. Its kernel, $\Delta = \ker\epsilon$, is called the* fundamental ideal *of RG.*

Proposition 5.1.4. *If X is a generating set for G, then $\Delta = id(\{x - 1 \mid x \in X\})$, where $id(Y)$ denotes the ideal generated by Y.*

Proof. Notice first that $\Delta = id(\{g - 1 \mid g \in G\})$. Indeed, if $u = \sum_{g\in G} r_g g \in \Delta$, then $0 = \epsilon(u) = \sum r_g$, so

$$u = u - 0 = \sum r_g g - \sum r_g = \sum r_g (g - 1).$$

Observe now that if $g, h \in G$, then

$$gh - 1 = gh - g + g - 1 = g(h - 1) + (g - 1). \tag{5.1}$$

Using this, show by induction on the word length of elements $g \in G$ that the ideal $\{g - 1 \mid g \in G\}$ is generated by $\bar{\Delta} = \{(x-1), (x^{-1} - 1) \mid x \in X\}$.

The base case is $g = x^{\pm 1}$ for some $x \in X$, so $g - 1 = x^{\pm 1} - 1$ and $g \in \bar{\Delta}$. Now suppose that the statement holds for all elements of length less than n and take $g \in G$ of length n. Then $g = hx^{\pm 1}$ for some $h \in G$ of length less than n and some generator $x \in X$. From equation (5.1),

$$g - 1 = hx^{\pm 1} - 1 = h(x^{\pm 1} - 1) + (h - 1).$$

Hence $g - 1$ is also in the set generated by $\bar{\Delta}$ and so $\Delta = id(\{(x-1), (x^{-1} - 1) \mid x \in X\})$. □

5.2 Free Solvable groups and the Magnus Embedding

Let $F = F(X)$ be a free group of rank r. Throughout, the guiding idea is to study F/N' via F/N. The following introduces the Magnus embedding, which gives a way to embed a free solvable group of rank $d+1$ into a matrix group depending on a free solvable group of degree d.

Denote by $\mu : F \to F/N$ the canonical epimorphism. Let T be the free $\mathbb{Z}(F/N)$-module with basis $\{t_1, \ldots, t_r\}$, i.e., $T \simeq \mathbb{Z}(F/N) \oplus \cdots \oplus \mathbb{Z}(F/N)$. Then the set of matrices

$$M(F/N) = \begin{pmatrix} F/N & T \\ 0 & 1 \end{pmatrix} = \left\{ \begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix} \middle| \begin{array}{c} g \in F/N \\ t \in T \end{array} \right\}$$

forms a group with respect to matrix multiplication. Indeed,

$$\begin{pmatrix} g_1 & u_1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} g_2 & u_2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} g_1g_2 & g_1u_2 + u_1 \\ 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} g^{-1} & -g^{-1}t \\ 0 & 1 \end{pmatrix}$$

The following lemma about the structure of the group $M(F/N)$ is crucial to the result on the Conjugacy Problem in free solvable groups in Section 6.2.

Lemma 5.2.1. $M(F/N) \simeq F/F'\mathrm{wr}F/N$.

Proof. Let $K = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$. Obviously, $K \cong T$. The strategy is to first show that $M(F/N) = K \rtimes F/N$ (see points 1, 2, 3) and then show that the action of F/N by conjugation on K corresponds to the action of F/N on $\mathrm{fun}(F/N, F/F')$ (see point 4).

1. $K \lhd M(F/N)$.

   Indeed, for any $\begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix} \in M(F/N)$ and for any $\begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix} \in K$,

$$\begin{aligned} \begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix}^{-1}\begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} g^{-1} & -g^{-1}t \\ 0 & 1 \end{pmatrix}\begin{pmatrix} g & t+u \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & g^{-1}u \\ 0 & 1 \end{pmatrix} \in K. \end{aligned}$$

2. $M(F/N)/K \simeq F/N$.

To see this, define a homomorphism $\psi : M(F/N) \to \begin{pmatrix} F/N & 0 \\ 0 & 1 \end{pmatrix} \simeq F/N$ by

$$\psi\left(\begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix}\right) = \begin{pmatrix} g & 0 \\ 0 & 1 \end{pmatrix}.$$

Obviously, $\psi\left(\begin{pmatrix} g & t \\ 0 & 1 \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ if and only if $g = 1$. Moreover, ψ is surjective, so $\ker \psi = K$ and by the first isomorphism theorem, $M(F/N)/K \cong F/N$.

3. $K \cong \mathrm{fun}(F/N, F/F')$ (as groups).

   Let $F/F' = \langle a_1, \dots, a_r \rangle$. Consider a function $f : F/N \to F/F'$. Then, since F/F' is an abelian group, one can write $f(x)$ in the normal form

$$f(x) = \sum_{i=1}^{r} n_x^{(i)} a_i,$$

   for some $n_x^{(i)} \in \mathbb{Z}$

   Consider the homomorphism $\lambda : \mathrm{fun}(F/N, F/F') \to T$, given by

$$\lambda(f) = \sum_{i=1}^{r} \left(\sum_{x \in \mathrm{supp}(f)} n_x^{(i)} x \right) t_i.$$

   Clearly, $\lambda(f) = 0$ if and only if, for every $1 \le i \le r$ and for every $x \in \mathrm{supp}(f)$, $n_x^{(i)} = 0$, i.e., $\lambda(f) = 0$ if and only if $f = 0$. Thus $\ker \lambda = 0$ and hence λ is injective.

   To see that λ is surjective, note that, since T is a free $\mathbb{Z}(F/N)$-module with basis $\{t_1, \dots, t_r\}$, one can write every element $t \in T$ as $t = \sum_{i=1}^{r} u_i t_i$, where $u_i \in$

$\mathbb{Z}(F/N)$. So one can in turn write $u_i = \sum\limits_{x \in F/N} n_x^{(i)} x$ with finitely many $n_x^{(i)} \neq 0$. Now define a function $f \in \mathrm{fun}(F/N, F/F')$ as

$$f(x) = \sum_{x \in F/N} n_x^{(i)} a_i.$$

This function is clearly well-defined, and

$$\lambda(f) = \sum_{i=1}^{r} \left(\sum_{x \in \mathrm{supp}(f)} n_x^{(i)} x \right) t_i = \sum_{i=1}^{r} u_i t_i = t,$$

so λ is also surjective. Hence, λ gives an isomorphism of abelian groups.

4. The action of F/N on K is the same as the action of F/N on $\mathrm{fun}(F/N, F/F')$.

It is easy to see this intuitively. To write it down, one needs to first get over the notation. The idea is to show the following diagram commutes.

$$\begin{tikzcd}
\mathrm{fun}({}^{F}/_{N}, {}^{F}/_{F'}) \arrow[r, "{}^{F}/_{N} \curvearrowright"] \arrow[d, "\lambda"'] & \mathrm{fun}({}^{F}/_{N}, {}^{F}/_{F'}) \arrow[d, "\lambda"] \\
K \arrow[r, "{}^{F}/_{N} \curvearrowright"] & K
\end{tikzcd}$$

Let $f \in \mathrm{fun}(F/N, F/F')$ be given by $f(x) = \sum\limits_{i=1}^{r} n_x^{(i)} a_i$, and take $g \in F/N$.

On the one hand, $f \mapsto \begin{pmatrix} 1 & \lambda(f) \\ 0 & 1 \end{pmatrix}$ and then F/N acts by conjugation:

$$\begin{pmatrix} g & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \lambda(f) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} g^{-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} g & g\lambda(f) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} g^{-1} & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & g\lambda(f) \\ 0 & 1 \end{pmatrix},$$

where

$$g\lambda(f) = g\sum_{i=1}^{r}\left(\sum_{x\in F/N} n_x^{(i)} x\right) t_i = \sum_{i=1}^{r}\left(\sum_{x\in F/N} n_x^{(i)} gx\right) t_i$$

On the other hand, F/N acts on fun$(F/N, F/F')$ by sending f to f^g. Now, since $f^g(x) = f(g^{-1}x)$, it can be represented by $\sum_{i=1}^{r} n_{g^{-1}x}^{(i)} a_i$. Thus,

$$\lambda(f^g) = \sum_{i=1}^{r}\left(\sum_{x\in F/N} n_{g^{-1}x}^{(i)} x\right) t_i.$$

It is easy to see that substituting $y = g^{-1}x$ makes $g\lambda(f) = \lambda(f^g)$.

□

Define a homomorphism $\varphi : F \to M(F/N)$ on generators by

$$\varphi(x_i) = \begin{pmatrix} \mu(x) & t_i \\ 0 & 1 \end{pmatrix}.$$

In the natural way, extend φ to a ring homomorphism, $\varphi^* : \mathbb{Z}F \to \mathbb{Z}(F/N)$. This homomorphism will be used in the sequel, and will in fact describe the Magnus embedding. But first, we describe some tools that will be needed for the Magnus embedding.

5.3 Fox Derivatives and Free Differential Calculus

In this section we develop free differential calculus, which will be later used to compute the Magnus embedding. Let G be a group and RG its group ring.

Definition 5.3.1. *A map $D : RG \to RG$ is a* derivation *if the following hold for any $u, v \in RG$ and for all $r \in R$:*

1. $D(u + v) = D(u) + D(v)$
2. $D(uv) = D(u)\epsilon(v) + uD(v)$

3. $D(ru) = rD(u)$

Remark. *D is completely determined by the values it takes on a generating set X of G. Indeed, for $u \in RG$*

$$D(u) = D(\sum r_i g_i) = \sum(D(r_i g_i)) = \sum r_i D(g_i),$$

so D is completely determined by its values on G. Now for $g, h \in G$,

$$D(gh) = D(g)\epsilon(h) + gD(h) = D(g) + gD(h).$$

Thus D is determined by the values it takes on $X \cup X^{-1}$. Finally,

$$D(1) = D(1 \cdot 1) = D(1)\epsilon(1) + 1D(1) = D(1) + D(1),$$

so $D(1) = 0$. Then for $x \in X$,

$$D(1) = D(xx^{-1}) = D(x)\epsilon(x^{-1}) + xD(x^{-1}) = D(x) + xD(x^{-1}),$$

hence $D(x^{-1}) = -x^{-1}D(x)$. Thus D is completely determined by the values it takes on X.

Theorem 5.3.2. *For any $x \in X$ there is a derivation, $\frac{\partial}{\partial x} : \mathbb{Z}F \to \mathbb{Z}F$ such that for any $y \in X$:*

$$\frac{\partial y}{\partial x} = \begin{cases} 1 & \text{if } y = x; \\ 0 & \text{otherwise.} \end{cases}$$

Proof. As mentioned above, a derivation is completely determined by the values it takes on generators, so one can define $\frac{\partial}{\partial x}$ as above and then extend it to a derivation on $\mathbb{Z}F$. The more elegant proof, presented here, involves using the map φ (with

$N = \{1\}$). Recall that $\varphi^* : \mathbb{Z}F \to M(F/N)$ is such that

$$\varphi^*(u) = \begin{pmatrix} u & \sum_{x\in X} u_x t_x \\ 0 & \epsilon(u) \end{pmatrix} = \begin{pmatrix} \alpha_{11}(u) & \alpha_{12}(u) \\ 0 & \alpha_{22}(u) \end{pmatrix}.$$

Now, since φ^* is a ring homomorphism, $\varphi^*(uv) = \varphi^*(u)\varphi^*(v)$. Further,

$$\begin{aligned} \varphi^*(uv) &= \begin{pmatrix} \alpha_{11}(uv) & \alpha_{12}(uv) \\ 0 & \alpha_{22}(uv) \end{pmatrix}, \text{ and} \\ \varphi^*(u)\varphi^*(v) &= \begin{pmatrix} \alpha_{11}(u) & \alpha_{12}(u) \\ 0 & \alpha 22(u) \end{pmatrix}\begin{pmatrix} \alpha_{11}(v) & \alpha_{12}(v) \\ 0 & \alpha_{22}(v) \end{pmatrix} \\ &= \begin{pmatrix} \alpha_{11}(u)\alpha_{11}(v) & \alpha_{11}(u)\alpha_{12}(v) + \alpha_{21}(u)\alpha_{22}(v) \\ 0 & \alpha_{22}(uv) \end{pmatrix}. \end{aligned}$$

It follows that

$$\begin{aligned} \alpha_{12}(uv) &= \alpha_{11}(u)\alpha_{12}(v) + \alpha_{12}(u)\alpha_{22}(v) \\ &= u\alpha_{12}(v) + \alpha_{12}(u)\varepsilon(v). \end{aligned}$$

Thus, the map $u \mapsto \alpha_{12}(u)$ is a derivation. (That this map is linear follows from the fact that it is defined via a ring homomorphism.)

Recall that T denotes the free $\mathbb{Z}F$-module (we are considering $N = 1$) with basis $\{t_x \mid x \in X\}$. Let $\rho_x : T \to \mathbb{Z}F$ be the projection on the x coordinate of T, i.e., for $u = \sum_{x\in X} u_x t_x$,

$$\rho(u) = u_x.$$

Define $D_x : \mathbb{Z}F \to \mathbb{Z}F$, as $D_x = \rho \circ \alpha_{12}$. Clearly, D_x is linear. Moreover, for $u, v \in \mathbb{Z}F$,

$$\begin{aligned} D_x(uv) &= \rho(\alpha_{12}(uv)) = \rho(u\alpha_{12}(v) + \alpha_{12}(u)\varepsilon(v)) \\ &= u\rho\alpha_{12}(v) + \varepsilon(v)\rho\alpha_{12}(u) \\ &= vD_x(v) + \varepsilon(u)D_x(v). \end{aligned}$$

Thus D_x is a derivation. Finally, for any $x, y \in X$,

$$D_x(y) = \rho(\alpha_{12}(y)) = \rho(t_y) = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise.} \end{cases}$$

Hence D_x satisfies the original criterion.

□

This derivation is called a *Fox partial derivative.* Free differential calculus is in spirit very similar to the calculus on the real line. This parallel will be pointed out throughout the discussion of Fox derivatives, as, although not necessary, it gives an idea of the general set-up and goals. It is not surprising then, that once partial derivatives have been defined, the equivalent of the gradient comes up naturally.

Proposition 5.3.3. *Let $u \in F$ and let $D : \mathbb{Z}F \to \mathbb{Z}F$ be a derivation. Then*

$$D(u) = \sum_{x \in X} \frac{\partial u}{\partial x} D(x).$$

This is called the formula of the complete differential*.*

Proof. Recall that $\Delta = \ker \varepsilon = \operatorname{id}(\{x - 1 \mid x \in X\})$, where $\varepsilon : \mathbb{Z}F \to \mathbb{Z}$ is the trivialisation map which maps an element to the sum of its coefficients. Hence, if $u \in F$,

$u - 1 \in \ker \varepsilon$ and therefore,

$$u - 1 = \sum_{x \in X} u_x (x - 1). \tag{5.2}$$

Applying the derivation $D : \mathbb{Z}F \to \mathbb{Z}F$ to the above,

$$\begin{aligned} D(u-1) &= \sum_{x \in X} \big(D(u_x)\varepsilon(x-1) + u_x D(x-1) \big) \\ D(u) - D(1) &= \sum_{x \in X} D(u_x) \cdot 0 + u_x \big(D(x) - D(1) \big). \end{aligned}$$

Therefore,

$$D(u) = \sum_{x \in X} u_x D(x). \tag{5.3}$$

Similarly, applying the derivation $\frac{\partial}{\partial y}$ with $y \in X$ to (5.2), gives

$$\frac{\partial u}{\partial y} = \sum_{x \in X} u_x \frac{\partial x}{\partial y} = u_y.$$

The last step follows from the fact that $\frac{\partial x}{\partial y} = 0$, whenever $x \neq y$. Now, one can replace the mysterious coefficients u_x in (5.3) to obtain the formula of the complete differential:

$$D(u) = \sum_{x \in X} \frac{\partial u}{\partial x} D(x).$$

□

Theorem 5.3.4 (The Fundamental Theorem of (Fox) Calculus)**.** *Let $u \in F$. Then,*

$$u - 1 = \sum_{x \in X} \frac{\partial u}{\partial x}(x - 1).$$

Proof. Notice that $u \mapsto u - 1$ with $u \in F$ is a derivation. Indeed, for $g, h \in F$,

$$gh \mapsto gh - 1 = gh - g + g - 1 = g(h-1) + g - 1 = g(h-1) + (g-1)\varepsilon(h)$$

and linearity is obvious. The formula of complete differential applied to this particular derivation yields

$$u - 1 = \sum_{x \in X} \frac{\partial u}{\partial x}(x - 1).$$

□

Since Fox derivatives will be used throughout, we present a few useful formulae analogous to the ones found in real calculus. Let $X = \{x_1, \ldots, x_n\}$.

1. 'Power' Rule: For $u \in F$ and $x \in X$,

   $$\frac{\partial u^{-1}}{\partial x} = -u^{-1}\frac{\partial u}{\partial x}.$$

   Proof. Since $1 = uu^{-1}$,

   $$0 = \frac{\partial}{\partial x}1 = \frac{\partial uu^{-1}}{\partial x} = \frac{\partial u}{\partial x} + u\frac{\partial u^{-1}}{\partial x}.$$

   Thus,

   $$\frac{\partial u^{-1}}{\partial x} = -u^{-1}\frac{\partial u}{\partial x}.$$

   □

2. Chain rule: Let $v = v(x_1, \ldots, x_n)$ be a word in generators. Denote by $v(u_1, \ldots, u_n)$ with $u_1, \ldots, u_n \in F$ the word obtained by replacing each x_i by u_i in v. By analogy with real caclulus, one thinks of the x_i as variables and of the u_i as values. Then

   $$\frac{\partial v(u_1, \ldots, u_n)}{\partial x} = \sum_{k=1}^{n} \frac{\partial v}{\partial u_k}\frac{\partial u_k}{\partial x}.$$

Proof. We proceed by induction on $m = |v|$. If $m = 1$, $v \in X^{\pm}$, so by the Power Rule and by definition of the derivation, the above formula holds.

Now suppose that $v = v_1 x_r^{\epsilon}$, where $x_r \in X$ and $\epsilon \in \{\pm 1\}$. Then

$$\begin{aligned}\frac{\partial v(u_1,\dots,u_k)}{\partial x} &= \frac{\partial v_1 u_r^{\epsilon}}{\partial x} = \frac{\partial v_1}{\partial x} + v_1 \frac{\partial u_r^{\epsilon}}{\partial x} \\ &= \sum_{i=1}^{k} \frac{\partial v_1(u_1,\dots,u_k)}{\partial u_i}\frac{\partial u_i}{\partial x} + v_1 \frac{\partial u_r^{\epsilon}}{\partial x}. \end{aligned} \tag{5.4}$$

Notice that

$$\frac{\partial v}{\partial u_i} = \frac{\partial v_1}{\partial u_i} + v_1 \frac{\partial u_r^{\epsilon}}{\partial u_i} = \begin{cases} \frac{\partial v_1}{\partial u_i} & \text{if } i \neq r; \\ \frac{\partial v_1}{\partial u_i} + v_1 \frac{\partial u_r^{\epsilon}}{\partial u_r} & \text{if } i = r. \end{cases}$$

With this, (5.4), becomes:

$$\begin{aligned}\frac{\partial v(u_1,\dots,u_k)}{\partial x} &= \sum_{i=1}^{k} \frac{\partial v_1(u_1,\cdots,u_k)}{\partial u_i}\frac{\partial u_i}{\partial x} + v_1 \frac{\partial u_r^{\epsilon}}{\partial x} \\ &= \sum_{i \neq r} \frac{\partial v}{\partial u_i}\frac{\partial u_i}{\partial x} + \frac{\partial v_1}{\partial u_r}\frac{\partial u_r}{\partial x} + v_1 \frac{\partial u_r^{\epsilon}}{\partial x} \\ &= \sum_{i \neq r} \frac{\partial v}{\partial u_i}\frac{\partial u_i}{\partial x} + \frac{\partial v_1}{\partial u_r}\frac{\partial u_r}{\partial x} + v_1 \frac{\partial u_r^{\epsilon}}{\partial u_r}\frac{\partial u_r}{\partial x} \\ &= \sum_{i \neq r} \frac{\partial v}{\partial u_i}\frac{\partial u_i}{\partial x} + \left(\frac{\partial v_1}{\partial u_r} + v_1 \frac{\partial u_r^{\epsilon}}{\partial u_r}\right)\frac{\partial u_r}{\partial x} \\ &= \sum_{i \neq r} \frac{\partial v}{\partial u_i}\frac{\partial u_i}{\partial x} + \frac{\partial v}{\partial u_r}\frac{\partial u_r}{\partial x} = \sum_{i=1}^{k} \frac{\partial v}{\partial u_i}\frac{\partial u_i}{\partial x} \end{aligned}$$

□

3. Special case of the product rule:

$$\frac{\partial x_{i_1}^{\epsilon_1}\cdots x_{i_k}^{\epsilon_k}}{\partial x} = \sum_{\substack{x_{i_m}=x \\ \epsilon_m=1}} x_{i_1}^{\epsilon_1}\cdots x_{i_{m-1}}^{\epsilon_{m-1}} - \sum_{\substack{x_{i_m}=x \\ \epsilon_m=-1}} x_{i_1}^{\epsilon_1}\cdots x_{i_m}^{\epsilon_m} = \sum_{x_{i_m}=x} \epsilon_m x_{i_1}^{\epsilon_1}\cdots x_{i_m}^{\frac{1}{2}(\epsilon_m-1)}. \tag{5.5}$$

5.4 The Magnus Embedding

Now that the basic free differential calculus has been developed, it is time to use it. Namely, it turns out to emerge naturally in connection with the Magnus embedding, in particular, in the module component.

Theorem 5.4.1 (Magnus theorem)**.** *The map $\varphi : F \to M(F/N)$ has kernel* $\ker \varphi = N'$ *and hence $\varphi : F/N' \hookrightarrow M$ is an embedding.*

The proof is done later in this section. With this theorem in mind one can define the long-awaited Magnus embedding.

Definition 5.4.2. *The map $\varphi : F/N' \to M(F/N)$ is called the* Magnus embedding.

Since φ is not surjective, it is crucial to have a good grasp of its image.

Theorem 5.4.3. *Let $\varphi : F \to M(F/N)$ be as before. Then, for $u \in F$,*

$$\varphi(u) = \begin{pmatrix} \mu(u) & \sum\limits_{x\in X} \mu\left(\frac{\partial u}{\partial x}\right) t_x \\ 0 & 1 \end{pmatrix}.$$

Proof. Let $u \in F$ and proceed by induction on the length, $|u|$, of u. If $|u| = 1$, then $u \in X$ is a generator and the result follows trivially by the definition of φ. Now suppose that $|u| > 1$, i.e., suppose $u = vy$, where v is reduced and $y \in X$. The case where $u = wy^{-1}$ is similar, so it will not be done in detail. Then,

$$\begin{aligned} \varphi(u) &= \varphi(v)\varphi(y) = \begin{pmatrix} \mu(v) & \sum\limits_{x\in X} \mu\left(\frac{\partial v}{\partial x}\right) t_x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu(y) & t_y \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \mu(v)\mu(y) & \mu(v)t_y + \sum\limits_{x\in X} \mu\left(\frac{\partial v}{\partial x}\right) t_x \\ 0 & 1 \end{pmatrix} \end{aligned}$$

Recall that, by the product rule, for $y \in X$:

1. If $y \neq x$,
$$\frac{\partial vy}{\partial x} = \frac{\partial v}{\partial x}\varepsilon(y) + v\frac{\partial y}{\partial x} = \frac{\partial v}{\partial x}.$$

2. If $y = x$,
$$\frac{\partial vy}{\partial y} = \frac{\partial v}{\partial y}\varepsilon(y) + v\frac{\partial y}{\partial y} = \frac{\partial v}{\partial y} + v.$$

Now observe that,

$$\begin{aligned}
\mu(v)t_y + \sum_{x\in X}\mu\left(\frac{\partial v}{\partial x}\right)t_x &= \mu(v)t_y + \mu\left(\frac{\partial v}{\partial y}\right)t_y + \sum_{x\neq y}\mu\left(\frac{\partial v}{\partial x}\right)t_x \\
&= \left(\mu(v) + \mu\left(\frac{\partial v}{\partial y}\right)\right)t_y + \sum_{x\neq y}\mu\left(\frac{\partial v}{\partial x}\right)t_x \\
&= \mu\left(\frac{\partial vy}{\partial y}\right) + \sum_{x\neq y}\mu\left(\frac{\partial vy}{\partial x}\right)t_x = \sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)t_x.
\end{aligned}$$

With this,

$$\varphi(u) = \begin{pmatrix} \mu(u) & \sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)t_x \\ 0 & 1 \end{pmatrix}.$$

□

5.4.1 The Magnus-Fox equivalence

This subsection is devoted to proving the Magnus Theorem. It is rather complicated to prove this theorem directly. The strategy will be to show that it is equivalent to Fox's theorem, which is easier to prove.

Theorem 5.4.4 (Fox theorem)**.** *Let N be a normal subgroup of F. Denote by μ the canonical epimorphism thereon, which is extended to the epimorphism $\mu : \mathbb{Z}F \to$*

$\mathbb{Z}(F/N)$ in the natural way. For all $u \in F$,

$$u \in N' \Leftrightarrow \mu\left(\frac{\partial u}{\partial x}\right) = 0 \textit{ for all } x \in X.$$

Proof. $\Rightarrow$. Suppose $u \in N' = [N, N]$. Then, keeping in mind that $[x, y]^{-1} = [y, x]$, one can write $u = u_1 \cdots u_k$, with $u_i = [r_i, s_i]$ for some $r_i, s_i \in N$. Compute for any $r, s \in N$ the formula

$$\begin{aligned}
\frac{\partial [r,s]}{\partial x} &= \frac{\partial rsr^{-1}s^{-1}}{\partial x} = \frac{\partial r}{\partial x} + r\frac{\partial sr^{-1}s^{-1}}{\partial x} \\
&= \frac{\partial r}{\partial x} + r\left(\frac{\partial s}{\partial x} + s\frac{\partial r^{-1}s^{-1}}{\partial x}\right) \\
&= \frac{\partial r}{\partial x} + r\frac{\partial s}{\partial x} + rs\left(\frac{\partial r^{-1}}{\partial x} + r^{-1}\frac{\partial s^{-1}}{\partial x}\right) \\
&= \frac{\partial r}{\partial x} + r\frac{\partial s}{\partial x} + rs\left(-r^{-1}\frac{\partial r}{\partial x}\right) + rsr^{-1}\left(-s^{-1}\frac{\partial s}{\partial x}\right) \\
&= \frac{\partial r}{\partial x} - rsr^{-1}\frac{\partial r}{\partial x} + r\frac{\partial s}{\partial x} - rsr^{-1}s^{-1}\frac{\partial s}{\partial x}.
\end{aligned}$$

Since $r, s \in N$, $\mu(r) = \mu(s) = 1$. Hence,

$$\mu\left(\frac{\partial [r,s]}{\partial x}\right) = \mu\left(\frac{\partial r}{\partial x}\right) - \mu\left(\frac{\partial r}{\partial x}\right) + \mu\left(\frac{\partial s}{\partial x}\right) - \mu\left(\frac{\partial s}{\partial x}\right) = 0.$$

Now, if $k = 1$, by the above, $\mu\left(\frac{\partial u}{\partial x}\right) = 0$. Suppose by induction, that $\mu\left(\frac{\partial u_1 \cdots u_{k-1}}{\partial x}\right) = 0$. Then,

$$\mu\left(\frac{\partial u}{\partial x}\right) = \mu\left(\frac{\partial u_1 \cdots u_{k-1} u_k}{\partial x}\right) = \mu\left(\frac{\partial u_1 \cdots u_{k-1}}{\partial x}\right) + u_1 \cdots u_{k-1}\mu\left(\frac{\partial u_k}{\partial x}\right) = 0.$$

$\Leftarrow$. Suppose that $u \in F$ is such that $\mu\left(\frac{\partial u}{\partial x}\right) = 0$ for all $x \in X$. One shows that $u \in N'$ by induction on the length $|u|$ of u. If $|u| = 0$, then $u = 1 \in N'$. Assume that the result holds if $|u| \leq k - 1$.

Suppose that $u = x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k}$ with $\epsilon_m = \pm 1$. Then, by the special case of the product rule,

$$\frac{\partial x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k}}{\partial x} = \sum_{x_{i_m} = x} \epsilon_m x_{i_1}^{\epsilon_1} \cdots x_{i_m}^{\frac{1}{2}(\epsilon_m - 1)}.$$

Write $c_{i_m} = \epsilon_m x_{i_1}^{\epsilon_1} \cdots x_{i_m}^{\frac{1}{2}(\epsilon_m - 1)}$. Now,

$$\mu\left(\frac{\partial u}{\partial x}\right) = \sum_{x_{i_m} = x} \mu(c_{i_m}) = 0.$$

But $\epsilon_m = \pm 1$, so it must be the case that the c_{i_m} cancel out in pairs, i.e., for every c_{i_j}, there is a c_{i_l} such that

(a) $\mu(x_{i_1}^{\epsilon_1} \cdots x_{i_j}^{\frac{1}{2}(\epsilon_j - 1)}) = \mu(x_{i_1}^{\epsilon_1} \cdots x_{i_l}^{\frac{1}{2}(\epsilon_l - 1)})$ in F/N

(b) $\epsilon_j = -\epsilon_l$.

Thus for each $x \in X$ the set of indices $\{1, \ldots, k\}$ can be partitioned naturally in a set of pairs $\{(p_1, q_1), \cdots, (p_s, q_s)\}$, according to the above criterion. Order this set so that $p_i < q_i$ and that $q_1 < \ldots < q_s$. For each $x \in X$, one gets such a sequence of indices $q_1(x) < \cdots < q_s(x)$. Choose $x \in X$ so that $q_1(x) \leq q_1(y)$ for all $y \neq x$.

Lemma. *Let p, q be such that $\mu(c_p) = \mu(c_q)$. Write*

$$u = A x_{i_p}^{\epsilon_p} A_1 x_{i_q}^{\epsilon_q} A_2 = A x^{\epsilon} A_1 x^{-\epsilon} A_2.$$

Then $A_1 \in N$.

Proof. Recall that $u = x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k}$ and $c_{i_m} = \epsilon_m x_{i_1}^{\epsilon_1} \cdots x_{i_m}^{\frac{1}{2}(\epsilon_m - 1)}$. So in the above situation,

(a) If $\epsilon = 1$, $c_p = A_x$ and $c_q = -AxA_1x^{-1}$. Since $\mu(c_p + c_q) = 0$, then

$$\mu(A - AxA_1x^{-1}) = \mu\big(A(1 - xA_1x^{-1})\big) = \mu(A)\mu\big(1 - xA_1x^{-1}\big) = 0$$

in the group ring. Note that no element of F/N can be a zero divisor, since otherwise, there would exist some $h \in F/N$ such that $g(kh) = kgh = 0$ for some $k \in \mathbb{Z}$. But this is clearly impossible. A is a group element, so $\mu(A) \neq 0$ and hence $\mu\big(1 - xA_1x^{-1}\big) = 0$, i.e., $A_1^{x^{-1}} = 1$ in F/N. So $A_1^{x^{-1}} \in N$, but N is normal, so $A_1 \in N$.

(b) If $\epsilon = -1$, $c_p = -Ax^{-1}$ and $c_q = Ax^{-1}A_1$. With this,

$$\mu(-Ax^{-1} + Ax^{-1}A_1) = \mu\big(- Ax^{-1}(1 - A_1)\big) = \mu(-Ax^{-1})\mu(1 - A_1) = 0.$$

Since $\mu(-Ax^{-1})$ is a group element, it is not a zero divisor in $\mathbb{Z}(F/N)$, so $\mu(1 - A_1) = 0$, and hence $\mu(A_1) = 1$ in F/N which means that $A_1 \in N$, as required.

□

Let $q_1 = l$. As before $q_1 < \cdots < q_s$. Recall that $x \in X$ was chosen to be such that q_1 is minimal among $q_1(y)$ for all $y \in X$. Then one can write

$$u = Ax_l^{\epsilon}A_1x_l^{-\epsilon}A_2,$$

with $A_1 \neq 1$, since u is reduced. Let $A_1 = Bx_m^{\nu}$. Since $\mu\left(\frac{\partial u}{\partial x}\right) = 0$ for any $x \in X$, in particular, $\mu\left(\frac{\partial u}{\partial x_m}\right) = 0$, so $x_m^{-\nu}$ must appear somewhere in u. But q_1 was chosen minimal, so $x_m^{-\nu}$ must occur after $x_l^{-\epsilon}$, i.e. $x_m^{-\nu}$ occurs in A_2. One can now write

$$u = Ax_l^{\epsilon} B x_m^{\nu} x_l^{-\epsilon} C x_m^{-\nu} D.$$

By the lemma,

(a) $n_1 = Bx_m^{\nu} \in N$

(b) $n_2 = x_l^{-\epsilon} C \in N$.

This yields:

$$\begin{aligned} u &= Ax_l^{\epsilon} n_1 n_2 x_m^{-\nu} D = Ax_l^{\epsilon}(n_1 n_2 n_1^{-1} n_2^{-1}) n_2 n_1 x_m^{-\nu} D \\ &\overset{N'}{\equiv} Ax_l^{\epsilon} n_2 n_1 x_m^{-\nu} D = Ax_l^{\epsilon} x_l^{-\epsilon} C B x_m^{\nu} x_m^{-\nu} D \\ &= ACBD \end{aligned}$$

Thus $u = ACBD \mod N'$. Denote $ACBD$ by v. Then $u = v(modN')$, so there exists $u' \in N'$ such that $u = vu'$ and $|v|, |u'| < |u| = k$. Now,

$$\frac{\partial u}{\partial x} = \frac{\partial vu'}{\partial x} = \frac{\partial v}{\partial x} + v\frac{\partial u'}{\partial x}.$$

But, since $u' \in N'$, by the first part of this theorem, $\mu\left(\frac{\partial u'}{\partial x}\right) = 0$. Hence,

$$0 = \mu\left(\frac{\partial u}{\partial x}\right) = \mu\left(\frac{\partial v}{\partial x}\right) + \mu\left(v\frac{\partial u'}{\partial x}\right) = \mu\left(\frac{\partial v}{\partial x} + 0\right) = \mu\left(\frac{\partial v}{\partial x}\right).$$

Thus $\mu\left(\frac{\partial v}{\partial x}\right) = 0$ with $|v| < k$, so by induction hypothesis, $v \in N'$. But then $u = vu' \in N'$.

□

Finally, we show that Magnus' and Fox's theorems are equivalent.

Theorem 5.4.5. *The following are equivalent:*

1) $\ker\varphi = N'$

2) *For every element* $u \in F$, $u \in N'$ *if and only if* $\mu\left(\frac{\partial u}{\partial x}\right) = 0$ *for all* $x \in X$.

Proof. On the one hand, assume that $\ker\varphi = N'$ Recall that for $u \in F$,

$$\varphi(u) = \begin{pmatrix} \mu(u) & \sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)t_x \\ 0 & 1 \end{pmatrix}.$$

Assuming 1), $u \in N'$ if and only if $\varphi(u) = 1$. This happens if and only if

$$\sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)t_x = 0 \text{ and } \mu(u) = 1,$$

i.e., if $\mu\left(\frac{\partial u}{\partial x}\right) = 0$ for all $x \in X$, so we have 2)

On the other hand, assume 2). Take $u \in N'$. Then

$$\mu(u) - 1 = \sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)\mu(x-1) = 0.$$

Hence $\mu(u) = 1$ and so $u \in \ker\varphi$.

Now take $u \in \ker\varphi$. Since $\varphi(u) = 1$,

$$\sum_{x\in X}\mu\left(\frac{\partial u}{\partial x}\right)t_x = 0, \text{ so } \mu\left(\frac{\partial u}{\partial x}\right) = 0 \text{ for all } x \in X.$$

From 2), it follows that $u \in N'$.

□

5.4.2 The Image of the Magnus embedding

The Magnus embedding $\varphi : F/N' \to M(F/N)$ is far from being surjective. However, if we are to work with it, we need to know what its image is. It was described by Remeslennikov and Sokolov in [17]. Since this is particularly important for the main result, the proof of their result is given here for completeness. First, we need a lemma.

Lemma 5.4.6 (Lyndon). *Let $u_1, \dots, u_r \in F$. Then,*

$$\mu\left(\sum_{i=1}^{r} u_i(x_i - 1)\right) = 0$$

if and only if there exists $p \in N$ such that for all $1 \le i \le r$,

$$\mu\left(\frac{\partial p}{\partial x_i}\right) = \mu(u_i).$$

Proof. The proof hinges on the Fundamental Theorem of Calculus.

$\Leftarrow$. Since $p \in N$, $\mu(p) = 1$. Hence,

$$\begin{aligned}\mu\left(\sum_{i=1}^{r} u_i(x_i - 1)\right) &= \sum_{i=1}^{r} \mu(u_i)\mu(x_i - 1) = \sum_{i=1}^{r} \mu\left(\frac{\partial p}{\partial x_i}\right)\mu(x_i - 1) = \\ &= \mu\left(\sum_{i=1}^{r}\left(\frac{\partial p}{\partial x_i}\right)(x_i - 1)\right) = \mu(p - 1) = 0.\end{aligned}$$

$\Rightarrow$. Recall that $\mu : \mathbb{Z}F \to \mathbb{Z}(F/N)$ is a ring homomorphism.

Lemma. *The kernel of μ is $\mathbb{Z}F(N-1)$.*

Proof. Clearly, for any $f \in \mathbb{Z}F, n \in N$, $\mu(f(n-1)) = \mu(f)(\mu(n) - 1) = 0$. Hence, $\mathbb{Z}F(N-1) \subseteq \ker \mu$. Conversely, suppose that $\mu(u) = 0$ for some $u \in \mathbb{Z}F$. One can write $u = \sum_i \epsilon_i v_i$ with $\epsilon_i \in \{\pm 1\}$ as a linear combination of words $v_i \in F$ by viewing an element of the form kv_i with $k \in \mathbb{Z}$ as the sum $v_1 + \dots + v_r$ when $k > 0$ and $-v_1 - \dots - v_r$, when $k < 0$. Then

$$\mu(u) = \mu(\sum_i \epsilon_i v_i) = \sum_i \epsilon_i \mu(v_i) = 0,$$

so for each v_i, there is a $v_{i'}$ such that $\epsilon_i \mu(v_i) = -\epsilon_{i'} \mu(v_{i'})$, i.e., $v_{i'} = v_i n$ for some $n \in N$. With this,

$$2u = 2\sum_i \epsilon_i v_i = \sum_i (\epsilon_i v_i + \epsilon_{i'} v_{i'}) = \sum_i (\epsilon_i v_i + \epsilon_{i'} v_i n) = \sum -\epsilon_i (n-1) v_i \in \mathbb{Z}F(N-1).$$

Thus $\ker \mu = \mathbb{Z}F(N-1)$, proving the claim. □

Now, let $u = \sum_{i=1}^r u_i(x_i - 1)$ and suppose $\mu(u) = 0$, i.e., $u \in \ker_{\mathbb{Z}F}(\mu)$. Hence $u \in \mathbb{Z}F(N-1)$ and one can write

$$u = \sum_{i=1}^r v_i(n_i - 1),$$

where $n_i \in N$ and $v_i \in \mathbb{Z}F$. Then

$$\frac{\partial u}{\partial x_j} = \sum_{i=1}^r \frac{\partial v_i(n_i-1)}{\partial x_j} = \sum_{i=1}^r \left(\frac{\partial v_i}{\partial x_j} \varepsilon(n_i - 1) + v_i \frac{\partial (n_i - 1)}{\partial x_j} \right) = \sum_{i=1}^r v_i \frac{\partial n_i}{\partial x_j}.$$

On the other hand, by the Fundamental Theorem of Calculus,

$$u = (u+1) - 1 = \sum_{i=1}^r \frac{\partial(u+1)}{\partial x_i}(x_i - 1),$$

from which it follows that $u_j = \frac{\partial u}{\partial x_j}$. Hence

$$u_j = \sum_{i=1}^r v_i \frac{\partial n_i}{\partial x_j}.$$

Now, take $n = \prod_{i=1}^{r} v_i n_i v_i^{-1}$. Then $\mu\left(\frac{\partial n}{\partial x_j}\right) = \mu(u_j)$. Indeed, observe that for any $g \in N$,

$$\begin{aligned}
\frac{\partial vgv^{-1}}{\partial x} &= \frac{\partial v}{\partial x} + v\frac{\partial g}{\partial x} + vg\frac{\partial v^{-1}}{\partial x}\\
&= \frac{\partial v}{\partial x} + v\frac{\partial g}{\partial x} - vgv^{-1}\frac{\partial v}{\partial x}\\
\Rightarrow \mu\left(\frac{\partial vgv^{-1}}{\partial x}\right) &= \mu(v)\mu\left(\frac{\partial g}{\partial x}\right).
\end{aligned}$$

With this, keeping in mind that $v_i n_i v_i^{-1} \in N$,

$$\begin{aligned}
\mu\left(\frac{\partial n}{\partial x_j}\right) &= \frac{\partial}{\partial x_j}\left(\prod_{i=1}^{r} v_i n_i v_i^{-1}\right) = \mu\left(\frac{\partial v_1 n_1 v_1^{-1}}{\partial x_j}\right) + \mu\left(v_1 n_1 v_1^{-1}\right)\mu\left(\frac{\partial}{\partial x_j}\left(\prod_{i=2}^{r} v_i n_i v_1^{-1}\right)\right)\\
&= \mu\left(\frac{\partial v_1 n_1 v_1^{-1}}{\partial x_j}\right) + \mu\left(\frac{\partial}{\partial x_j}\left(\prod_{i=2}^{r} v_i n_i v_1^{-1}\right)\right)\\
&= \cdots = \mu\left(\frac{\partial v_1 n_1 v_1^{-1}}{\partial x_j}\right) + \cdots + \mu\left(\frac{\partial v_r n_r v_r^{-1}}{\partial x_j}\right)\\
&= \mu\left(v_1\frac{\partial n_1}{\partial x_j}\right) + \cdots + \mu\left(v_r\frac{\partial n_r}{\partial x_j}\right)\\
&= \mu\left(\sum_{i=1}^{r} v_i\frac{\partial n_i}{\partial x_j}\right) = \mu(u_j).
\end{aligned}$$

□

With this, we are ready to attack the main theorem of this subsection.

Theorem 5.4.7 (Remeslennikov-Sokolov)**.** *Let*

$$M = \begin{pmatrix} u & \sum_{i=1}^{r} u_i t_i \\ 0 & 1 \end{pmatrix} \in M(F/N).$$

Then, $M \in \operatorname{Im}(\varphi)$ if and only if $u - 1 = \sum_{i=1}^{r} u_i\big(\mu(x_i) - 1\big)$ in $\mathbb{Z}(F/N)$.

Proof. ⇒. Let $u \in F$. Then,

$$M = \varphi(u) = \begin{pmatrix} \mu(u) & \sum\limits_{i=1}^{r} \mu\left(\frac{\partial u}{\partial x_i}\right) t_i \\ 0 & 1 \end{pmatrix}.$$

Hence, $u_i = \mu\left(\frac{\partial u}{\partial x_i}\right)$. By the Fundamental Theorem of Calculus,

$$u - 1 = \sum_{i=1}^{r} \frac{\partial u}{\partial x_i}(x_i - 1) \text{ in } \mathbb{Z}F,$$

thus, in $\mathbb{Z}(F/N)$,

$$\mu(u-1) = \sum_{i=1}^{r} \mu\left(\frac{\partial u}{\partial x_i}\right)(\mu(x_i) - 1) = \sum_{i=1}^{r} \mu(u_i)(\mu(x_i) - 1).$$

⇐. Let $M = \begin{pmatrix} u & \sum_{i=1}^{r} u_i t_i \\ 0 & 1 \end{pmatrix}$ with $u \in F/N$ and $u_i \in \mathbb{Z}\left(F/N\right)$ be such that

$$\mu(u-1) = \mu\left(\sum_{i=1}^{r} u_i(x_i - 1)\right).$$

The goal is to show that $M \in \text{Im}(\varphi)$. Since $u \in F/N$ and μ is surjective, there is some $v \in F$ such that $u = \mu(v)$. The idea is to show that $\varphi(v^{-1})M \in \text{Im}(\varphi)$ and hence, since $\text{Im}(\varphi)$ is a subgroup, $M \in \text{Im}(\varphi)$. Now,

$$\begin{aligned} \varphi(v^{-1}) &= \begin{pmatrix} \mu(v^{-1}) & \sum\limits_{i=1}^{r} \mu\left(\frac{\partial v^{-1}}{\partial x_i}\right) t_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mu(v^{-1}) & \sum\limits_{i=1}^{r} \mu\left(-v^{-1}\frac{\partial v}{\partial x_i}\right) t_i \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \mu(v^{-1}) & -\mu(v^{-1})\sum\limits_{i=1}^{r} \mu\left(\frac{\partial v}{\partial x_i}\right) t_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} u^{-1} & -u^{-1}\sum\limits_{i=1}^{r} \mu\left(\frac{\partial v}{\partial x_i}\right) t_i \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

From this, it follows that

$$\begin{aligned}
\varphi(v^{-1})M &= \begin{pmatrix} u^{-1} & -u^{-1}\sum\limits_{i=1}^{r}\mu\left(\frac{\partial v}{\partial x_i}\right)t_i \\ 0 & 1 \end{pmatrix}\begin{pmatrix} u & \sum\limits_{i=1}^{r}u_i t_i \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & u^{-1}\sum\limits_{i=1}^{r}u_i t_i - u^{-1}\sum\limits_{i=1}^{r}\mu\left(\frac{\partial v}{\partial x_i}\right)t_i \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & u^{-1}\sum\limits_{i=1}^{r}\left(u_i - \mu\left(\frac{\partial v}{\partial x_i}\right)\right)t_i \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

Observe that,

$$\begin{aligned}
u^{-1}\mu\left(\sum_{i=1}^{r}\left(u_i - \frac{\partial v}{\partial x_i}\right)\left(x_i - 1\right)\right) &= u^{-1}\left(\sum_{i=1}^{r}u_i\big(\mu(x_i)-1\big) - \sum_{i=1}^{r}\left(\frac{\partial v}{\partial x_i}\right)\big(\mu(x_i)-1\big)\right) \\
&= u^{-1}\left(u-1-(u-1)\right) = 0.
\end{aligned}$$

Now by Lyndon's lemma, there exists an $n \in N$ such that for all $1 \leq j \leq r$,

$$\mu\left(\frac{\partial n}{\partial x_j}\right) = u^{-1}\left(u_j - \mu\left(\frac{\partial v}{\partial x_j}\right)\right).$$

Thus,

$$\begin{aligned}
\varphi(n) &= \begin{pmatrix} \mu(n) & \sum\limits_{i=1}^{r}\mu\left(\frac{\partial n}{\partial x_i}\right)t_i \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & u^{-1}\sum\limits_{i=1}^{r}\left(u_j - \mu\left(\frac{\partial v}{\partial x_j}\right)\right)t_i \\ 0 & 1 \end{pmatrix} = \varphi(v^{-1})M.
\end{aligned}$$

Thus, $\varphi(v^{-1})M \in \mathrm{Im}(\varphi)$ and hence $M \in Im(\varphi)$.

□

5.5 The Word Problem in Free Solvable Groups

It is finally time to turn our attention to the Word Problem. It was shown by Miasnikov, Roman'kov, Ushakov and Vershik in [16] that its time complexity is cubic. Besides giving an algorithm to solve the Word Problem the authors do so by using a fascinating geometric approach.

5.5.1 Geometric interpretation of Fox derivatives

The main idea is to consider flows on the Cayley graph $\Gamma(F/N, X)$.

Definition 5.5.1. *Let* $\Gamma = \big(V, E\big)$ *be a directed edge-labelled graph with labels from* X. *A* network *on* Γ *is a function*

$$f : E \to \mathbb{Z}.$$

Given a network as above, define the *flow* through a vertex v to be the function $f : V \to \mathbb{Z}$ given by

$$f(v) = \sum_{\sigma(e)=v} f(e) - \sum_{\tau(e)=v} f(e).$$

Definition 5.5.2. *A* flow network *is a network,* $\big(\Gamma, f\big)$, *satisfying* Kirchhoff's law*:*

$$f(v) = 0, \tag{5.6}$$

for all $v \in V$, *except for two designated vertices, the* source s *and the* sink t.

In other words, in a flow network the net flow out of each vertex, except for the source and sink, is zero.

Definition 5.5.3. *A flow* f *is a* circulation *if* $f(v) = 0$ *for all* $v \in V$, *including the source and the sink.*

Let $C\big(\Gamma\big)$ be the set of all circulations on the graph Γ . It forms an abelian group with respect to addition, which is defined as:

$$(f+g)(e) = f(e) + g(e),$$

for every edge $e \in E$ where f and g are circulations on Γ.

One would like to view the Cayley graph of F/N, $\Gamma\big(F/N, X\big)$, as a flow network. However, the fact that Γ does not have specific edges corresponding to the inverses of generators presents a slight technical problem. We consider a modification $\tilde{\Gamma}$ of Γ constructed in the following way: for every edge $e \in E(\Gamma)$ labelled by $x \in X$, introduce an edge, e^{-1} such that

1. $\sigma(e^{-1}) = \tau(e)$,
2. $\tau(e^{-1}) = \sigma(e)$,
3. e^{-1} has label x^{-1}.

For a *flow*, $\tilde{f}$ on $\tilde{\Gamma}$ satisfying

$$\tilde{f}(e^{-1}) = -\tilde{f}(e),$$

Kirchhoff's law takes the form

$$f(v) = \sum_{\sigma(e)=v} f(e) = 0, \text{ for all } v \in V \smallsetminus \{s,t\}.$$

While the idea behind the geometric interpretation is very intuitive, describing it formally can be convoluted. This may lead to the occasional abuse of notation.

Recall that a word $w = x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k} \in F(X)$ corresponds to a path, $p_w = e_1 \cdots e_k$, in the Cayley graph, Γ. Here, $\sigma(e_1) = 1$ and the label of e_j is $x_{i_j}^{\epsilon_j}$. Now, for every path

p in Γ there is a corresponding natural flow, given by the algebraic number of times the path traverses an edge. More precisely, a path $p = e_1^{\epsilon_1} \cdots e_k^{\epsilon_k}$ in Γ defines a flow $\pi_p : E(\Gamma) \to \mathbb{Z}$ in the following way:

$$\pi_p(e) = \sum_{i=1}^{k} \delta(e_i^{\epsilon_i}, e) - \sum_{i=1}^{k} \delta(e_i^{\epsilon_i}, e^{-1}),$$

where $\delta(e_i^{\epsilon_i}, e) = \begin{cases} 1 & \text{if } e_i^{\epsilon_i} = e \\ 0 & \text{otherwise.} \end{cases}$

For any path p in the Cayley graph, π_p is indeed a flow, since every time p goes into a vertex, by some edge it comes out of it through another one, except possibly at the endpoints. Moreover, since a word is has finite length, p is a finite path and hence $|\mathrm{supp}(\pi_p)| < \infty$.

Now that graph theory is slowly creeping in, in the shape of flows on a Cayley graph, it is time to establish a link between flows and Fox derivatives.

Lemma 5.5.4. *Let N be a normal subgroup of F and let $w \in F$. Denote by $\Gamma(F/N, X)$ the Cayley graph of F with respect to the set of generators X. Then, for $x \in X$,*

$$\mu\left(\frac{\partial w}{\partial x}\right) = \sum_{g \in F/N} \big(\pi_w(g, gx)\big) g,$$

where (g, gx) is an edge in the Cayley graph of F/N.

Proof. The statement is obviously true if $|w| = 0$. We proceed by induction on the length of w. Let $w = w_1 y^{\epsilon}$ with $y \in X$, $\epsilon = \pm 1$ and suppose that the theorem holds for w_1. Recall that $\pi_w(g, gx)$ is the algebraic number of times the edge (g, gx) appears in the path w. Since w_1 is a subword of w, whenever an edge (g, gx) appears in w_1, it also appears in w. So the value of $\pi_w(g, gx)$ can differ from that of $\pi_{w_1}(g, gx)$, only for the edge $(w_1, w_1 y^{\epsilon})$. Thus $\pi_w(g, gx) = \pi_{w_1}(g, gx)$ for all $g \in F/N \smallsetminus w_1$. Hence,

$$\begin{aligned}\sum_{g \in F/N} \big(\pi_w(g, gx)\big)g &= \sum_{g \in F/N \smallsetminus w_1} \big(\pi_w(g, gx)\big)g + \pi_w(w_1, w_1 x)w_1 \\ &= \sum_{g \in F/N \smallsetminus w_1} \big(\pi_{w_1}(g, gx)\big)g + \pi_w(w_1, w_1 x)w_1\end{aligned}$$

1. If $y \neq x$, then by the product rule,

$$\frac{\partial w}{\partial x} = \frac{\partial w_1}{\partial x}.$$

   Moreover, since $y \neq x$, the edge $(w_1, w_1 x)$ does not appear in w and so $\pi_w(w_1, w_1 x) = 0$. The result follows from the induction hypothesis.

2. If $y = x$, then by the product rule,

$$\frac{\partial w}{\partial x} = \frac{\partial w_1}{\partial x} + \epsilon w_1 y^{\frac{1}{2}(\epsilon - 1)}.$$

$\epsilon = \ 1.$ In this case the edge $(w_1, w_1 x)$ is indeed on the path w, so it contributes +1 to the flow along this edge. Moreover, since the edge $(w_1, w_1 x)$ is not on the path w_1, $\pi_{w_1}(w_1, w_1 x) = 0$, so $\pi_w(w_1, w_1 x) = 1$. Thus,

$$\begin{aligned}\sum_{g \in F/N} \big(\pi_w(g, gx)\big)g &= \sum_{g \in F/N \smallsetminus w_1} \big(\pi_w(g, gx)\big)g + \pi_w(w_1, w_1 x)w_1 \\ &= \sum_{g \in F/N \smallsetminus w_1} \big(\pi_w(g, gx)\big)g + w_1 \\ &= \mu\left(\frac{\partial w_1}{\partial x}\right) + w_1 = \mu\left(\frac{\partial w}{\partial x}\right).\end{aligned}$$

$\epsilon = -1.$ Since $w = w_1 y^{-1}$ and in the sum we only consider edges (g, gx) with $x \in X$, the last edge to consider is (w, wy). Furthermore, since the last edge is oriented backwards, the flow along it is −1. Now,

$$\begin{aligned}\sum_{g\in F/N}\big(\pi_w(g,gx)\big)g &= \sum_{g\in F/N\smallsetminus w}\big(\pi_w(g,gx)\big)g+\pi_w(w,wy)w\\ &= \sum_{g\in F/N\smallsetminus w_1}\big(\pi_w(g,gx)\big)g-w\\ &= \sum_{g\in F/N\smallsetminus w_1}\big(\pi_w(g,gx)\big)g-w_1y^{-1}\\ &= \mu\left(\frac{\partial w_1}{\partial x}\right)-w_1y^{-1}=\mu\left(\frac{\partial w}{\partial x}\right).\end{aligned}$$

□

Corollary 5.5.5. *Let $u,v\in F$. Then $\pi_u=\pi_v$ if and only if*

$$\frac{\partial u}{\partial x}=\frac{\partial v}{\partial x}\ \textit{for all } x\in X.$$

Proof. By Lemma 5.5.4, $\mu\left(\frac{\partial w}{\partial x}\right)=\sum\limits_{g\in F/F^{(d)}}\pi_w(g,gx)g$. Hence, if for all $x\in X$ $\frac{\partial u}{\partial x}=\frac{\partial v}{\partial x}$, then

$$\sum_{g\in F/F^{(d)}}\pi_u(g,gx)g=\sum_{x\in X}\pi_v(g,gx)g.$$

It follows that for all $x\in X$ and for all $g\in F/F^{(d)}$ $\pi_u(g,gx)=\pi_v(g,gx)$, i.e., $\pi_u=\pi_v$. Conversely, if $\pi_u=\pi_v$, then for each $x\in X$, $\frac{\partial u}{\partial x}=\frac{\partial v}{\partial x}$. Thus, if one is to check whether $\pi_u=\pi_v$, it is enough to check that $\frac{\partial u}{\partial x}=\frac{\partial v}{\partial x}$. □

Theorem 5.5.6. *Let $u,v\in F(X)$. Then $u=v$ in F/N' if and only if*

$$\pi_u=\pi_v,$$

where π_u, π_v are flows on $\Gamma(F/N)$.

Proof. $\Rightarrow$. Suppose that $u = v$ in F/N', i.e., $uv^{-1} \in N'$. Then by Fox theorem,

$$\mu\left(\frac{\partial uv^{-1}}{\partial x}\right) = 0.$$

Now, by the product rule,

$$\frac{\partial uv^{-1}}{\partial x} = \frac{\partial u}{\partial x} + u\frac{\partial v^{-1}}{\partial x} = \frac{\partial u}{\partial x} - uv^{-1}\frac{\partial v}{\partial x}.$$

Hence,

$$\mu\left(\frac{\partial uv^{-1}}{\partial x}\right) = \mu\left(\frac{\partial u}{\partial x}\right) - \mu\left(\frac{\partial v}{\partial x}\right) = 0.$$

This yields $\mu\left(\frac{\partial u}{\partial x}\right) = \mu\left(\frac{\partial v}{\partial x}\right)$ and so by Lemma 5.5.4, for all $x \in X$,

$$\sum_{g \in F/N} \big(\pi_u(g, gx)\big)g = \sum_{g \in F/N} \big(\pi_v(g, gx)\big)g.$$

Since $g \in F/N$ are linearly independent, this gives $\pi_u(g, gx) = \pi_v(g, gx)$ for all $x \in X$ and for all $g \in F/N$. Hence,

$$\pi_u = \pi_v.$$

$\Leftarrow$. Now suppose that $\pi_u = \pi_v$. Then for every $g \in F/N$ and for every $x \in X$, $\pi_u(g, gx) = \pi_v(g, gx)$, so by Lemma 5.5.4, $\mu\left(\frac{\partial u}{\partial x}\right) = \mu\left(\frac{\partial v}{\partial x}\right)$. By the Fundamental Theorem of Calculus,

$$\begin{aligned} \mu(u) - \mu(v) &= \mu(u-1) - \mu(v-1) = \sum_{x \in X} \mu\left(\frac{\partial u}{\partial x}\right)\big(\mu(x) - 1\big) - \sum_{x \in X} \mu\left(\frac{\partial v}{\partial x}\right)\big(\mu(x) - 1\big) \\ &= \sum_{x \in X} \left[\mu\left(\frac{\partial u}{\partial x}\right) - \mu\left(\frac{\partial v}{\partial x}\right)\right]\big(\mu(x) - 1\big) = 0. \end{aligned}$$

Thus $\mu(u) = \mu(v)$ and hence $\mu(uv^{-1}) = 1$ in F/N. Differentiating on both sides gives for all $x \in X$

$$\mu\left(\frac{\partial uv^{-1}}{\partial x}\right) = 0.$$

Finally, by Fox's theorem, this yields $uv^{-1} \in N'$, i.e., $u = v$ in N'. □

This theorem is the key to solving the Word Problem in free solvable groups. The idea is that in order to check whether two words are equal, it is enough to check whether their flows are equal, and this can be computed using Fox derivatives. What remains to be done is to give an efficient algorithm to compute Fox derivatives.

5.5.2 On Computing Fox Derivatives Efficiently

As shown before, in order to compute the Magnus embedding of a word $w \in F/N'$, one needs to compute its Fox derivatives $\frac{\partial w}{\partial x_i}$ in F/N with respect to each generator $x_i \in X$. The special case of the product rule (equation (5.5)) provides an algorithm whose complexity is polynomial in that of the Word Problem in F/N, as well as in word length. To wit, given a word $w = x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k} \in F/N$, one only needs to compute $\sum\limits_{x_{i_m}=x} \epsilon_m x_{i_1}^{\epsilon_1} \cdots x_{i_m}^{\frac{1}{2}(\epsilon_m - 1)}$, i.e., one needs to combine like terms in this sum, so one needs to recognise like terms. In other words, one needs to compare all terms in the sum to each other and determine which are equal. This amounts to solving $O(k^2)$ Word Problems in F/N. Our interest lies in free solvable groups $F/F^{(d+1)}$ (i.e., $N = F^{(d)}$). It is important to note here that suing the algorithm presented here in order to solve a Word Problem in $F/F^{(d)}$, one needs to use the Magnus embedding. Thus one needs to compute *all* the Fox derivatives again, "one level down". One can compute Fox derivatives in the abelianisation F/F' in linear time. Denote the time function of solving the Word Problem in $F/F^{(d)}$ by $T_d(|w|)$. Considering that

$T_1(|w|) = |w|$, it is then easy to see that

$$T_{d+1} = r|w|^2 T_d.$$

Solving this recursion gives an algorithm in $O(r^d|w|^{2d})$. While this algorithm is polynomial, its complexity is not uniform over the rank r free solvable groups, since its degree depends on the degree of solvability. This is why one wants a more subtle method to compute Fox derivatives efficiently. Miasnikov, Roman'kov, Ushakov and Vershik use in [16] certain data structures which enables them to give an algorithm with uniform complexity for computing Fox derivatives, and hence for computing the Magnus embedding.

Definition 5.5.7. *Let $G = \langle X \rangle$ be a group and let $D \subseteq F(X)$ be finite. A G-*partition *of D is a partition of non-empty subsets D_i of D, such that $u, v \in D_i$ whenever $u = v$.*

This partition is unique up to reordering of the factors. Observe that if H is a quotient of G, the G-partition of D is finer than the H-partition. If $D = \{w_1, \dots, w_n\}$ is ordered, then the G-partition of D can be represented by the function $P : \{0, \dots, n\} \to \{0, \dots, n\}$, where $P(j) = i$, where i is minimal such that $w_i = w_j$ in G.

Given a word $w = x_{i_1}^{\epsilon_1} \cdots x_{i_n}^{\epsilon_n}$ in $F(X)$, let

$$D_w = \{1, x_{i_1}^{\epsilon_1}, x_{i_1}^{\epsilon_1} x_{i_2}^{\epsilon_2}, \dots, x_{i_1}^{\epsilon_1} \cdots x_{i_n}^{\epsilon_n}\}.$$

Order D_w as follows: let $w_0 = 1$, $w_n = w$ and in general, $w_k = x_{i_1}^{\epsilon_1} \cdots x_{i_k}^{\epsilon_k}$ and set $w_i < w_j$ if $i < j$. The first goal is to give an efficient algorithm to compute the $F/F^{(d)}$-partition of D_w, or equivalently, to compute efficiently the corresponding function P.

Lemma 5.5.8. *The $F/F^{(d)}$-partition of D_w can be computed in time $O(rd|w|^3)$.*

Proof. The proof is done by induction on the degree d. Compute first the F/F'-partition of D_w. This is easily done, since it is the Word Problem in a free abelian group and hence can be done in $O(|w|)$. Now, suppose that the $F/F^{(d-1)}$ partition of D_w has been computed in the form of the function P_{d-1}. Since $F/F^{(d-1)}$ is a quotient of $F/F^{(d)}$, in order to obtain a partition for $F/F^{(d)}$, it is enough to refine the one for $F/F^{(d-1)}$. In other words, one only needs to compare in $F/F^{(d)}$ elements which are equal in $F/F^{(d-1)}$.

Suppose that $w_s, w_t \in D_w$ with $s < t$ are such that $w_s = w_t$ in $F/F^{(d-1)}$. By Corollary 5.5.5, in order to check whether $w_s = w_t$ in $F/F^{(d)}$, it is enough to check that $\frac{\partial w_s}{\partial x_k} = \frac{\partial w_t}{\partial x_k}$ for all $x_k \in X$. For each $1 \le k \le r$, using the special case of the product rule (equation (5.5)), we get the following:

$$\begin{aligned}
\frac{\partial w_s}{\partial x_k} - \frac{\partial w_t}{\partial x_k} &= \left(\sum_{\substack{1\le j\le s \\ i_j=k,\epsilon_j=1}} x_{i_1}^{\epsilon_1}\cdots x_{i_{j-1}}^{\epsilon_{j-1}} - \sum_{\substack{1\le j\le s \\ i_j=k,\epsilon_j=-1}} x_{i_1}^{\epsilon_1}\cdots x_{i_j}^{\epsilon_j} \right) - \\
&\quad \left(\sum_{\substack{1\le j\le t \\ i_j=k,\epsilon_j=1}} x_{i_1}^{\epsilon_1}\cdots x_{i_{j-1}}^{\epsilon_{j-1}} - \sum_{\substack{1\le j\le t \\ i_j=k,\epsilon_j=-1}} x_{i_1}^{\epsilon_1}\cdots x_{i_j}^{\epsilon_j} \right) \\
&= \sum_{\substack{s+1\le j\le t \\ i_j=k,\epsilon_j=-1}} x_{i_1}^{\epsilon_1}\cdots x_{i_j}^{\epsilon_j} - \sum_{\substack{s+1\le j\le t \\ i_j=k,\epsilon_j=1}} x_{i_1}^{\epsilon_1}\cdots x_{i_{j-1}}^{\epsilon_{j-1}} \\
&= \sum_{\substack{s+1\le j\le t \\ i_j=k,\epsilon_j=-1}} w_j - \sum_{\substack{s+1\le j\le t \\ i_j=k,\epsilon_j=1}} w_{j-1}. \qquad (5.7)
\end{aligned}$$

Computing this as a formal expression (i.e., writing it out) can be done in linear time. In order to check that the expression in equation5.7 is equal to zero as an element of $\mathbb{Z}\left(F/F^{(d-1)}\right)$, we represent it in the standard group ring form $\sum_{g\in F/F^{(d-1)}} m_g g$

with $m \in \mathbb{Z}$ and check that all coefficients m_g are equal to zero. First one needs to describe a procedure for converting equation (5.7) in this form. The only obstacle to doing so efficiently is recognising and collecting similar terms. Note that from the way the function P_{d-1} was defined,

$$\sum_{\substack{s+1 \le j \le t \\ i_j = k, \epsilon_j = -1}} w_j - \sum_{\substack{s+1 \le j \le t \\ i_j = k, \epsilon_j = 1}} w_{j-1} = \sum_{\substack{s+1 \le j \le t \\ i_j = k, \epsilon_j = -1}} w_{P(j)} - \sum_{\substack{s+1 \le j \le t \\ i_j = k, \epsilon_j = 1}} w_{P(j)-1}. \tag{5.8}$$

Computing the latter as a formal expression can also be done in linear time. Now, $w_p = w_q$ if and only if $p = q$ and so like terms can be recognised easily. Hence the standard group ring presentation of the expression in equation (5.8) can be computed in time $O(|w|)$. Thus checking that $\frac{\partial w_s}{\partial x_k} = \frac{\partial w_t}{\partial x_k}$ in $F/F^{(d-1)}$ takes time $O(|w|)$. In order to check that $w_s = w_t$ in $F/F^{(d)}$, one needs to check that $\frac{\partial w_s}{\partial x_k} = \frac{\partial w_t}{\partial x_k}$ for all $1 \le k \le r$. Hence, checking that $w_s = w_t$ in $F/F^{(d)}$ can be done in time $O(r|w|)$.

Now, given an $F/F^{(d-1)}$-partition for D_w, compute the $F/F^{(d)}$-partition of D_w as follows: for each $1 \le s < t \le |w|$, for which $w_s = w_t$ in $F/F^{(d-1)}$ decide whether $w_s = w_t$ in $F/F^{(d)}$. There are at most $\binom{|w|}{2} \in O(|w^2|)$ such pairs (s,t) and checking equality for each is done in time $O(r|w|)$. Thus, given the $F/F^{(d-1)}$-partition of D_w, one can compute its $F/F^{(d)}$-partition in $O(r|w|^3)$.

As mentioned in the beginning, in order to compute the $F/F^{(d)}$-partition of D_w, one computes iteratively, the F/F'-partition, then the $F/F^{(2)}$, etc. There are d iterations to be made in total, and each takes time $O(r|w|^3)$, so computing the $F/F^{(d)}$-partition of D_w can be done in time $O(rd|w|^3)$.

□

Proposition 5.5.9. *Given a word $w \in F$ and some $1 \leq k \leq r$, the standard group ring form of the Fox derivative $\frac{\partial w}{\partial x_k}$ in $\mathbb{Z}\left(F/F^{(d)}\right)$ can be computed in $O(rd|w|^3)$.*

Proof. Let $w = x_{i_1}^{\epsilon_1} \cdots x_{i_n}^{\epsilon_n} \in F$. First, using the product rule, compute

$$\frac{\partial w}{\partial x_k} = \sum_{x_{i_m} = x_k} \epsilon_m x_{i_1}^{\epsilon_1} \cdots x_{i_m}^{\frac{1}{2}(\epsilon_m - 1)} \text{ in } \mathbb{Z}F.$$

In order to convert this expression to the standard group ring form in $\mathbb{Z}\left(F/F^{(d)}\right)$, one needs to combine terms which are equal in $F/F^{(d)}$. Compute the $F/F^{(d)}$ partition of D_w. By Lemma 5.5.8 this can be done in time $O(rd|w|^3)$. Now for any two summands, check whether they are in the same subset of the partition. If so, add their coefficients. As there are $\binom{|w|}{2}$ pairs of summands to be compared, this can be done in time $O(|w|^2)$. This yields the standard group ring form of $\frac{\partial w}{\partial x_k}$ in $F/F^{(d)}$ with all computations done in time $O(rd|w|^3)$. □

This has two useful corollaries.

Corollary 5.5.10. *Given a word $w \in F$, the Magnus embedding $\varphi(w)$ can be computed in time $O(r^2 d|w|^3)$.*

Proof. Recall that for $w \in F$,

$$\varphi(w) = \begin{pmatrix} \mu(w) & \sum\limits_{x \in X} \frac{\partial w}{\partial x} t_x \\ 0 & 1 \end{pmatrix}.$$

By Proposition 5.5.9, $\frac{\partial w}{\partial x}$ can be computed for each $x \in X$ in time $O(rd|w|^3)$, hence computing $\frac{\partial w}{\partial x}$ for all $x \in X$ can be done in time $O(r^2 d|w|^3)$. □

Another corollary of Proposition 5.5.9 is a polynomial time algorithm to solve the Word problem in free solvable groups.

Corollary 5.5.11. *The Word Problem in the free solvable group of rank r and degree d, $F/F^{(d)}$ is decidable in time $O(rd|w|^3)$, where w is the input word.*

Proof. By Theorem 5.5.6, two words u and v are equal in $F/F^{(d)}$ if and only if $\pi_u = \pi_v$ in the Cayley graph of $F/F^{(d-1)}$. Thus, in order to check whether a given word w is the identity in $F/F^{(d+1)}$, it is enough to check that the flow induced by w in the Cayley graph of $F/F^{(d)}$ is the same as that induced by the identity. By Corollary 5.5.5, this amounts to checking whether $\frac{\partial w}{\partial x} = 0$ for all $x \in X$. This can be done in time $O(rd|w|^3)$, as it is enough to compute their group ring representation as in Lemma 5.5.9 and to check whether each of them is zero. □

CHAPTER 6
Conjugacy in Wreath Products and Free Solvable Groups

This section comprises the main results involving original work on the part of the author.

6.1 The Conjugacy Problem in Wreath Products

As mentioned before, the decidability of the Conjugacy Problem in wreath products was shown by Jane Matthews in [14]. The algorithm presented here is based on the one she gave, but a crucial part has been modified, in order to make its complexity polynomial. The general idea, however is the same.

6.1.1 A Conjugacy Criterion for Wreath Products

Here we present work which is entirely due to Matthews, namely the derivation of a good conjugacy criterion for a wreath product $A\mathrm{wr}B$.

Let $x = bf, y = cg \in A\mathrm{wr}B$, where $b, c \in B$ and $f, g \in A$. Denote $\mathrm{supp}(f) = \{b_1, \dots, b_n\}$ and $\mathrm{supp}(g) = \{\beta_1, \dots, \beta_m\}$ using the order described in Section 3.4. Recall that all elements are given as words in generators. Let $\bar{b}$ and $\bar{\beta}$ be the longest elements in $\mathrm{supp}(f)$ and in $\mathrm{supp}(g)$, and $\bar{a}$ and $\bar{\alpha}$ be the longest element in the range of f and of g, respectively.

For each left $\langle b\rangle$-coset in B that intersects $\mathrm{supp}(f) \cup \mathrm{supp}(g)$, choose one coset representative t_i and let $T_b = \{t_i\}_{i \in I}$. Since $b_i, b_j \in \mathrm{supp}(f) \cup \mathrm{supp}(g)$ are in the same coset if and only if $b_i b_j^{-1} = b^k$ for some k, this is an instance of the Power Problem in B. To find T_b one needs to check the Power Problem $\binom{n+m}{2}$ times (for all pairs

$(b_i, b_j))$. Hence it takes time $\binom{n+m}{2}T_{PB}(2|\bar{b}| + |b|)$, where T_{PB} is the time function for the Power Problem in B. Denote by $\bar{t}$ the longest element in T_b. For each $\gamma \in B$ and $i \in I$, associate with T_b the map $\pi_i^{(\gamma)} : A^{(B)} \to A$ defined by

$$\pi_i^{(\gamma)}(h) = \begin{cases} \prod_{j=0}^{N-1} h(t_i b^j \gamma^{-1}) & \text{if } b \text{ is of finite order } N \\ \\ \prod_{j=-\infty}^{\infty} h(t_i b^j \gamma^{-1}) & \text{if } b \text{ is of infinite order} \end{cases}$$

Note that in the above all the products are finite, since h has finite support. Denote $\pi_i^{(1)}$ by π_i. With this, we are ready to give a conjugacy criterion.

Theorem 6.1.1. *Let A, B be finitely generated groups. Two elements $x = bf, y = cg$ in $A\mathrm{wr}B$ are conjugate if and only if there exists $d \in B$ such that for all $t_i \in T_b$ the following hold:*

(1) $db = cd$,

(2) when the order of b is finite, $\pi_i^{(d)}(g)$ is conjugate to $\pi_i(f)$ in A,

(3) when the order of b is infinite, $\pi_i^{(d)}(g) = \pi_i(f)$ in A.

Before proving this theorem we need a few technical lemmas.

Lemma 6.1.2. *Two elements $x = bf, y = cg \in A\mathrm{wr}B$ are conjugate via $z = dh$ if and only if*

1. $db = cd$

2. $g^d = h^b f h^{-1}$.

Proof. Note that

$$\begin{aligned} zx &= dhbf = dbh^b f, \text{ and} \\ yz &= cgdh = cdg^d h. \end{aligned}$$

Hence, $zx = yz$ if and only if $dbh^b f = cdg^d h$. This is equivalent to

$$db = cd \text{ and } g^d = h^b f h^{-1}.$$

□

Lemma 6.1.3. *Let $b, d \in B$ and $f, g, h \in A^{(B)}$ be such that $g^d = h^b f h^{-1}$. Then for all coset representatives $t_i \in T_b$and for all integers $n \geq 0$ and m:*

$$\prod_{j=m}^{m+n} g\big(t_i b^j d^{-1}\big) = h\big(t_i b^{m-1}\big) \left(\prod_{j=m}^{m+n} f\big(t_i b^j\big) \right) h^{-1}\big(t_i b^{m+n}\big).$$

Proof. Fix i. Since $g^d = h^b f h^{-1}$, then for any integer j,

$$\begin{aligned} g\big(t_i b^j d^{-1}\big) &= g^d\big(t_i b^j\big) = h^b f h^{-1}\big(t_i b^j\big) = h^b\big(t_i b^j\big) f\big(t_i b^j\big) h^{-1}\big(t_i b^j\big) \\ &= h\big(t_i b^{j-1}\big) f\big(t_i b^j\big) h^{-1}\big(t_i b^j\big). \end{aligned}$$

Now taking the product over $m \le j \le m+n$ gives,

$$\begin{aligned}
&\prod_{j=m}^{m+n} h\big(t_i b^{j-1}\big) f\big(t_i b^j\big) h^{-1}\big(t_i b^j\big)\\
= \;& h\big(t_i b^{m-1}\big) f\big(t_i b^m\big) h^{-1}\big(t_i b^m\big) h\big(t_i b^m\big) f\big(t_i b^{m+1}\big) h^{-1}\big(t_i b^{m+1}\big) \cdots h^{-1}\big(t_i b^{m+n}\big)\\
= \;& h\big(t_i b^{m-1}\big) f\big(t_i b^m\big) f\big(t_i b^{m+1}\big) h^{-1}\big(t_i b^{m+1}\big) \cdots h\big(t_i b^{m+n-1}\big) f\big(t_i b^{m+n}\big) h^{-1}\big(t_i b^{m+n}\big)\\
= \;& h\big(t_i b^{m-1}\big) \left(\prod_{j=m}^{m+n} f\big(t_i b^j\big) \right) h^{-1}\big(t_i b^{m+n}\big).
\end{aligned}$$

□

Finally, we are ready to prove Theorem 6.1.1.

Proof of Theorem 6.1.1. $\Rightarrow$. Suppose that $x = bf$ and $y = cd$ are conjugate in $A\mathrm{wr}B$, i.e., suppose there exists a $z = dh$ such that $zx = yz$. By Lemma 6.1.2 this is true if and only if $db = cd$ and $g^d = h^b f h^{-1}$. Hence condition 1) is satisfied.

First, suppose that b is of finite order N. By setting $m = 0$ and $n = N-1$ in Lemma 6.1.3 one obtains:

$$\prod_{j=0}^{N-1} g\big(t_i b^j d^{-1}\big) = h\big(t_i b^{-1}\big) \left(\prod_{j=0}^{N-1} f\big(t_i b^j\big) \right) h^{-1}\big(t_i b^{N-1}\big).$$

Since the order of b is N, $b^{N-1} = b^{-1}$. Then recalling the definition of the maps $\pi_i^{(d)}$ and π_i, the above equation becomes

$$\pi_i^{(d)}(g) = h\big(t_i b^{-1}\big) \pi_i(f) h^{-1}\big(t_i b^{-1}\big).$$

Hence $\pi_i^{(d)}(g)$ and $\pi_i(f)$ are conjugate in A.

Now, suppose that the order of b is infinite. Since g and f are of finite support, there exist numbers M and $N \geq 0$ such that

$$\pi_i^{(d)}(g) = \prod_{j=M}^{M+N} g\left(t_i b^j d^{-1}\right) \text{ and } \pi_i(f) = \prod_{j=M}^{M+N} f\left(t_i b^j\right).$$

Setting $m = M$ and $n = N$ in the formula from Lemma 6.1.3, it follows that

$$\pi_i^{(d)}(g) = h\left(t_i b^{M-1}\right)\pi_i(f)h^{-1}\left(t_i b^{M+N}\right). \tag{6.1}$$

Claim that $h\left(t_i b^{M-1}\right)$ and $h^{-1}\left(t_i b^{M+N}\right)$ must be the identity in A. Indeed, setting, for all integers $k \geq 0$, $m = M - k$ and $n = N + k$ in Lemma 6.1.3 yields

$$\prod_{j=M-k}^{M+N} g\left(t_i b^j d^{-1}\right) = h\left(t_i b^{M-k-1}\right)\left(\prod_{j=M-k}^{M+N} f\left(t_i b^j\right)\right)h^{-1}\left(t_i b^{M+N}\right).$$

Notice that M and N were chosen large enough that $\pi_i^{(d)}(g) = \prod_{j=M}^{M+N} g\left(t_i b^j d^{-1}\right)$ and $\pi_i(f) = \prod_{j=M}^{M+N} f\left(t_i b^j\right)$. Hence, for all integers $k \geq 0$,

$$\pi_i^{(d)}(g) = h\left(t_i b^{M-k-1}\right)\pi_i(f)h^{-1}\left(t_i b^{M+N}\right). \tag{6.2}$$

Since h has finite support, there must be a $k_1 \geq 0$ for which $h\left(t_i b^{M-k_1-1}\right) = 1$ in A. Thus, setting $k = k_1$ in equation (6.2) gives

$$\begin{aligned}
\pi_i^{(d)}(g) &= \pi_i(f)h^{-1}\left(t_i b^{M+N}\right) \\
&= h^{-1}\left(t_i b^{M-1}\right)h\left(t_i b^{M-1}\right)\pi_i(f)h^{-1}\left(t_i b^{M+N}\right) \\
&= h^{-1}\left(t_i b^{M-1}\right)\pi_i^{(d)}(g).
\end{aligned}$$

Hence, $h\big(t_i b^{M-1}\big) = 1$. Similarly, setting for all integers $k \geq 0$, $m = M$ and $n = N + k$ yields

$$\prod_{j=M}^{M+N+k} g\big(t_i b^j d^{-1}\big) = h\big(t_i b^{M-1}\big)\left(\prod_{j=M}^{M+N+k} f\big(t_i b^j\big)\right) h^{-1}\big(t_i b^{M+N+k}\big),$$

i.e.,

$$\pi_i^{(d)}(g) = h\big(t_i b^{M-1}\big)\pi_i(f)h^{-1}\big(t_i b^{M+N+k}\big).$$

But, again there must be an integer $k_2 \geq 0$ for which $h^{-1}\big(t_i b^{M+N+k_2}\big) = 1$. Then the above equation becomes

$$\begin{aligned} \pi_i^{(d)}(g) &= h\big(t_i b^{M-1}\big)\pi_i(f) \\ &= h\big(t_i b^{M-1}\big)\pi_i(f)h^{-1}\big(t_i b^{M+N}\big)h\big(t_i b^{M+N}\big) \\ &= \pi_i^{(d)}(g)h\big(t_i b^{M+N}\big). \end{aligned}$$

Hence $h^{-1}\big(t_i b^{M+N}\big) = 1$ which proves the claim. Finally, we have that $h\big(t_i b^{M-1}\big) = 1$ and $h^{-1}\big(t_i b^{M+N}\big) = 1$ and hence equation (6.1) gives $\pi_i^{(d)}(g) = \pi_i(f)$.

$\Leftarrow$. Now suppose that there is some $d \in B$, for which $db = cd$.

Suppose further that b is of finite order N and that $\pi_i^{(d)}(g)$ and $\pi_i(f)$ are conjugate in A for all i via $\alpha_i \in A$, i.e., $\pi_i^{(d)}(g) = \alpha_i \pi_i(f)\alpha_i^{-1}$. Define $h \in A^{(B)}$ by setting, for any $\xi = t_i b^k \in B$

$$h(\xi) = h(t_i b^k) = \left(\prod_{j=0}^{k} g\big(t_i b^j d^{-1}\big)\right)^{-1} \alpha_i \left(\prod_{j=0}^{k} f(t_i b^j)\right)$$

Lemma. *The function $h : B \to A$ is well-defined.*

Proof. One needs to show that the definition of h does not depend on the choice of coset representatives. Let $\overline{T_b} = \{s_i\}$ be another set of coset representatives for $\langle b \rangle$ in

B and let $\xi = t_i b^{k_1} = s_i b^{k_2}$. Note that $s_i = t_i b^l$ for some $l \in \mathbb{Z}$. The maps π_i are defined in terms of the set of coset representatives T_b, so to define h in terms of the new set of coset representatives, one needs to define the maps π_i in terms of this new set as well. Let

$$\overline{\pi}_i^{(d)}(g) = \prod_{j=0}^{N-1} g(s_i b^j d^{-1}) = \prod_{j=0}^{N-1} g(t_i b^{j+l} d^{-1}) = \prod_{j=l}^{N+l-1} g(t_i b^j d^{-1}), \text{ and}$$

$$\overline{\pi}_i(f) = \prod_{j=0}^{N-1} f(s_i b^j) = \prod_{j=0}^{N-1} f(t_i b^{j+l}) = \prod_{j=l}^{N+l-1} f(t_i b^j).$$

Now,

$$\begin{aligned} \pi_i^{(d)}(g) &= \prod_{j=0}^{N-1} g(t_i b^j d^{-1}) = \left(\prod_{j=0}^{l-1} g(t_i b^j d^{-1})\right)\left(\prod_{j=l-1}^{N+l-1} g(t_i b^j d^{-1})\right)\left(\prod_{j=N}^{N+l-1} g(t_i b^j d^{-1})\right)^{-1} \\ &= \left(\prod_{j=0}^{l-1} g(t_i b^j d^{-1})\right)\overline{\pi}_i^{(d)}(g)\left(\prod_{j=0}^{l-1} g(t_i b^j d^{-1})\right)^{-1}. \end{aligned}$$

Similarly,

$$\begin{aligned} \pi_i(f) &= \prod_{j=0}^{N-1} f(t_i b^j) = \left(\prod_{j=0}^{l-1} f(t_i b^j)\right)\left(\prod_{j=l-1}^{N+l-1} f(t_i b^j)\right)\left(\prod_{j=N}^{N+l-1} f(t_i b^j)\right)^{-1} \\ &= \left(\prod_{j=0}^{l-1} f(t_i b^j)\right)\overline{\pi}_i(f)\left(\prod_{j=0}^{l-1} f(t_i b^j)\right)^{-1}. \end{aligned}$$

Denote $\prod_{j=0}^{l-1} g(t_i b^j d^{-1})$ by γ_1 and $\prod_{j=0}^{l-1} f(t_i b^j)$ by γ_2. Then, it follows from the above and from the fact that $\pi_i^{(d)}(g)$ and $\pi_i(f)$ are conjugate via α_i that

$$\overline{\pi}_i^{(d)}(g) = \gamma_1^{-1}\pi_i^{(d)}(g)\gamma_1 = \gamma_1^{-1}\alpha_i\pi_i(f)\alpha_i^{-1}\gamma_1 = \gamma_1^{-1}\alpha_i\gamma_2\overline{\pi}_i(f)\gamma_2\alpha_i^{-1}\gamma_1.$$

Thus $\pi_i^{(d)}(g) = \overline{\alpha}_i \overline{\pi}_i(f) \overline{\alpha}_i^{-1}$, with

$$\overline{\alpha}_i = \left(\prod_{j=0}^{l-1} g(t_i b^j d^{-1}) \right)^{-1} \alpha_i \left(\prod_{j=0}^{l-1} f(t_i b^j) \right).$$

With this, $h(s_i b^{k_2})$ is given by

$$h(s_i b^{k_2}) = \left(\prod_{j=0}^{k_2} g(s_i b^j d^{-1}) \right)^{-1} \overline{\alpha}_i \left(\prod_{j=0}^{k_2} f(s_i b^j) \right)^{-1}.$$

Keeping in mind that $s_i = t_i b^l$ and hence that $k_2 = k_1 - l$, the above becomes

$$\begin{aligned}
h(s_i b^{k_2}) &= \left(\prod_{j=0}^{k_1-l} g(s_i b^j d^{-1}) \right)^{-1} \overline{\alpha}_i \left(\prod_{j=0}^{k_1-l} f(s_i b^j) \right)^{-1} \\
&= \left(\prod_{j=0}^{k_1-l} g(s_i b^j d^{-1}) \right)^{-1} \left(\prod_{j=0}^{l-1} g(t_i b^j d^{-1}) \right)^{-1} \alpha_i \left(\prod_{j=0}^{l-1} f(t_i b^j) \right) \left(\prod_{j=0}^{k_1-l} f(s_i b^j) \right)^{-1} \\
&= \left(\prod_{j=0}^{k_1} g(t_i b^j d^{-1}) \right)^{-1} \alpha_i \left(\prod_{j=0}^{k_1} f(t_i b^j) \right) \\
&= h(t_i b^{k_1}).
\end{aligned}$$

Hence, h is well-defined. □

Now for any $\xi = t_i b^k \in B$,

$$\begin{aligned}
\left(h^b f h^{-1}\right)(\xi) &= h^b(\xi) f(\xi) h^{-1}(\xi) = h\left(t_i b^{k-1}\right) f\left(t_i b^k\right) h^{-1}\left(t_i b^k\right) \\
&= \left(\prod_{j=0}^{k-1} g\left(t_i b^j d^{-1}\right) \right)^{-1} \alpha_i \left(\prod_{j=0}^{k-1} f\left(t_i b^j\right) \right) f\left(t_i b^k\right) \left(\prod_{j=0}^{k} f\left(t_i b^j\right) \right)^{-1} \alpha_i^{-1} \left(\prod_{j=0}^{k} g\left(t_i b^j d^{-1}\right) \right) \\
&= \left(\prod_{j=0}^{k-1} g\left(t_i b^j d^{-1}\right) \right)^{-1} \alpha_i \left(\prod_{j=0}^{k} f\left(t_i b^j\right) \right) \left(\prod_{j=0}^{k} f\left(t_i b^j\right) \right)^{-1} \alpha_i^{-1} \left(\prod_{j=0}^{k} g\left(t_i b^j d^{-1}\right) \right) \\
&= \left(\prod_{j=0}^{k-1} g\left(t_i b^j d^{-1}\right) \right)^{-1} \left(\prod_{j=0}^{k} g\left(t_i b^j d^{-1}\right) \right) = g\left(t_i b^k d^{-1}\right) = g^d\left(t_i b^k\right) \\
&= g^d(\xi).
\end{aligned}$$

Hence, $h^b f h^{-1} = g^d$ and since $db = cd$, by Lemma 6.1.2, $zx = yz$, for $z = dh$ with h constructed as above.

Now, suppose that the order of b is infinite and that $\pi_i^{(d)}(g) = \pi_i(f)$. Define $h \in A^{(B)}$ by setting

$$h(t_i b^k) = \left(\prod_{j \le k} g\left(t_i b^j d^{-1}\right)\right)^{-1} \left(\prod_{j \le k} f\left(t_i b^j\right)\right),$$

for all $t_i \in T$ and for all integers k. Showing that h is well defined is similar to the case where b is of finite order. With this, for any $\xi = t_i b^k \in B$,

$$\begin{aligned}
\left(h^b f h^{-1}\right)(\xi) &= h^b(\xi) f(\xi) h^{-1}(\xi) = h\left(t_i b^{k-1}\right) f\left(t_i b^k\right) h^{-1}\left(t_i b^k\right) \\
&= \left(\prod_{j \le k-1} g\left(t_i b^j d^{-1}\right)\right)^{-1} \left(\prod_{j \le k-1} f\left(t_i b^j\right)\right) f\left(t_i b^k\right) \left(\prod_{j \le k} f\left(t_i b^j\right)\right)^{-1} \left(\prod_{j \le k} g\left(t_i b^j d^{-1}\right)\right) \\
&= \left(\prod_{j \le k-1} g\left(t_i b^j d^{-1}\right)\right)^{-1} \left(\prod_{j \le k} f\left(t_i b^j\right)\right) \left(\prod_{j \le k} f\left(t_i b^j\right)\right)^{-1} \left(\prod_{j \le k} g\left(t_i b^j d^{-1}\right)\right) \\
&= \left(\prod_{j \le k-1} g\left(t_i b^j d^{-1}\right)\right)^{-1} \left(\prod_{j \le k} g\left(t_i b^j d^{-1}\right)\right) = g\left(t_i b^k d^{-1}\right) = g^d\left(t_i b^k\right) \\
&= g^d(\xi).
\end{aligned}$$

Hence, $h^b f h^{-1} = g^d$ and by Lemma 6.1.2 x and y are conjugate with $zx = yz$, where $z = dh$.

□

6.1.2 A polynomial Time Algorithm for Deciding the Conjugacy Problem in Wreath Products

In this section we present an algorithm to solve the Conjugacy Problem in wreath products in polynomial time. It is largely based on the one given by Matthews in

[14]. In [14], the running time of the algorithm was not analysed. We analyse it and show that by changing a main case, the original version of which ran in unbounded time, we obtain a polynomial time algorithm.

Theorem 6.1.4. *Let A and B be finitely generated groups such that the following hold:*

1) *there are decision algorithms for the Conjugacy Problem in A and in B with polynomial time functions, T_{CA}, T_{CB}, respectively;*
2) *there is a decision algorithm for the Power Problem in B with polynomial time function T_{PB}.*

Then the Conjugacy Problem in $A\mathrm{wr}B$ is decidable with complexity

$$O\big(NT_{CA}(N^2) + NT_{CB}(N) + N^2T_{PB}(N)\big),$$

where $N = |x| + |y|$ is the length of the input pair $x, y \in A\mathrm{wr}B$.

Since the conjugacy criterion discussed in Section 6.1.1 involves the maps π_i, an important subroutine is one for computing them effectively.

Lemma 6.1.5. *There is a polynomial time algorithm which computes $\pi_i^{(\gamma)}(f)$.*

Proof. The algorithm is as follows:

Step 1: For each $b_k \in \mathrm{supp}(f)$ check whether there is some j such that $t_i b^j \gamma^{-1} = b_k$, i.e., $t_i^{-1} b_k \gamma = b^j$. This is an instance of the Power Problem in B and so can be done in time $T_{PB}(|\bar{b}| + |\bar{t}| + |b| + |\gamma|)$. If such j exists, look up the corresponding value $a_k = f(b_k)$. Otherwise, a_k does not occur in the product.

Step 2: There are n elements in $\mathrm{supp}(f)$ to perform computations on, so computing $\pi_i^{(\gamma)}(f)$ takes time $nT_{PB}(|\bar{b}| + |\bar{t}| + |b| + |\gamma|)$.

Note that $|\pi_i^{(\gamma)}(f)| \leq n|\bar{a}|$, since each factor in the product $\pi_i^{(\gamma)}(f)$ is in the image of f. Thus, computing $\pi_i^{(\gamma)}(f)$ can be done in time $O(NT_{PB}(N))$ and the output is of length N^2. Observe that he same bounds hold for computing $\pi_i^{(d)}(g)$, since $N = |x| + |y|$. □

In particular, the above gives the complexity of the algorithm, which will be used in the sequel. Finally, we present an algorithm to decide the Conjugacy Problem in wreath products.

Proof of Theorem 6.1.4. The algorithm is as follows.

Step 1. Determine whether b and c are conjugate in B. This takes time $T_{CB}(|b| + |c|) \in O(T_{CB}(N))$. If not, x and y are not conjugate. If b and c are conjugate in B, let $d \in B$ be such that $db = cd$ (it is not required to find this d).

Step 2. Recall that when B was partitioned in $\langle b \rangle$ left cosets, we only partition $\operatorname{supp}(f)$, so that $t_i \in \operatorname{supp}(f)$ and so in our estimations, $|\bar{t}| \leq |x|$.

Case 1: $g = 1$. Then $\pi_i^{(d)}(g) = 1$, so x and y are conjugate if and only if $\pi_i(f) = 1$. To check this compute $\pi_i(f)$ as in Lemma 6.1.5 and solve the Word Problem in A. This takes time

$$O\Big(|x|\big(T_{PB}(|x|) + |x|\big) + T_{CA}(|x||x| + 1)\Big) \in O\big(NT_{PB}(N) + T_{CA}(N^2)\big). \tag{6.3}$$

Case 2: $g \neq 1$, and $\pi_i(f) = 1$ for all i. In order to check the latter, simply compute $\pi_i(f)$ for all i. This will take time $|x|T_{PB}(|x|)$. Then x is conjugate to y if and only if $\pi_i^{(d)}(g) = 1$ for all i. We proceed to show that we need not know what d actually is – its existence is enough. Since $db = cd$, $g(t_i b^j d^{-1}) = g(t_i d^{-1} c^j)$. Consider the maps $\bar{\pi}_i(g) = \prod_j g(s_i c^j)$ where s_i is a set of coset representatives of $\langle c\rangle$ in B. Clearly, if $\bar{\pi}_i = 1$ for some set of coset representatives it will be 1 for any set, since changing the set of coset representatives will only produce a cyclic permutation of the factors in the product. The induced set of coset representatives is $\{t_i d^{-1}\}$, but in order to eliminate d, we choose another, say $T_c = \{s_i\}$, the same way T_b was chosen. Then x and y are conjugate if and only if $\bar{\pi}_i(g) = 1$ for all i. Finding T_c takes time $\binom{m}{2}T_{PB}(|c|)$, as discussed before. Thus checking that $\bar{\pi}_i(g) = 1$ takes time

$$O\big(|y|^2 T_{PB}(|y|) + T_{CA}(|y|^2)\big) \in O\big(N^2 T_{PB}(N) + T_{CA}(N^2)\big). \tag{6.4}$$

Case 3: $g \neq 1$ and some $\pi_i(f) \neq 1$. There are two subcases:

1) *The order of* b *is finite.* Let $\bar{\pi}_i$ be as in Case 2 with coset representatives in $\{s_i\}$. Observe that changing coset representatives amounts to a cyclic permutation of the factors in the product, so $\bar{\pi}_i^{(d)}(g)$ is conjugate to $\pi_i^{(d)}(g)$. Thus $\pi_i(f)$ is conjugate to $\pi_i^{(d)}(g)$ if and only if $\pi_i(f)$ is conjugate to $\bar{\pi}_i^{(d)}(g)$. Checking this involves computing $\bar{\pi}_i(g)$ and $\pi_i(f)$ and solving the corresponding Conjugacy Problem in A. This takes time $O\big(|y|^2 T_{PB}(3|y|) + |y| T_{PB}(3|y|) + |x| T_{PB}(3|x|) + T_{CA}(|y|^2 + |x|^2)\big)$. Simplifying this, we get a complexity of

$$O\big(|y|^2 T_{PB}(|y|) + |x| T_{PB}(|x|) + T_{CA}(|x|^2 + |y|^2)\big) \in O\big(N^2 T_{PB}(N) + T_{CA}(N^2)\big). \tag{6.5}$$

2) *The order of* b *is infinite.* Let k be a fixed integer such that $\pi_k(f) \neq 1$ (such a k must be found already in the beginning of Case 3). We proceed to check that $\pi_k(f) = \pi_k^{(d)}(g)$ without finding d. Claim that it suffices to check whether $\pi_i^{(d)}(g) = \pi_i(f)$ for certain d to be described in detail later. If for some d,

$\pi_k^{(d)}(g) = 1$, then it will certainly not be equal to $\pi_i(f)$. Hence we only consider d for which $\pi_k^{(d)}(g) = \prod_j g(t_k b^j d^{-1}) \neq 1$. In this case, there is some integer l for which $g(t_k b^l d^{-1}) \neq 1$. Then $t_k b^l d^{-1} = \beta_p$ for some $\beta_p \in \operatorname{supp}(g)$ and so $d = \beta_p^{-1} t_k b^l$. It would suffice to check, for all d of the form $d = \beta_p^{-1} t_k b^l$ satisfying $db = cd$, whether $\pi_i(f) = \pi_i^{(d)}(g)$.

In order to check the former, we need to check for all $\beta_p \in \operatorname{supp}(g)$ whether $\beta_p^{-1} t_k b^l b = c\beta_p^{-1} t_k b^l$, i.e., it is enough to check whether $\beta_p^{-1} t_k b = c\beta_p^{-1} t_k$, where l no longer appears. This is a Word Problem in B, so it can be solved in time $T_{WB}(2|\beta_p| + 2|t_k| + |b| + |c|)$, where T_{WB} is the time function for a solution to the Word Problem in B (which is polynomial since T_{CB} is polynomial). Thus checking whether d satisfies $db = cd$ can be done in time $O\big(T_{WB}(3(|x| + |y|))\big)$. It remains to check whether $\pi_i(f) = \pi_i^{(d)}(g)$. Notice that

$$\begin{aligned}
\pi_i^{(d)}(g) &= \prod_{j=-\infty}^{\infty} g(t_i b^j d^{-1}) &&= \prod_{j=-\infty}^{\infty} g(t_i b^j b^{-l} t_k^{-1} \beta_p) \\
&= \prod_{j=-\infty}^{\infty} g(t_i b^{j-l} t_k^{-1} \beta_p) &&= \prod_{j=-\infty}^{\infty} g(t_i b^j t_k^{-1} \beta_p) \;=\; \pi_i^{(\beta_p^{-1} t_k)}(g).
\end{aligned}$$

So we need to check whether $\pi_i^{(\beta_p^{-1} t_k)}(g) = \pi_i(f)$ for all $\beta_p \in \operatorname{supp}(g)$ and for all $t_k \in T_b$. Using 6.1.5 this can be done in time $O\Bigg(|y||x|\big(|x|T_{PB}(|x|) + |y|T_{PB}(|y|)\big)\Bigg) \in O\big(N^3 T_{PB}(N)\big)$. Thus, the overall complexity is

$$O\big(|y|T_{WB}(|y| + |x|) + |y||x|\big(|x|T_{PB}(|x|) + |y|T_{PB}(|y|)\big) + T_{CA}(|y|^2 + |x|^2)\big), \quad (6.6)$$

which after simplifying can be seen to be in

$$O\big(NT_{WB}(N) + N^3 T_{PB}(N) + T_{CA}(N^2)\big).$$

The complexity of the Conjugacy Problem in $A\mathrm{wr}B$ is

$$O\big(T_{CA}(N^2) + T_{CB}(N) + NT_{WB}(N) + N^3 T_{PB}(N)\big),$$

which is clearly polynomial since T_{CA}, T_{CB} and T_{PB} are polynomial. □

6.2 The Conjugacy Problem in Free Solvable Groups

The decidability of the Conjugacy Problem in free solvable groups (with no mention of complexity) was proved by Remeslennikov and Sokolov in [17]. The main idea is to embed a free solvable group in a wreath product, and then by decidability of the Conjugacy Problem there, deduce its decidability in free solvable groups. In order to execute this idea, one needs to show that the Magnus embedding is 'conjugacy-preserving', that it is computable in polynomial time and that the Conjugacy Problem in the particular wreath product induced by the embedding is decidable in polynomial time. In order to do the latter, one needs to show that a certain Power Problem is decidable in polynomial time.

Theorem 6.2.1. *The Power Problem in $F/F^{(d+1)}$ is decidable in time $O(dr|w|^3)$, where r is the rank of F.*

Proof. We will prove a slightly stronger version: given two elements $x, y \in F/F^{(d)}$ one can, in time $O(dr(|x| + |y|))$, solve the Power Problem for x and y, and if $x \in \langle y\rangle$ find an $n \in \mathbb{Z}$ such that $x = y^n$.

The proof is by induction on d. For $d = 1$, F/F' is a free abelian group, so the elements x and y can be uniquely presented in the form $x = x_1^{a_1} \cdots x_r^{a_r}$ and $y = x_1^{b_1} \cdots x_r^{b_r}$, where $X = \{x_1, \cdots, x_r\}$ is the basis for F. Obviously, this decomposition can be found in linear time. Then for each $1 \le i \le r$ set $n_i = a_i/b_i$. If all n_i are equal and integer, then $x = y^{n_1}$, as required. Otherwise, $x \notin \langle y \rangle$ and we are done. Clearly, this can be done in time $O(r(|x| + |y|))$.

By Theorem 5.5.11 there exists an algorithm $\mathcal{A}$ that solves the Word Problem in $F/F^{(d)}$ in time $O(dr|w|^3)$, where w is the input word. For given $x, y \in F/F^{(d+1)}$ do the following:

1) If one of x or y is equal to 1, checking whether $x = y^n$ reduces to a Word Problem, since $F/F^{(d+1)}$ is torsion free.
2) Otherwise, since φ is an embedding, $x = y^n$ in $F/F^{(d+1)}$ if and only if $\varphi(x) = \varphi(y)^n$. Let
$$x \mapsto \begin{pmatrix} \mu(x) & t_x \\ 0 & 1 \end{pmatrix} \text{ and } y \mapsto \begin{pmatrix} \mu(y) & t_y \\ 0 & 1 \end{pmatrix}.$$
Notice that
$$\varphi(y)^n = \begin{pmatrix} \mu(y)^n & \big(\mu(y)^{n-1} + \cdots + 1\big)t_y \\ 0 & 1 \end{pmatrix}.$$
Hence $x = y^n$ if and only if
$$\begin{pmatrix} \mu(x) & t_x \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mu(y)^n & \big(\mu(y)^{n-1} + \cdots + 1\big)t_y \\ 0 & 1 \end{pmatrix},$$
which is equivalent to
$$\mu(x) = \mu(y)^n \quad \text{and} \quad t_x = \big(\mu(y)^{n-1} + \cdots + 1\big)t_y.$$
Since $\mu(x), \mu(y) \in F/F^{(d)}$, one can verify by induction in time $O((d-1)r(|x| + |y|)^3)$ whether there exists n such that $\mu(x) = \mu(y)^n$ and, if so, find it. If no such

n exists then $x \notin \langle y \rangle$ and we are done. Note that since $F/F^{(d)}$ is torsion-free, such an n, if it exists, is unique. Indeed, if $\mu(x) = \mu(y)^n$ and $\mu(x) = \mu(y)^m$, then $\mu(y)^n = \mu(y)^m$ and hence $\mu(y)^{n-m} = 1$, which is only possible if $n = m$. Now, one has to check whether the equation $t_x = \big(\mu(y)^{n-1} + \cdots + 1\big)t_y$ holds in T. Since T is a free module, the latter holds if and only if it holds componentwise, which can be verified in time $O(dr|w|^3)$ by the algorithm $\mathcal{A}$. This solves the Power Problem in $F/F^{(d+1)}$.

Note that all steps take time at most $O(dr|w|^3)$, as required. □

Finally, one needs to make sure that the Magnus embedding is conjugacy preserving. We give here the proof of the more general version as presented by Remeslennikov and Sokolov in [17].

Proposition 6.2.2. *Let N be a normal subgroup of the free group F and suppose that F/N is torsion free. Two elements $v, w \in F/N'$ are conjugate in F/N' if and only if their images $\varphi(v), \varphi(w)$ are conjugate in $M(F/N)$.*

Proof. $\Rightarrow$. This is obvious. If there is some $u \in F/N'$ such that $u^{-1}vu = w$, then certainly $\varphi(u)^{-1}\varphi(v)\varphi(u) = \varphi(w)$.

$\Leftarrow$. Now suppose that $\varphi(v)$ and $\varphi(w)$ are conjugate in $M(F/N)$. There are two cases to be considered.

Case 1: $\mu(v) = \mu(w) = 1$. Suppose that $\varphi(v)$ and $\varphi(w)$ are conjugate via the matrix $\begin{pmatrix} \overline{u} & a \\ 0 & 1 \end{pmatrix}$. Let $u \in F/N'$ be such that $\mu(u) = \overline{u}$. Then compute

$$\begin{aligned}
\varphi(u)^{-1}\varphi(v)\varphi(u) &= \begin{pmatrix} \mu(u)^{-1} & -\mu(u)^{-1}t_u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & t_v \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \mu(u) & t_u \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \mu(u)^{-1} & -\mu(u)^{-1}t_u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \mu(u) & t_u + t_v \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \mu(u)^{-1}t_u + \mu(u)^{-1}t_v - \mu(u)^{-1}t_u \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \mu(u)^{-1}t_v \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

On the other hand, since $\varphi(v)$ and $\varphi(w)$ are conjugate,

$$\begin{aligned}
\varphi(w) &= \begin{pmatrix} \overline{u}^{-1} & -\overline{u}^{-1}a_u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & t_v \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \overline{u} & a_u \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \overline{u}^{-1} & -\overline{u}^{-1}a_u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \overline{u} & a_u + t_v \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & \overline{u}^{-1}a_u + \overline{u}^{-1}t_v - \overline{u}^{-1}a_u \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \overline{u}^{-1}t_v \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

Thus, since $\overline{u} = \mu(u)$, $\varphi(w) = \varphi(u)^{-1}\varphi(v)\varphi(u)$. It follows, since φ is injective that $w = u^{-1}vu$ in F/N'.

Case 2: $\mu(v) \neq 1$, $\mu(w) \neq 1$.

Lemma. *Without loss of generality, $\varphi(v)$ and $\varphi(w)$ are conjugate via an idempotent matrix.*

Proof. Let $\varphi(v)$ and $\varphi(w)$ be conjugate with the aid of the matrix $\begin{pmatrix} \overline{u} & a \\ 0 & 1 \end{pmatrix}$. Let $u \in F/N'$ be such that $\mu(u) = \overline{u}$. Observe that the elements $\varphi(v)^{\varphi(u)}$ and $\varphi(w)$ are conjugate with the aid of an idempotent matrix. Indeed, suppose that $b^{-1}\varphi(v)^{\varphi(u)}b = \varphi(w)$. Then,

$$\begin{aligned} & b^{-1}\big(\varphi(u)^{-1}\varphi(v)\varphi(u)\big)b = \varphi(w) \\ \Leftrightarrow \quad & \big(\varphi(u)b\big)^{-1}\varphi(v)\big(\varphi(u)b\big) = \varphi(w) \end{aligned}$$

Now set

$$b = \varphi(u)^{-1}\begin{pmatrix} \overline{u} & a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \overline{u}^{-1} & -\overline{u}^{-1}t_u \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \overline{u} & a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & a - t_u \\ 0 & 1 \end{pmatrix}.$$

Note that $\varphi(v)$ and $\varphi(w)$ are conjugate if and only if $\varphi(v)^{\varphi(u)}$ and $\varphi(w)$ are conjugate. Thus one can assume, without loss of generality, that $\varphi(v)$ and $\varphi(w)$ are conjugate with the aid of an idempotent matrix. □

The idea is to show that the idempotent matrix $b = \begin{pmatrix} 1 & t_b \\ 0 & 1 \end{pmatrix}$, where $t_b = \sum_{i=1}^{r} b_i t_i$, with the aid of which $\varphi(v)$ and $\varphi(w)$ are conjugate, is in the image of φ. By Theorem 5.4.7 this is the case if and only if $\sum\limits_{i=1}^{r} b_i(\mu(x_i) - 1) = 0$ in $\mathbb{Z}(F/N)$.

Now, since $b^{-1}\varphi(v)b = \varphi(w)$,

$$\begin{aligned}
\begin{pmatrix} \mu(w) & t_w \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} 1 & t_b \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mu(v) & t_v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & t_b \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & -t_b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu(v) & t_v \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & t_b \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \mu(v) & t_v - t_b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & t_b \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} \mu(v) & t_v + \big(\mu(v) - 1\big)t_b \\ 0 & 1 \end{pmatrix}.
\end{aligned}$$

It follows that $\mu(v) = \mu(w)$ and that $t_v + \big(\mu(v) - 1\big)t_b = t_w$. Since $\mu(v) = \mu(w)$, then $wv^{-1} \in N$, so there exists $z \in N$ such that $wv^{-1} = z$. Differentiating both sides of this equation with respect to any generator $x \in X$ gives

$$\frac{\partial z}{\partial x} = \frac{\partial\big(wv^{-1}\big)}{\partial x} = \frac{\partial w}{\partial x} + w\frac{\partial v^{-1}}{\partial x} = \frac{\partial w}{\partial x} - wv^{-1}\frac{\partial v}{\partial x}.$$

But $\mu(wv^{-1}) = 1$, so in $\mathbb{Z}(F/N)$, the above yields

$$\mu\left(\frac{\partial z}{\partial x}\right) = \mu\left(\frac{\partial w}{\partial x}\right) - \mu\left(\frac{\partial v}{\partial x}\right).$$

Now, turning our attention back to the equation $t_v + \big(\mu(v) - 1\big)t_b = t_w$, we get

$$\begin{aligned} & \sum_{i=1}^{r} \mu\left(\frac{\partial v}{\partial x_i}\right) t_i + \big(\mu(v) - 1\big) \sum_{i=1}^{r} b_i t_i = \sum_{i=1}^{r} \mu\left(\frac{\partial w}{\partial x_i}\right) t_i \\ \Leftrightarrow \quad & \big(\mu(v) - 1\big) \sum_{i=1}^{r} b_i t_i = \sum_{i=1}^{r} \left[\mu\left(\frac{\partial w}{\partial x_i}\right) - \mu\left(\frac{\partial v}{\partial x_i}\right) \right] t_i \\ \Leftrightarrow \quad & \big(\mu(v) - 1\big) \sum_{i=1}^{r} b_i \big(\mu(x_i) - 1\big) = \sum_{i=1}^{r} \frac{\partial z}{\partial x_i} \big(\mu(x_i) - 1\big). \end{aligned}$$

By the Fundamental Theorem of Calculus (Theorem 5.3.4), $\sum_{i=1}^{r} \frac{\partial z}{\partial x_i}\big(\mu(x_i) - 1\big) = \mu(z) - 1$. But $z \in N$, so $\mu(z) - 1 = 0$ in $\mathbb{Z}(F/N)$. Thus,

$$\big(\mu(v) - 1\big) \sum_{i=1}^{r} b_i \big(\mu(x_i) - 1\big) = 0 \text{ in } \mathbb{Z}(F/N).$$

Since $\mu(v) \neq 1$ and $\mathbb{Z}(F/N)$ has no zero divisors (since F/N is torsion free), it must be the case that

$$\sum_{i=1}^{r} b_i \big(\mu(x_i) - 1\big) = 0 \text{ in } \mathbb{Z}(F/N).$$

Thus, $b \in \operatorname{Im}(\varphi)$. Finally, this means that there exists $s \in F/N'$ such that $\varphi(s) = b$ and we get $\varphi(s)^{-1}\varphi(v)\varphi(s) = \varphi(w)$. Since φ is an embedding, $s^{-1}vs = w$ in F/N', so v and w are conjugate in F/N'.

□

With this in mind, we are ready to apply the result from Section 6.1 to a wreath product of free solvable groups.

Theorem 6.2.3. *The Conjugacy Problem in the free solvable group $F/F^{(d)}$ is in $O\big(rd|w|^6\big)$.*

Proof. The proof is by induction on the degree of solvability, d. The base case is the abelian group F/F', where the Conjugacy Problem is linear. Now suppose there is an algorithm, which solves the Conjugacy Problem in $F/F^{(d-1)}$ in $O\big(r(d-1)|w|^6\big)$. By Proposition 6.2.2, one can reduce the Conjugacy Problem in $F/F^{(d)}$ to the Conjugacy Problem in $M(F/F^{(d-1)})$. Since $M(F/F^{(d)}) \cong F/F' \mathrm{wr} F/F^{(d)}$ and F/F' is abelian, apply Theorem 6.1.4. In order to do this we need to compute the images of the input elements under the Magnus embedding and then find polynomial bounds for the Conjugacy Problems of F/F', the Word Problem in $F/F^{(d-1)}$ and the Power Problem in $F/F^{(d-1)}$.

The Conjugacy Problem in F/F' is the same as the Word Problem and so it is in $O(|w|)$. By the induction hypothesis, there is an algorithm, which solves the Conjugacy Problem in $F/F^{(d-1)}$ in $O\big(r(d-1)|w|^6\big)$. By Theorem 6.2.1 there is an algorithm which solves the Power Problem in $F/F^{(d)}$ in $O(dr|w|^3)$ and by Corollary 5.5.11 there is an algorithm to decide the Word Problem in $F/F^{(d-1)}$ in time $O(r(d-1)|w|^3)$. Then from Theorem 6.1.4, the complexity of the Conjugacy Problem in $F/F^{(d)} = F/F' \mathrm{wr} F/F^{(d-1)}$ is

$$O\big(|w|^2 r(d-1)|w|^6 + |w| r(d-1)|w|^3 + |w|^3 r(d-1)|w|^3\big).$$

It is easily seen now that the complexity of the Conjugacy Problem in free solvable groups is

$$O\big(rd|w|^6\big).$$

□

CHAPTER 7
Conclusion

This thesis consisted of a survey of the decidability and computability of the Word and Conjugacy Problems in various classes of groups, and more specifically in solvable groups. It focused on the innovative approach of Myasnikov, Roman'kov, Ushakov and Vershik for computing Fox derivatives efficiently. Their methods proved very useful in constructing a polynomial time algorithm to solve the Conjugacy Problem in free solvable groups. An important property of this algorithm is that its complexity is a polynomial whose degree does not depend on the degree of solvability d. More precisely, one may consider the free solvable group itself, with rank r and degree d, as part of the input along with the word w and the algorithm runs in time $O(rd|w|^6)$. Here the complexity is computed with respect to the RAM model of computation. If we consider the same algorithm within the Turing machine framework, the complexity will be $O(rd|w|^6 \log |w|)$. The extra logarithmic factor appears due to the fact that in a Turing model of computation, adding two integers requires time $O(\log n)$, where n is the length of the integers encoded in binary. In RAM, addition takes constant time. Similarly, other results have slightly different complexities when considered from the Turing point of view. Thus, the Conjugacy Problem in free groups and in word-hyperbolic groups is decidable in time $O(n \log n)$, where n is the length of the two input words (and is linear in the RAM model). The algorithm presented for deciding the Word Problem in free solvable groups, as well as the one for computing Fox derivatives and the Magnus embedding, has complexity $O(rdn^3 \log n)$, where n is the length of the input. In all of these cases, there is a minor change in the exact complexity, but the algorithms still run in polynomial time.

It is worth noting that the algorithm to decide the Conjugacy Problem in free solvable groups is almost constructive. This suggests that an efficient solution to the Conjugacy Search Problem, which asks to provide a conjugator for the two input words, is at hand. Indeed, given two words $x, y \in F/F^d$ the computation of the Magnus embedding $\varphi(x)$ and $\varphi(y)$ using Fox derivatives is constructive, the solution to the Power Problem as described in this thesis produces a power n such that $x^n = y$ and the conjugacy criterion for wreath products given by Matthews exhibits specific functions as conjugators.

The result on wreath products is much more general and can potentially be applied to deduce the polynomial-time decidability of the Conjugacy Problem in many classes of groups. Indeed, for a group G, suppose one has an embedding $\psi : G \hookrightarrow A \mathrm{wr} B$ for some wreath product $A \mathrm{wr} B$. If x and y are two words in G, the fact that $\psi(x)$ and $\psi(y)$ are conjugate in $A \mathrm{wr} B$ is not enough to deduce that x and y are conjugate in G, since the conjugator we find in $A \mathrm{wr} B$ might not have a preimage in G. There are two approaches to deal with this problem. The first consists of finding other conjugators until one of them is in G. More precisely, if x and y are conjugated in G, there is a conjugator $d \in A \mathrm{wr} B$, such that $dC_G(x) \cap G \neq \varnothing$. This suggests that being able to find centralisers and deciding the subgroup intersection problem in polynomial time for a given group enables us to decide the Conjugacy Problem there in polynomial time.

Another approach consists of finding a ‘good’ embedding. If the embedding ψ is *conjugacy-preserving*, i.e., if any two words are conjugate in $\psi(G)$ if and only if they are conjugate in $A\mathrm{wr}B$, then the decidability of the Conjugacy Problem in G will follow directly from the decidability of the Conjugacy Problem in $A\mathrm{wr}B$. The Magnus embedding has this property, which enables us to decide the Conjugacy Problem in free solvable groups. Unfortunately, of the groups which are known to embed in wreath products, very few have embeddings which are known to be conjugacy-preserving. Hence a possible goal, emerging from this new result, is to look for groups with appropriate embeddings and deduce new results from the polynomial-time decidability of the Conjugacy Problem in wreath products.

We hope that this thesis will serve as basis for many more interesting results in this direction.

References

[1] E. Artin. Theorie der zopfe. *Abh. Mth. Sem. Hamburg*, 4:47–72, 1925.

[2] M. R. Bridson and A. Haefliger. *Metric spaces of non-positive curnature.* Springer-Verlag, 1999.

[3] I. Bumagin. The conjugacy problem for relatively hyperbolic groups. *Algebraic and Geometric Topology*, 4:1013 – 1040, 2004.

[4] D. Epstein and D. Holt. The linearity of the conjugacy problem in word-hyperbolic groups. *Internat. J. Algebra Comput.*, 16:287–305, 2006.

[5] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word processing in groups.* Jones and Bartlett Publishers, 1992.

[6] S. Gersten and H. Short. Small cancellation theory and automatic groups II. *Invent. Math.*, 105:641–662, 1991.

[7] R. I. Grigorchuk. An example of a finitely presented amenable group not belonging to the class EG. *Sbornik Math*, 189:75–95, 1998.

[8] M. Gromov. Hyperbolic groups. *Math. Sci. Res. Inst. Publ.*, 8:75–263, 1987.

[9] J. E. Hopcroft and J. D. Ulman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.

[10] O. Kharlampovich. A finitely presented solvable group with unsolvable word problem. *Izvest. Ak. Nauk, Ser. Mat.*, 45(4):852–873, 1981.

[11] J. C. Lennox and D. J. S. Robinson. *The Theory of Infinite Soluble Groups.* Oxford Science Publications, 2004.

[12] Yu. G. Leonov. The conjugacy problem in a class of 2-groups. *Mat. Zametki*, 64:573–583, 1998.

[13] J. Marshall. Computation problems in hyperbolic groups. *Internat. J. Algebra. Comput.*, 15(1):1 – 13, 2005.

[14] J. Matthews. The conjugacy problem in wreath products and free metabelian groups. *T. Am. Math. Soc.*, 121:329–339, 1966. English transl., Soviet Math. Dokl. **8** (1967), 555–557.

[15] C. F. Miller III. *On group-theoretic decision problems and their classification*, volume 68 of *Annals of Mathematics Studies.* Princeton University Press, 1971.

[16] A. Myasnikov, V. Roman'kov, A. Ushakov, and A. Vershik. The word and geodesic problems in free solvable groups, 2008.

[17] V. N. Remeslennikov and V. G. Sokolov. Certain properties of Magnus embedding. *Algebra i Logika*, 9(5):566–578, 1970.

[18] N. N. Repin. Equations with one unknown in nilpotent groups. *(English translation) Math Notes*, 34(1–2):582–585, 1983.

[19] N. N. Repin. Solvability of equations with one indeterminate in nilpotent groups. *(Russian) Izv. Akad. Nauk SSSR Ser. Mat.*, 48(6):1295–1313, 1984.

[20] V. A. Roman'kov. On equations in free metabelian groups. *Siberian Math. J*, 20(3):671–673, 1979.

[21] V. A. Roman'kov. Universal theory of nilpotent groups. *(Russian) Mat. Zametki*, 25(4):487–495, 1979.