# Free Groups and Stallings' folding


Dario Ascari

Lady Margaret Hall

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

Trinity 2023

# Acknowledgements

Above all, I would like to express my deepest gratitude to my supervisor Martin Bridson for his invaluable expertise, guidance, and support throughout the completion of this thesis. His insightful and constructive feedback at every stage of the research process helped me to refine my ideas and sharpen my arguments.

I am also grateful to Ric Wade for his encouragement, and for insightful conversations that were helpful at many stages of my PhD journey. I wish to thank Jonathan Fruchter for all the fruitful discussions about both maths and non-maths, and Chris Weis for his support and helpful suggestions. My heartfelt thanks go to my collaborator Francesco Milizia for all the interesting discussions and for his patience throughout our collaboration.

I am grateful to all my colleagues and friends who supported me along the way, and to my family. Their unwavering support and encouragement have been a constant source of inspiration to me. In particular I want to thank my brother Flavio, whose presence in my life has been constant and extremely important, and my dearest friend Giovanni Gozzi.

Thank you all for being part of my journey!

# Abstract

In the present work we investigate various aspects of free groups and their automorphisms, utilizing Stallings' folding techniques introduced in 1983. These techniques have provided new insights into several classical results and have been widely applied over the years to study various properties of free groups and to generalize results to wider families of groups.

We apply folding techniques to uncover a fine property of the classical Whitehead's algorithm for recognizing primitive elements and free factors in a free group. This property is then used to the study of the complex of free factors, a crucial object on which $\mathrm{Out}(F_n)$ acts. We also make use of these techniques to investigate echelon subgroups of $F_n$, a particular type of subgroup which is relevant to the study of fixed-point subgroups of automorphisms of $F_n$.

Next we turn our attention to the problem of studying equations in a free group. Given two free groups $H \leq F$ and an element $g \in F$, we study the ideal of the equations $w(x)$ in $H * \langle x \rangle$ with $g$ as a solution. In particular, we focus on the relationship between equations and their degree, providing an algorithm that determines the minimum integer $d$ that appears as the degree of an equation in a given ideal. More generally, we study the subset of the equation in a given ideal of a fixed degree $d$. This is done using two different techniques, one based on Stallings' folding operations and the other using context-free languages.

# Contents

# INTRODUCTION

One of the most natural and important families of groups to study is given by *free groups*, i.e. groups with a set of generators without relations. Free groups were introduced at the end of 19th century, and the study of their structure turned out to be an extremely rich area. Since we can think of free groups as fundamental groups of graphs, a fundamental tool in the study of free groups is the operation of *folding* for graphs, introduced by J. R. Stallings in his seminal paper [Sta83] in 1983. By introducing this operation, Stallings was able to provide a new and valuable insight into some of the classical results about free groups, such as the subgroup membership problem, Marshall Hall's theorem, and Howson's Theorem. Since then, the idea of folding has been used in a fruitful way in the study of several different aspects of free groups and of their automorphisms. Generalizations of the same idea also facilitated the extension of some of the results to larger families of groups (e.g. graphs of groups and right-angled Artin groups). In this work we show several new results about finitely generated free groups, obtained using Stallings' graphs.

Our first family of results is related to primitive elements in a finitely generated free group $F_n$: we extract a fine property of the classical Whitehead's algorithm, and we make use of this property to provide new results about the structure of free factors in $F_n$. We use these techniques to improve the known results about computation of distances in the free factor complex of $F_n$. We also provide an algorithm to determine whether a subgroup of $F_n$ admits an echelon basis, allowing one to put the subgroup in a particularly convenient form (which is related to the notion of inertia and to the study of fixed-point subgroups of automorphisms of $F_n$).

Our second family of results is related to equations in a finitely generated free group $F_n$: we study the set $\mathfrak{I}_g$ of equations $w(x)$ with coefficients in a fixed subgroup $H \leq F_n$ and with a given solution $g \in F_n$ (i.e. the "ideal" of the element $g$ with coefficients in the subgroup $H$). This is done with a particular focus on the degree of the equations $w(x)$, i.e. the number of $x$ appearing in the cyclic reduction of $w(x)$. We provide an algorithm to produce an equation in the ideal $\mathfrak{I}_g$ with minimum possible degree. We provide algorithms to determine which numbers $d$ can appear as degrees of equations $w(x)$ in $\mathfrak{I}_g$, and we also study the asymptotic growth of the number of equations $w(x)$ in $\mathfrak{I}_g$ of a given fixed degree. This is done using two different approaches: the former interpreting equations as paths in a suitable Stallings' graph, and controlling the cancellation that can happen along these paths;

the latter using context-free languages, a tool which is widely used in computer science. This second approach enables us to prove polynomial bounds on the running time of the algorithms.

The first two chapters are expository and the remaining chapters contain original results.

## Chapter one

In Chapter 1 we set up the notation and we review several well-known results. We introduce the Stallings' folding operation and we describe some of its most important applications in the theory of finitely generated free groups. We explain the solution to the subgroup membership problem, we prove Marshall Hall's theorem, and we provide an algorithm to compute the intersection of two finitely generated subgroups. We define the notion of rank-preserving folding operation, and we use it to provide an algorithm to compute the kernel of a homomorphism between free groups. We introduce the notion of Whitehead automorphism, and we use Stallings graphs to show that the group of automorphisms of a free group is finitely generated.

## Chapter two

In Chapter 2 we give an overview about the literature on Whitehead's algorithm. The algorithm was discovered by Whitehead in 1936 and allows one to determine whether an element of a finitely generated free group $F_n$ is primitive, i.e. if it's part of some basis for the group $F_n$. This is strictly related to the study of the group $\mathrm{Aut}(F_n)$, making Whitehead's algorithm one of the cornerstones in the study of automorphisms of free groups. We explain Whitehead's algorithm and we prove Whitehead's Theorem (here $|\cdot|_{\mathrm{cyc}}$ denotes the cyclic length of a word in a fixed basis):

**Theorem 1** (Whitehead, [Whi36a])**.** *Let $w$ be a cyclically reduced primitive word of length greater than one. Then there is a Whitehead automorphism $\varphi$ such that $|\varphi(w)|_{\mathrm{cyc}} < |w|_{\mathrm{cyc}}$.*

In Section 2.2 we provide a first proof in the spirit of Whitehead's original paper. In Section 2.3 we sketch another independent proof of the algorithm, based on the peak-reduction techniques introduced by Rapaport in [Rap58]. This technique allows one to obtain several improvements of the algorithm, leading among other things to efficient finite presentations of $\mathrm{Aut}(F_n)$, and to an algorithm for the recognition of free factors (see [Ger84]). In Section 2.4 we provide a third proof of Whitehead's algorithm, found by H. Wilton, see Lemma 2.10 of [Wil18], and independently by M. Heusener and R. Weidmann, see [HW19]. This third argument is remarkably short and completely based on folding techniques; it will be of fundamental importance in the subsequent Chapter 3.

Finally, in Section 2.5 we provide another algorithm to recognize primitive elements, completely different from Whitehead's algorithm in nature, that has been found more recently by M. Bestvina and M. Bridson and independently by D. Puder, see [Pud13].

## Chapter three

Chapter 3 is devoted to our new results related to Whitehead's algorithm; it is mostly based on [Asc21].

We build on a refinement of Whitehead's Theorem 1, introducing a fine property about the pattern of cancellations that occur inside the words while running Whitehead's algorithm. Recall that, given a finitely generated free group $F_n$ with basis $a_1, ..., a_n$, a Whitehead automorphism is a map $\varphi : F_n \to F_n$ such that, for some $a \in \{a_1, ..., a_n, a_1^{-1}, ..., a_n^{-1}\}$, we have $\varphi(a) = a$ and $\varphi(a_i) \in \{a_i, aa_i, a_i a^{-1}, aa_i a^{-1}\}$ for $a_i \neq a, a^{-1}$. A generic element $w \in F_n$ consists of a finite (reduced) sequence of symbols from $\{a_1, ..., a_n, a_1^{-1}, ..., a_n^{-1}\}$, and in order to obtain the image $\varphi(w)$ we can just apply $\varphi$ letter by letter to the sequence of symbols, and then reduce the resulting word.

**Theorem A** (Fine Property, Theorem 3.1.2)**.** *The automorphism in Theorem 1 can be chosen in such a way that every $a$ or $a^{-1}$ letter, which is added when we apply $\varphi$ to $w$ letter by letter, immediately cancels (in the cyclic reduction process).*

This refinement can be applied directly to elucidate novel features of primitive elements and free factors, such as the following theorem:

**Theorem B** (Theorem 3.1.3)**.** *Let $H \leq F_n$ be a finitely generated subgroup. Suppose that every element which is primitive in $H$ is also primitive in $F_n$. Then $H$ is a free factor.*

We also generalize the fine property of Theorem A to subgroups. Subgroups $H \leq F_n$ are represented by labelled graphs $\mathrm{core}(H)$. The standard statement is the following Theorem 2 and we add the fine property of Theorem C.

**Theorem 2** (Gersten, [Ger84])**.** *Let $H \leq F_n$ be a free factor and suppose* $\mathrm{core}(H)$ *has more than one vertex. Then there is a Whitehead automorphism $\varphi$ such that* $\mathrm{core}(\varphi(H))$ *has strictly fewer edges than* $\mathrm{core}(H)$*.*

**Theorem C** (Theorems 3.2.7 and 3.2.9)**.** *The automorphism in Theorem 2 can be chosen in such a way that* $\mathrm{core}(\varphi(H))$ *can be obtained from* $\mathrm{core}(H)$ *by means of a quotient that collapses some of the edges to points whilst preserving the labels and orientations on the other edges.*

This additional property turns out to have several interesting features, and in particular it behaves well with respect to subgroups, see Lemmas 3.2.11 and 3.2.12.

It also allows us to deduce a relative version of Whitehead's algorithm, see Theorems 3.2.13 and 3.2.15. In Sections 3.3 and 3.4 we provide two other important applications of the additional property of Theorem C: the former about the study of the free factor complex of a free group, the latter about echelon subgroups of a free group.

**The free factor complex.** In order to better understand the structure of $\mathrm{Aut}(F_n)$, and in analogy with the case of surfaces, several spaces have been introduced; among these, of particular importance are Culler-Vogtmann's *Outer Space* $\mathrm{CV}_n$, which is analogous to the Teichmuller space of a surface, and the *Free Factor Complex* $\mathrm{FF}_n$, which is analogous to the curve complex of a surface.

Many properties of the curve complex hold in a similar way, with much harder proofs, for $\mathrm{FF}_n$. A famous rigidity theorem of Ivanov states that the isometries of the curve complex are essentially the mapping class group of the surface; in the same way, there is a rigidity theorem due to Bestvina and Bridson stating that the isometries of $\mathrm{FF}_n$ are essentially the outer automorphisms of $F_n$, see [BB22]. Just as the curve complex turned out to be hyperbolic, $\mathrm{FF}_n$ is hyperbolic too, as shown in [BF14]. However, while the structure of the curve complex of a surface is overall quite well-understood, many aspects of the free factor complex still remain mysterious; even proving that it is unbounded takes a considerable effort, and the following is a big open question:

**Question 3.** *Is there an algorithm that computes the distance between two vertices of* $\mathrm{FF}_n$*?*

We make use of the Theorems 2 and C to provide an algorithm that determines whether two vertices of $\mathrm{FF}_n$ are at distance $d$ for $d = 1, 2, 3$; we are also able to do that for $d = 4$ in the particular case when one of the two vertices represents a conjugacy class of free factors of rank $n - 1$. The case $d = 4$ in full generality remains still open.

**Echelon subgroups.** Fixed-point subgroups of automorphisms of a free group have been widely studied over the years. The introduction of train-track graph representatives, due to M. Bestvina and M. Handel in [BH92], opened the way to a vast amount of new results in this direction, including a proof to the Scott's conjecture, stating that $\mathrm{rank}(\mathrm{Fix}(\varphi)) \leq n$ for every automorphism $\varphi$ of $F_n$. A full algorithm that, given an automorphism $\varphi$ of $F_n$, computes the subgroup $\mathrm{Fix}(\varphi)$ was found later, see [BM12]. However, the converse question remains open:

**Question 4.** *Is there an algorithm that, given a finitely generated subgroup* $H \leq F_n$*, tells us whether* $H$ *coincides with* $\mathrm{Fix}(\varphi)$ *for some automorphism* $\varphi : F_n \to F_n$*?*

In general, fixed-point subgroups behave in a very controlled manner with respect to intersections: for $\varphi \in \mathrm{Aut}(F_n)$, the subgroup $H = \mathrm{Fix}(\varphi)$ is always *inert*, i.e. it satisfies $\mathrm{rank}(H \cap K) \leq \mathrm{rank}(K)$ for every other subgroup $K \leq F_n$, see [DV96].

The notion of echelon subgroup was introduced recently by A. Rosenmann, in analogy with matrices in echelon form in linear algebra: a subgroup $H \leq F_n$ is

called *echelon* if there is a basis $b_1, ..., b_n$ for $F_n$ such that $\text{rank}(H \cap \langle b_1, ..., b_i \rangle) \leq \text{rank}(H \cap \langle b_1, ..., b_{i-1} \rangle) + 1$ for $i = 1, ..., n$. It follows from the work of A. Martino and E. Ventura, see [MV04], that all fixed-point subgroups of automorphisms can be put into echelon form. In [Ros13], Rosenmann shows that echelon subgroups are inert, providing an alternative proof of the fact that fixed-points subgroups of automorphisms are inert. In other words we have the chain of implications

$$\text{fixed-point subgroup of an automorphism} \Rightarrow \text{echelon} \Rightarrow \text{inert}$$

We use Theorems 2 and C to prove the following theorem, answering a question asked by Rosenmann:

**Theorem D** (Theorem 3.4.14)**.** *There is an algorithm that takes in input a finitely generated subgroup $H \leq F_n$ and tells us whether it is echelon or not. In case the answer is affirmative, it also computes a basis with respect to which $H$ is echelon.*

We also show by means of a counterexample that the intersection of two echelon subgroups isn't always echelon; this answers in the negative another question of Rosenmann [Ros13].

## Equations in free groups

The study of polynomial equations over fields and rings has played a huge role in mathematics over the centuries. In analogy with the standard algebraic geometry, a partial theory of equations in the non-commutative group theoretic setting has been developed, see for example [BMR99]. Given two groups $H \leq G$, an *equation* in the variables $x_1, ..., x_m$ with coefficients in $H$ is an element $w(x_1, ..., x_m) \in H * F(x_1, ..., x_m)$ where $F(x_1, ..., x_m)$ denotes the free group generated by $x_1, ..., x_m$; an $m$-tuple $(g_1, ..., g_m)$ of elements of $G$ is a *solution* to the equation if $w(g_1, ..., g_m) = 1$ in $G$. However, the theory of solving (systems of) equations over groups turned out to be much harder than the usual polynomial counterpart.

In the case when $H \leq G$ are both free groups, the first algorithm to solve equations was found by G. S. Makanin, see [Mak83]; although the original algorithm was completely about combinatorics of words, later works gave more abstract and geometric insights about it, showing strict relations with *limit groups* (groups which are fully residually free, or equivalently groups that have the same universal first-order theory of free groups), and leading to the construction of the *Makanin-Razborov diagrams*. This led to a rich and fruitful theory, see [Sel06] and [KM06].

More recently, the study of the dual problem was initiated: instead of fixing the equation and looking for the solutions, we fix the solution and we study the set of equations with that given solution. Let $H \leq G$ be finitely generated free groups and let $g \in G$. Define the **ideal** $\mathfrak{I}_g$ to be

$$\mathfrak{I}_g := \{w(x) \in H * F(x) : w(g) = 1 \text{ in } G\}$$

This is a normal subgroup of $H * \langle x \rangle$. Also define the *degree* of an equation $w(x) \in H * F(x)$ as the number of $x$ and $x^{-1}$ (not counted with sign) that appear in (the cyclic reduction of) $w(x)$.

A. Rosenmann and E. Ventura answered the question of determining whether the ideal $\mathfrak{I}_g$ is non-trivial: in [RV21] they proved that there is an algorithm that, given $H$ and $g$, produces elements $g_1, ..., g_k$ such that the ideal $\mathfrak{I}_g$ is non-trivial if and only if $g$ belongs to one of the double cosets $Hg_1H, ..., Hg_kH$. In the case where $\mathfrak{I}_g$ is non-trivial, we have that $\mathfrak{I}_g$ is finitely generated as a normal subgroup of $H * F(x)$, and it's fairly easy to explicitly compute a basis for it (see Theorem 1.3.4).

## Chapter four

In Chapter 4 we go deeper into the study of the ideal $\mathfrak{I}_g$, providing new results about its structure, with a particular focus on the degree of the equations in the ideal, following [Asc22b]. In [RV21] the authors ask whether it is possible to algorithmically compute a non-trivial equation of minimum possible degree in the ideal $\mathfrak{I}_g$; we give affirmative answer to this question.

**Theorem E** (Corollary 4.2.17)**.** *There is an algorithm that, given $H \leq F_n$ and $g$ such that $g$ depends on $H$, produces a non-trivial equation $w \in \mathfrak{I}_g$ of minimum possible degree.*

The proof of Theorem E has the notable feature of providing an explicit bound on the length of an equation of minimum possible degree. To be precise, if $d_{\min}$ is the minimum degree of a non-trivial equation in $\mathfrak{I}_g$, then there is an equation $w(x)$ of degree $d_{\min}$ and of length which is bounded by a polynomial in the length of the generators for $H$, in the length of $g$ and in $d_{\min}$; for the precise bound see Theorem 4.2.2. An explicit upper bound on $d_{\min}$ can be obtained by looking at any finite set of generators for the ideal $\mathfrak{I}_g$.

A completely analogous result holds for equations of any fixed degree $d$.

**Theorem F** (Corollary 4.3.11)**.** *There is an algorithm that, given $H \leq F_n$ finitely generated and $g \in F_n$ and an integer $d \geq 1$, tells us whether $\mathfrak{I}_g$ contains non-trivial equations of degree $d$, and, if so, produces an equation $w \in \mathfrak{I}_g$ of degree $d$.*

In the same way as above, we also prove that there is an equation of degree $d$ whose length is bounded by a polynomial in the length of the generators of $H$, the length of $g$, and the degree $d$, see Theorem 4.3.10.

We study also the set $D_g = \{d \in \mathbb{N} : \mathfrak{I}_g \text{ contains a non-trivial equation of degree } d\}$. We show that either $D_g = \mathbb{N} \setminus E$ or $D_g = 2\mathbb{N} \setminus E$ for some finite set $E$. We prove that $D_g$ can be algorithmically computed, as follows:

**Theorem G** (Theorem 4.3.17)**.** *Given $H \leq F_n$ finitely generated and $g \in F_n$ that depends on $H$, there is an algorithm that:*

*(a) Determines whether $D_g$ coincides with $\mathbb{N} \setminus E$ or with $2\mathbb{N} \setminus E$ for some finite set $E$.*

*(b) Computes the finite set $E$.*

## Chapter five

In Chapter 5 we improve some of the results of Chapter 4, by looking at the same problems from another perspective, making extensive use of context-free languages, following [Asc22a].

Context-free languages are subsets of a free monoid that can be generated in a certain way, and have been widely studied in computer science. A relation between such languages and group theory had already been exploited by D. E. Muller and P. E. Schupp in [MS83], where they proved that a group has context-free word problem if and only if the group is virtually free. One of the main features of context-free languages, is that several problems related to them can be solved algorithmically in polynomial time. We prove that $\mathfrak{I}_g$ is context-free as a subset of $H * \langle x \rangle$, yielding the following:

**Theorem H** (Theorems 5.2.2 and 5.4.22)**.** *There is a polynomial-time algorithm that tells us whether $\mathfrak{I}_g$ contains a non-trivial equation or not.*

We then turn our attention to the set $\mathfrak{J}_{g,d}$ of the equations in $\mathfrak{I}_g$ of a certain degree $d$. We prove that $\mathfrak{J}_{g,d}$ is context-free as a subset of $H * \langle x \rangle$; this allows one to improve the result of Theorem F as follows:

**Theorem I** (Theorems 5.2.4 and 5.4.23)**.** *There is a polynomial-time algorithm that tells us whether the set $\mathfrak{J}_{g,d}$ is empty or not and, if such exists, produces an element in $\mathfrak{J}_{g,d}$.*

We also study the growth of the number of equations in the sets $\mathfrak{I}_g$ and $\mathfrak{J}_{g,d}$. Define $\rho_g(M)$ to be the number of cyclically reduced words of length at most $M$ that represent an equation in $\mathfrak{I}_g$. Similarly, define $\rho_{g,d}(M)$ to be the number of cyclically reduced words of length at most $M$ that represent an equation in $\mathfrak{J}_{g,d}$. The growth rate of arbitrary context-free languages has already been studied, see [BG02] and [Inc01] and [GKRS08], allowing us to prove the following results:

**Theorem J** (Theorem 5.3.4)**.** *The function $\rho_g(M)$ has exponential growth.*

**Theorem K** (Theorems 5.3.5 and 5.4.27)**.** *Let $d \in \mathbb{N}$ be a non-negative integer. Then the function $\rho_{g,d}(M)$ either has exponential growth or else it is bounded above and below by polynomials of degree $k$ for some $k \in \mathbb{N}$. Moreover, there is a polynomial-time algorithm that tells us which case takes place, and in the second case it computes the integer $k$.*

We improve the result of Theorem G by showing that we can also compute a partition of $D_g$ based on the growth of the functions $\rho_{g,d}$. Consider the partition

$$D_g = D_g^{\exp} \sqcup \bigsqcup_{k \in \mathbb{N}} D_g^{\mathrm{pol},k}$$

where $D_g^{\mathrm{pol},k} := \{d \in D_g : \rho_{g,d} \text{ has polynomial growth of degree } k\}$ and $D_g^{\exp} = \{d \in D_g : \rho_{g,d} \text{ has exponential growth}\}$.

**Theorem L** (Theorem 5.3.9)**.** *Suppose $H$ has rank at least 2. Then there is an algorithm that computes the finite sets $D_g^{\mathrm{pol},k}$ for $k \in \mathbb{N}$; all but finitely many of these are empty.*

Theorem E provides us with an algorithm that, given $H$ and $g$, computes the value of $d_{\min}$. We improve this result by providing a polynomial-time algorithm for the same purpose, see Theorem 5.4.24. It is natural to wonder about how big $d_{\min}$ can be: we provide an explicit upper bound for $d_{\min}$, which is polynomial in the length $L$ of the generators for $H$ and in the length of $g$, but (at least) exponential in $\mathrm{rank}(H)$, see Theorem 5.4.26. We prove that this bound is "sharp", by providing counterexamples where the dependence of $d_{\min}$ on $\mathrm{rank}(H)$ is exponential.

The approach adopted in Chapter 5, based on context-free languages, seems to prove stronger results about equations than the approach of Chapter 4, based on Stallings folding. However, as we said before, Muller and Schupp proved that a group has context-free word problem if and only if the group is virtually free; as a consequence, context-free languages are intrinsically related to free groups, and thus it's hard to imagine how the same techniques could be used to generalize the same results to many other kinds of groups. Conversely, it seems likely that the techniques based on Stallings folding can be modified to work for other families of groups too.

# 1 | Free groups, graphs and Stallings folding

This chapter's aim is to set up the notation for the present work, and to review several classical results about free groups and Stallings' folding operations on graphs.

## 1.1 Free groups and graphs

In this section we set up the notation that we are going to use throughout the rest of the thesis. We begin by reviewing some of the most basic definitions and lemmas about graphs and their fundamental groups.

### 1.1.1 Graphs

**Definition 1.1.1.** *A **graph** $G$ is a non-empty 1-dimensional CW complex.*

We allow for multiple edges between the same pair of vertices, and we allow for edges from a vertex to itself. For a graph $G$ we denote with $V = V(G)$ the 0-skeleton of $G$, and each point of $V$ is called *vertex*; each connected component of $G \setminus V$ is called an *open edge* and its closure is called an *edge*; we denote with $E(G)$ the set of edges of a graph $G$. Sometimes we will consider graphs with a basepoint; in these cases, we always mean the basepoint to be a vertex.

**Definition 1.1.2.** *A **combinatorial map** $f : G \to G'$ between graphs is a continuous map which sends each vertex of $G$ to a vertex of $G'$, and each open edge of $G$ homeomorphically onto an open edge of $G'$.*

For $l \geq 1$ we define $I_l$ to be the graph obtained from a subdivision of the unit interval $[0, 1]$ into $l$ arcs, see Figure 1.1. More precisely, $I_l$ has $l + 1$ vertices at $\frac{i}{l}$ for $i = 0, ..., l$, and $l$ edges given by closed intervals. A **combinatorial path** in a graph $G$ is a combinatorial map $\sigma : I_l \to G$ for some $l \geq 1$.

For $l \geq 1$ we define $C_l$ to be the graph obtained from a subdivision of the unit circle $\{(x, y) : x^2 + y^2 = 1\} \subseteq \mathbb{R}^2$ into $l$ arcs, see Figure 1.1. More precisely $C_l$ has $l$ vertices at the points $(\cos(\frac{2\pi i}{l}), \sin(\frac{2\pi i}{l}))$ for $i = 0, ..., l - 1$, and $l$ edges given by

closed arcs on the unit circle. A **combinatorial loop** in a graph $G$ is a combinatorial map $\sigma : C_l \to G$ for some $l \geq 1$.

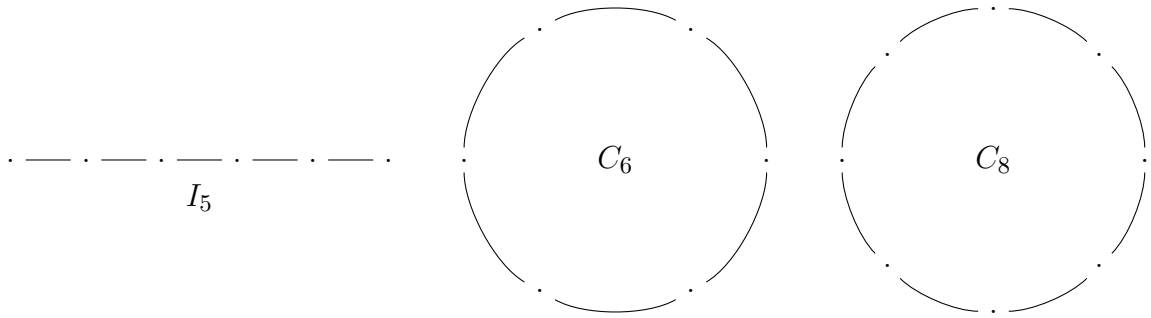


Figure 1.1: The graphs $I_5, C_6$ and $C_8$.

We say that a combinatorial path (resp. loop) is **reduced** if it is locally injective. The local injectivity has to be checked only at the vertices of $I_l$ (resp. $C_l$): in the interior of the edges, every combinatorial path (resp. loop) is locally injective by definition. We say that a combinatorial path $\sigma : I_l \to G$ with $\sigma(0) = \sigma(1)$ is **cyclically reduced** if it is reduced when seen as a combinatorial loop.

**Proposition 1.1.3** (Reducing paths)**.** *We have the following:*
*(i) Every non-trivial homotopy class of paths $\sigma : [0,1] \to G$ (relative to the endpoints) contains a unique reduced path.*
*(ii) The homotopy class of a constant path $\sigma : [0,1] \to G$ (relative to the endpoints) contains no reduced path.*

*Proof.* (i) It's easy to see that every non-trivial homotopy class of paths contains a combinatorial path; consider a combinatorial path $\overline{\sigma} : I_l \to G$ with $l \geq 1$ minimum possible. Suppose $\overline{\sigma}$ isn't reduced: then at some point it crosses an edge and then the same edge again in opposite direction; we can then change $\overline{\sigma}$ by means of a homotopy, removing the two crossing of that edge. This either defines a combinatorial path $I_{l-2} \to G$ homotopic to $\overline{\sigma}$, contradicting the minimality of $l$, or shows that $\overline{\sigma}$ is homotopic to a constant path, contradiction since the homotopy class was non-trivial.

Suppose we are given $\sigma_1 : I_{l_1} \to G$ and $\sigma_2 : I_{l_2} \to G$ reduced paths that are distinct but homotopic (relative to the endpoints). Take such $\sigma_1, \sigma_2$ with $l_1 + l_2$ minimum possible, and notice that this implies that $\sigma_1$ and $\sigma_2$ are injective and disjoint, except at the endpoints. In particular, by taking the concatenation of $\sigma_1$ with the reverse of $\sigma_2$, we obtain an injective combinatorial loop $\tau : C_l \to G$ where $l = l_1 + l_2$.

We are now going to build a non-trivial degree two covering space of $G$. Let $H = \overline{G \setminus \mathrm{im}(\tau)}$ be a subgraph of $G$ and let $H_1, H_2$ be two isomorphic copies of $H$. Consider the degree two covering map $p : I_{2l} \to \mathrm{im}(\tau)$, and for each vertex $v$ of $H \cap \mathrm{im}(\tau)$ consider the two corresponding vertices $v_1, v_2$ of $H_1, H_2$ respectively, and identify them with the two points of $p^{-1}(v)$; call $\widetilde{G}$ the graph obtained from $I_{2l} \cup H_1 \cup H_2$ by

means of these identifications. The degree-2 covering map $p : I_{2l} \to \mathrm{im}(\tau)$ extends to a degree-2 covering map $\widetilde{p} : \widetilde{G} \to G$.

We notice that the two paths $\sigma_1, \sigma_2$ can be lifted to $\widetilde{G}$ with the same starting point, but with different endpoints. This shows that $\sigma_1, \sigma_2$ aren't homotopic, contradiction.

(ii) A reduced path which is homotopic to a constant path can be seen as a reduced loop. As above, we can consider such a reduced loop of minimum possible lenght, and this implies that the loop is injective. We now construct a degree-2 covering space as above, and this shows that the loop can't be homotopically trivial since it doesn't lift, contradiction. □

**Definition 1.1.4.** *For a path $\sigma : [0,1] \to G$ we define its **reduction** $\overline{\sigma} : [0,1] \to G$ as the unique reduced path homotopic to $\sigma$ relative to its endpoints, or as the constant path if $\sigma$ is homotopically trivial.*

An analogue of Proposition 1.1.3 can be proved for loops, in the exact same way. We can also define the **reduction** of a loop as the unique (up to rotation) reduced loop in its homotopy class, or as the trivial loop if the homotopy class is trivial.

### 1.1.2 Trees

**Proposition 1.1.5.** *Let $G$ be a graph. Then the following are equivalent:*

*(i) $G$ is connected and for every open edge $e$ the graph $G \setminus e$ is disconnected.*

*(ii) $G$ is connected and contains no reduced loop.*

*(iii) For every two distinct vertices, there is a unique reduced path from one to the other.*

*(iv) $G$ is connected and has trivial fundamental group.*

*(v) $G$ is contractible.*

*Proof.* The equivalence of (ii), (iii), (iv) follows from Proposition 1.1.3.

(i) $\Rightarrow$ (ii) If $G$ contains a reduced loop, then it contains one of minimum length, which has thus to be injective. We can then remove any open edge of the reduced loop, and the graph remains connected, contradiction.

(ii) $\Rightarrow$ (i) Let $e$ be any open edge of $G$, with endpoints $v, w$. Notice that $v \neq w$, as otherwise $e$ would be a reduced loop in $G$. If $G \setminus e$ is connected, then by Proposition 1.1.3 we find a reduced path connecting $v$ with $w$, and adding $e$ to this path gives a reduced loop in $G$, contradiction.

(v) $\Rightarrow$ (iv) Trivial.

(iii) $\Rightarrow$ (v) We choose a vertex $*$ of $G$ and we want to prove that $f : G \to G$ given by $f \equiv *$ is homotopic to the identity. For each vertex $v$ of $G$, consider the unique reduced path $\sigma_v : [0,1] \to G$ with $\sigma_v(0) = *$ and $\sigma_v(1) = v$. On the subset $\mathrm{im}(\sigma_v) \times [0,1]$ of the space $G \times [0,1]$, define the map $F_v : \mathrm{im}(\sigma_v) \times [0,1] \to G$ given

by $F_v(\sigma_v(t), s) = \sigma_v(ts)$. We notice that for two distinct vertices $v, w$ of $G$ the two maps $F_v, F_w$ coincide on the intersection of their domains. We can thus define a map $F : G \times [0, 1] \to G$ given by the union of all the maps $F_v$ for $v$ vertex of $G$. This gives the desired homotopy. □

**Definition 1.1.6.** *A graph $G$ is called **tree** if it satisfies any of the equivalent conditions of Proposition 1.1.5.*

**Definition 1.1.7.** *Let $G$ be a connected graph. A **maximal tree** is a subgraph which is a tree and is maximal with respect to inclusion.*

It follows from Zorn's Lemma that every connected graph contains at least one maximal tree. More generally, every subgraph which is a tree can be extended to a maximal tree. Notice also that every vertex of $G$ has to be contained in every maximal tree.

### 1.1.3 Free groups

Let $\mathcal{A}$ be a set, whose elements have to be thought of as letters.

**Definition 1.1.8.** *Define the **free group** $F(\mathcal{A})$ as the set of all words in the letters of $\mathcal{A}$ and in their formal inverses, considered up to cancellation. The set $F(\mathcal{A})$ is a group with the operation given by concatenation of words.*

For $a \in \mathcal{A}$ we denote with $\overline{a} = a^{-1}$ the formal inverse of $a$. By "cancellation" we mean that, if a word contains an occurrence of $a$ adjacent to an occurrence of $\overline{a}$, then we are allowed to remove both those occurrences from the word; this means that we are taking the quotient under the equivalence relation generated by these cancellations. An element of $F(\mathcal{A})$ is an equivalence class of words, but with an abuse of notation we will denote in the same way an equivalence class $w \in F(\mathcal{A})$ and a word $w$ belonging to that class.

A word is called **reduced** if no cancellation is possible inside it. Every equivalence class of words in $F(\mathcal{A})$ contains exactly one reduced word, which is called the **reduction** of all the other words in the equivalence class. For an equivalence class $w \in F(\mathcal{A})$ we define its **length** $|w|$ as the number of letters in the unique reduced word in that class.

For a word $w \in F(\mathcal{A})$ we can allow for the first and the last letter of $w$ to cancel against each other, as if they were adjacent. We can reduce the word as much as possible using standard cancellations and this extra cancellation move: the resulting word $w'$ is independent, up to cyclic permutation, on the order in which we perform the cancellations, and is called the **cyclic reduction** of $w$. We define the **cyclic length** $|w|_{\text{cyc}} = |w'|$ and we say that a word is **cyclically reduced** if it coincides with its cyclic reduction.

Free groups enjoy the following well-known universal property: for every map $f : \mathcal{A} \to G$ from the set $\mathcal{A}$ to a group $G$ there is a unique homomorphism $\widehat{f} : F(\mathcal{A}) \to G$ extending $f$.

Free groups can also be defined as fundamental groups of one-vertex graphs, as we now explain. Let $R$ be a graph with one vertex $*$ and with set of edges $E = E(R)$; we will sometimes refer to such a graph as a **rose**. Suppose also for each $e \in E$ we are given an orientation, and let $\sigma_e : [0, 1] \to R$ be the reduced path that goes once around the edge $e$, according to the orientation. Then we can consider the free group $F(E)$ together with the unique homomorphism $f : F(E) \to \pi_1(R, *)$ that sends the edge $e$ to the homotopy class $[\sigma_e]$.

**Proposition 1.1.9.** *The homomorphism $f : F(E) \to \pi_1(R, *)$ is an isomorphism.*

*Sketch of proof.* When $E$ is finite this follows by induction from Van Kampen's Theorem. Suppose $E$ is infinite. Surjectivity can be proved using the fact that each path has compact image, and in particular it can only cover completely a finite number of edges. Injectivity can be proved with a similar argument, using that each homotopy has compact image, and thus can only cover completely a finite number of edges. □

When talking about free groups, it's fundamental the notion of basis. Let $F = F(\mathcal{A})$ be a free group.

**Definition 1.1.10.** *A subset $S \subseteq F$ is called **basis** if the set map $f : S \to F$ given by inclusion extends to an isomorphism $\widetilde{f} : F(S) \to F$.*

Of course $\mathcal{A}$ is a basis, but it's not the only one in general. It is relevant to notice that all the bases for a free group have the same cardinality, as we now show.

**Proposition 1.1.11.** *Let $\mathcal{A}, \mathcal{B}$ be sets. If $F(\mathcal{A}) \cong F(\mathcal{B})$ then $|\mathcal{A}| = |\mathcal{B}|$.*

*Proof.* If $\mathcal{A}$ is uncountable then $F(\mathcal{A})$ has the same cardinality as $\mathcal{A}$. If $\mathcal{A}$ is countable or finite, then $F(\mathcal{A})$ is countable. This shows that $\mathcal{A}$ is uncountably infinite if and only if $\mathcal{B}$ is, and in that case they have the same cardinality.

Suppose now that $\mathcal{A}, \mathcal{B}$ are countable or finite. The abelianization of $F(\mathcal{A})$ is $\mathbb{Z}^{\oplus\mathcal{A}}$ and we also observe that $\mathbb{Z}^{\oplus\mathcal{A}}/2\mathbb{Z}^{\oplus\mathcal{A}} \cong (\mathbb{Z}/2\mathbb{Z})^{\oplus\mathcal{A}}$. If $F(\mathcal{A}) \cong F(\mathcal{B})$ then $\mathbb{Z}^{\oplus\mathcal{A}} \cong \mathbb{Z}^{\oplus\mathcal{B}}$ and thus $\mathbb{Z}^{\oplus\mathcal{A}}/2\mathbb{Z}^{\oplus\mathcal{A}} \cong \mathbb{Z}^{\oplus\mathcal{B}}/2\mathbb{Z}^{\oplus\mathcal{B}}$, yielding that $(\mathbb{Z}/2\mathbb{Z})^{\oplus\mathcal{A}} \cong (\mathbb{Z}/2\mathbb{Z})^{\oplus\mathcal{B}}$ and in particular they must have the same cardinality. This shows that $\mathcal{A}$ is finite if and only if $\mathcal{B}$ is finite, and in that case we have $2^{|\mathcal{A}|} = 2^{|\mathcal{B}|}$ and thus $|\mathcal{A}| = |\mathcal{B}|$. □

**Definition 1.1.12.** *For a free group $F$ we define the **rank** $\operatorname{rank}(F)$ as the cardinality of any basis for $F$.*

When $\mathcal{A} = \{a_1, ..., a_n\}$ is a finite set, we denote with $F_n = F(a_1, ..., a_n) = F(\mathcal{A})$ the free group generated by this finite set and we denote with $\overline{a}_1, ..., \overline{a}_n$ the inverses of the generators. Of course we have $\operatorname{rank}(F_n) = n$.

### 1.1.4 The fundamental group of a graph

The fundamental group of a connected graph is a free group, and we provide here an explicit method to compute a basis for such a free group; this construction will be needed later.

Let $G$ be a connected graph with a basepoint $*$, and let $T$ be a maximal tree; let $E$ be the set of edges which are not contained in $T$, and suppose we are given an orientation on each edge $e \in E$. For $e \in E$, there is a unique reduced path $\sigma_e$ in $G$ that starts at the basepoint, moves along $T$ to the initial vertex of $e$, crosses $e$ according to the orientation, and moves along $T$ from the final vertex of $e$ to the basepoint.

**Proposition 1.1.13.** *The fundamental group $\pi_1(G, *)$ is a free group with basis given by the homotopy classes of the paths $\sigma_e$ for $e \in E$.*

*Proof.* Let $q : G \to G/T$ be the map that collapses the tree $T$ to a single point: since $T$ is contractible, the map $q$ induces an isomorphism of fundamental groups $q_* : \pi_1(G, *) \to \pi_1(G/T, *)$. The space $G/T$ is a one-vertex graph with edges corresponding to edges $e \in E$, and the path $q \circ \sigma_e$ goes around the petal corresponding to $e$. Thus the homotopy classes of paths $q_*([\sigma_e]) = [q \circ \sigma_e]$ form a basis for the free group $\pi_1(G/T, *)$: it follows that the homotopy classes of paths $[\sigma_e]$ for a basis for the free group $\pi_1(G, *)$. □

The fundamental group of a finite connected graph $G$ is thus a finitely generated free group. In particular we can define the **rank** of $G$ to be $\mathrm{rank}(G) = \mathrm{rank}(\pi_1(G, *))$, which doesn't depend on the choice of the basepoint. We define $\|G\|_{\mathrm{e}}$ to be the number of edges of $G$.

**Lemma 1.1.14.** *Let $G$ be a finite connected graph. Then every maximal tree for $G$ has the same number of edges, which is equal to $\|G\|_{\mathrm{e}} - \mathrm{rank}(G)$.*

*Proof.* By Proposition 1.1.13 we have that $\pi_1(G, *)$ is a free group whose basis is in bijection with the edges of $G \setminus T$, where $T$ is any maximal tree for $G$. □

### 1.1.5 The core graph

**Definition 1.1.15.** *Let $(G, *)$ be a connected pointed graph. Define its **pointed core graph** $\mathrm{core}_*(G)$ as the subgraph given by the basepoint and the union of (the images of) all the reduced paths from the basepoint to itself.*

We observe that $\mathrm{core}_*(G)$ is connected. If $C$ is a connected component of $G \setminus \mathrm{core}_*(G)$, then $\overline{C}$ is a tree that is the union of $C$ and of a single point of $\mathrm{core}_*(G)$; in other words, we can think $G$ as the union of $\mathrm{core}_*(G)$ and of a family of trees, where each tree has a single vertex which is glued onto a vertex of $\mathrm{core}_*(G)$. It

follows that $\text{core}_*(G)$ is a deformation retract of $G$, and in particular the inclusion $i : \text{core}_*(G) \to G$ is a (pointed) homotopy equivalence and induces an isomorphism $\pi_1(\text{core}_*(G), *) = \pi_1(G, *)$.

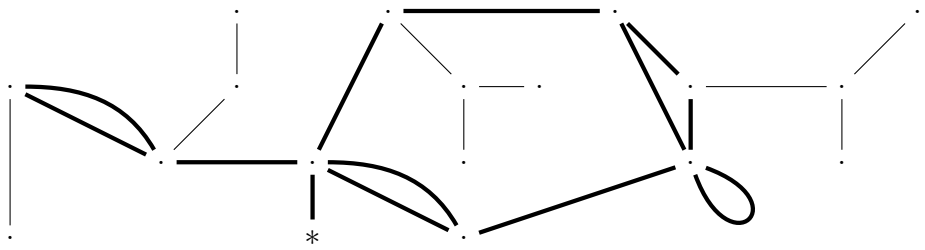


Figure 1.2: A graph $G$ decomposed as the $\text{core}_*(G)$ (in bold) with four trees glued around.

**Definition 1.1.16.** *Let $G$ be a connected graph which is not a tree. Define its* ***core graph*** $\text{core}(G)$ *as the subgraph given by the union of (the images of) all the reduced loops.*

Suppose $(G, *)$ is a pointed graph. If the basepoint belongs to $\text{core}(G)$ then we have $\text{core}_*(G) = \text{core}(G)$. Otherwise there is a unique shortest reduced path $\gamma$ connecting the basepoint to a vertex of $\text{core}(G)$, and we have that $\text{core}_*(G) = \text{core}(G) \cup \text{im}(\gamma)$.

## 1.2 Stallings' folding

This whole section is based on the fundamental correspondence between subgroups of $F_n$ and covering spaces of the rose with $n$ petals $R_n$. Usually, the covering space associated to a subgroup is infinite; however, when the subgroup is finitely generated, it is possible to restrict our attention to the subgraph given by the (pointed) core, which turns out to be finite, and already contains all the needed information. We show how the core graph can be obtained with the classical Stallings' folding operations, introduced by Stallings in [Sta83]. We also discuss some of the most classical consequences of these notions.

### 1.2.1 Labeled graphs

Let $F_n$ be a finitely generated free group generated by $a_1, ..., a_n$. We denote with $\overline{a}_i = a_i^{-1}$ the inverse $a_i$. We denote with $R_n$ the standard $n$-rose, i.e. the graph with one vertex $*$ and $n$ oriented edges labeled $a_1, ..., a_n$, see figure 1.3. The fundamental group $\pi_1(R_n, *)$ will be identified with $F_n$: the path going along the edge labeled $a_i$ (with the right orientation) corresponds to the element $a_i \in F_n$.

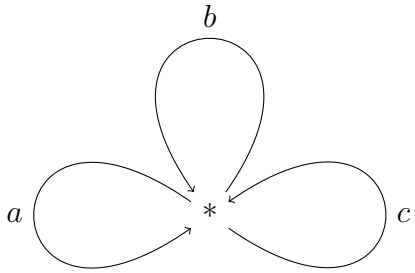


Figure 1.3: The rose $R_3$ for the free group $F_3 = \langle a, b, c \rangle$.

**Definition 1.2.1.** *A **labeled graph** is a graph $G$ with a combinatorial map $f : G \to R_n$*

This means that every edge of $G$ is equipped with a label in $\{a_1, ..., a_n\}$ and an orientation, according to which edge of $R_n$ it is mapped to; the map $f : G \to R_n$ is called the **labeling map** for $G$.

**Definition 1.2.2.** *Let $G_0, G_1$ be labeled graphs with labeling maps $f_0, f_1$ respectively. A map $h : G_0 \to G_1$ is called **label-preserving** if $f_1 \circ h = f_0$.*

This means that the map $h$ sends each vertex to a vertex, and each open edge homeomorphically onto an edge with the same label and orientation. In particular, $h$ is a combinatorial map.

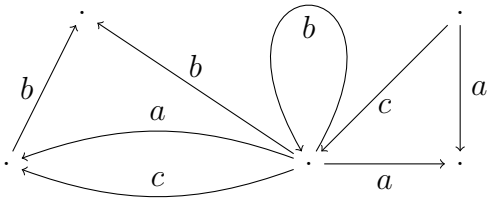


Figure 1.4: An example of labeled graph over the free group $F_3 = \langle a, b, c \rangle$.

If $\sigma : I_l \to G$ is a combinatorial path inside a labeled graph $G$, then we can define the **word that we read along** $\sigma$ to be the word $w_\sigma$ given by the sequence of letters that we read while going along $\sigma$; when we cross an edge labeled $a_i$ with the right orientation, we add $a_i$ at the end of our word, while when we cross an edge labeled $a_i$ with the opposite orientation, we add $\overline{a}_i$ at the end of our word. The word $w_\sigma$ represents the element of the free group $F_n$ given by the homotopy class $[f \circ \sigma] \in \pi_1(R_n, *)$ where $f : G \to R_n$ is the labeling map.

### 1.2.2 The core graph of a subgroup

Let $H \leq F_n$ be a subgroup. From the theory of covering spaces, we know that there is a pointed covering space $c : (G_H, *) \to (R_n, *)$ such that $c_*(\pi_1(G_H, *)) = H$, and it is unique, up to isomorphism of pointed covering spaces.

**Definition 1.2.3.** *For a subgroup $H \leq F_n$ define the pointed graph* $(\mathrm{cov}(H), *) = (G_H, *)$.

The space $\mathrm{cov}(H)$ is a labeled graph with labeling map $c : \mathrm{cov}(H) \to R_n$. The map $c_* : \pi_1(\mathrm{cov}(H), *) \to F_n$ is injective and thus we can identify $\pi_1(\mathrm{cov}(H), *) = H$. In Figures 1.5, 1.6, 1.7, 1.8 we can see the space $\mathrm{cov}(H)$ for some examples of subgroups $H \leq F_2$.

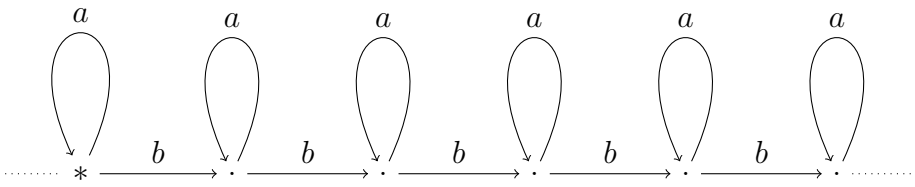


Figure 1.5: (A portion of) the covering space $\mathrm{cov}(H)$ for the normal subgroup $H = \langle\langle a \rangle\rangle$ of the free group $F_2 = \langle a, b \rangle$. Notice that $H$ is finitely generated as a normal subgroup, but not as a subgroup.

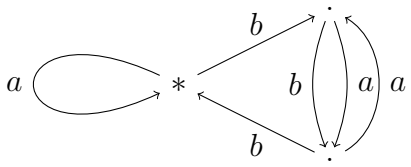


Figure 1.6: The covering space $\mathrm{cov}(H)$ for the (finite index) subgroup $H = \langle a, b^3, bab, b\bar{a}b \rangle$ of the free group $F_2 = \langle a, b \rangle$.

**Definition 1.2.4.** *For a non-trivial subgroup $H \leq F_n$, define its* ***pointed core graph*** *as the pointed graph* $(\mathrm{core}_*(H), *) = (\mathrm{core}_*(\mathrm{cov}(H)), *)$.

Of course $\mathrm{cov}(H)$ characterizes the conjugacy class of the subgroup $H$, but is usually infinite, unless $H$ has finite index in $F_n$. The idea is that $\mathrm{core}_*(H)$ still contains all the information that we need about the subgroup $H$, but is usually smaller than $\mathrm{cov}(H)$; as we will see, $\mathrm{core}_*(H)$ is finite whenever $H$ is finitely generated. We will provide an explicit, algorithmic construction that, given a finite set of generators for $H$, produces the finite graph $\mathrm{core}_*(H)$. We will show that $\mathrm{cov}(H)$ can be reconstructed from $\mathrm{core}_*(H)$ by gluing a finite number of branches of the universal cover.

**Definition 1.2.5.** *For a non-trivial subgroup $H \leq F_n$, define its* ***core graph*** *as the unpointed graph* $\mathrm{core}(H) = \mathrm{core}(\mathrm{cov}(H))$.

When considering $\mathrm{core}(H)$ instead of $\mathrm{core}_*(H)$, we lose track of the basepoint: two subgroups have the same core graph if and only if they are conjugate to each other.

### 1.2.3 Gluing branches of the universal cover

**Lemma 1.2.6.** *Let $G$ be a labeled graph with labeling map $f : G \to R_n$. Then the following are equivalent:*

*(i) The map $f$ is a covering map.*

*(ii) For every vertex $v$ of $G$ and each $i = 1, ..., n$, there is exactly one edge labeled $a_i$ going into $v$ and one going out.*

*Proof.* Immediate from the definition of covering space. □

In order to make an arbitrary labeled graph $G$ into a covering space, we have to deal with two kinds of problems. One problem is given by vertices $v$ of $G$ that do not have an edge with a certain label going out or into $v$; this will be solved in this section by gluing branches of the universal cover $\widetilde{R}_n$ to $G$ at $v$. The other problem is given by vertices $v$ of $G$ that have multiple edges with the same label going out or into $v$; this will be solved by folding together the edges with the same label, see Section 1.2.4.

Let $\widetilde{R}_n$ be the universal cover of $R_n$, which is a tree. We call **$a_i$-branch of $\widetilde{R}_n$** the subgraph given by the union of (the images of) all reduced paths that start at the basepoint $* \in \widetilde{R}_n$ and whose first edge is labeled $a_i$, going out of $*$ (see Figure 1.7); similarly, we call **$\overline{a}_i$-branch of $\widetilde{R}_n$** the subgraph given by the union of (the images of) all reduced paths that start at $*$ and whose first edge is labeled $a_i$, going into $*$.

Let $G$ be a labeled graph with labeling map $f : G \to R_n$. Fix $i \in \{1, ..., n\}$ and fix a vertex $v$ of $G$. Suppose there is no edge going into (resp. out of) $v$ with label $a_i$: then we take a copy $B$ of the $a_i$-branch (resp $\overline{a}_i$-branch) of $\widetilde{R}_n$ and we glue it to $G$, identifying the unique valence-1 vertex of $B$ with $v$. We perform this gluing for each $i = 1, ..., n$ and for each vertex $v$ of $G$ at the same time: we denote by $\widehat{G}$ the resulting labeled graph.

**Proposition 1.2.7.** *Let $(G, *)$ be a connected pointed labeled graph with labeling map $f : G \to R_n$ and let $H = f_*(\pi_1(G, *)) \leq F_n$. Suppose for each vertex $v$ of $G$ and for each $i = 1, ..., n$ there is at most one edge labeled $a_i$ going into $v$ and at most one going out of $v$. Then the pointed graph $(\widehat{G}, *)$ built above coincides with $(\mathrm{cov}(H), *)$.*

*Proof.* Let $\widehat{f} : \widehat{G} \to R_n$ be the labeling map. From the definition of $\widehat{G}$ we have that, for each vertex $v$ of $\widehat{G}$ and for each $i = 1, ..., n$, there is exactly one edge

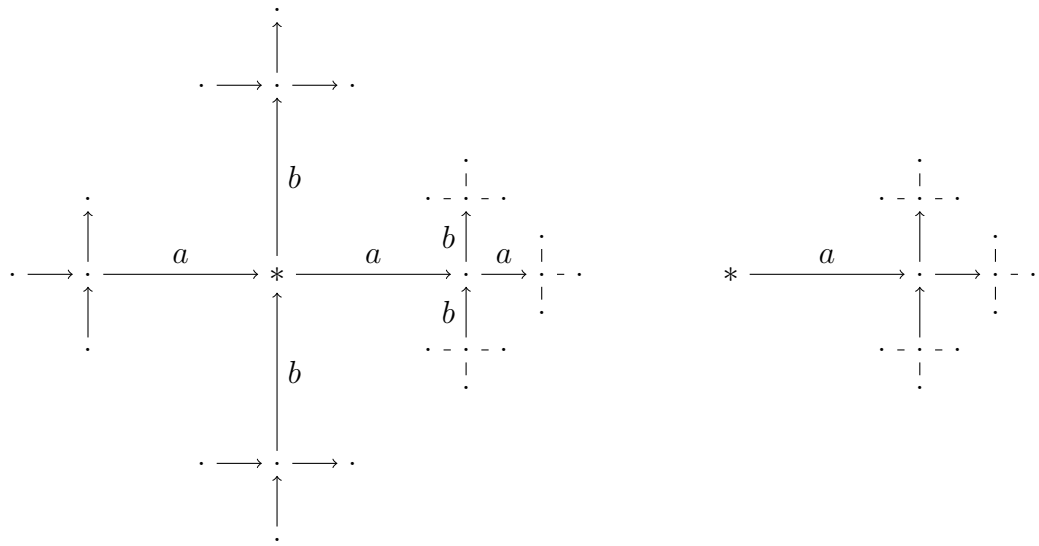


Figure 1.7: The space $\mathrm{cov}(1)$ associated with the trivial subgroup $1 \leq F_n$ coincides with the universal cover $\widetilde{R}_n$ of the rose $R_n$. On the left we can see (a portion of) the graph $\widetilde{R}_2$ when the free group is $F_2 = \langle a, b \rangle$. On the right we can see the $a$-branch of $\widetilde{R}_2$.

labeled $a_i$ going out of $v$ and one going into $v$. By Lemma 1.2.6 this shows that $f$ is a covering map. But since $\widehat{G}$ is obtained from $G$ by gluing trees, we have $\pi_1(G, *) = \pi_1(\widehat{G}, *)$ and thus $\widehat{f}_*(\pi_1(\widehat{G}, *)) = f_*(\pi_1(G, *)) = H$. The conclusion follows from the uniqueness of the covering space $\mathrm{cov}(H)$. □

**Corollary 1.2.8.** *Let $H \leq F_n$ be a subgroup. Then $\mathrm{cov}(H)$ coincides with $\widehat{\mathrm{core}_*(H)}$ as pointed labeled graph.*

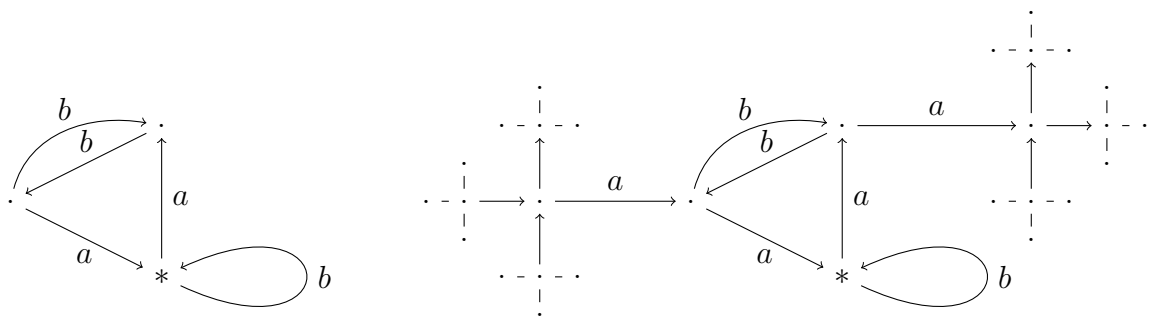


Figure 1.8: Consider the subgroup $H = \langle b, aba, a\bar{b}a \rangle$ of the free group $F_2 = \langle a, b \rangle$. On the left we can see the pointed core graph $\mathrm{core}_*(H)$, while on the right we can see (a portion of) the graph $\mathrm{cov}(H)$. Notice that $\mathrm{cov}(H)$ consists of $\mathrm{core}_*(H)$ with two branches of $\widetilde{R}_2$ glued at two different vertices.

### 1.2.4 Stallings' folding

We now define the fundamental folding operations, first introduced by Stallings in [Sta83]. Let $G$ be a labeled graph and suppose there are two distinct edges $e_1, e_2$ with endpoints $v, v_1$ and $v, v_2$ respectively. Suppose that $e_1$ and $e_2$ have the same label and orientation. We can identify $v_1$ with $v_2$, and $e_1$ with $e_2$: we then get a label-preserving quotient map of graphs $p : G \to G'$.

**Definition 1.2.9.** *The quotient map $p : G \to G'$ is called **Stallings folding**.*

Folding operations are the graph-analogue of canceling two adjacent letters in a word when the letters are inverse to each other. If a combinatorial path $\sigma : I_l \to G$ goes from $v_1$ to $v$ to $v_2$ through $e_1$ and $e_2$ respectively, then the word that we read along $\sigma$ will contain an occurrence of $...a_i\overline{a}_i...$ or $...\overline{a}_i a_i...$ where $a_i$ is the label of $e_1, e_2$. After the folding operation, we can change the path $\sigma : I_l \to G'$ by a homotopy, since now we have $v_1 = v_2$, so we don't need to cross neither of $e_1, e_2$, and in the word that we read along $\sigma$ those two letters disappear.

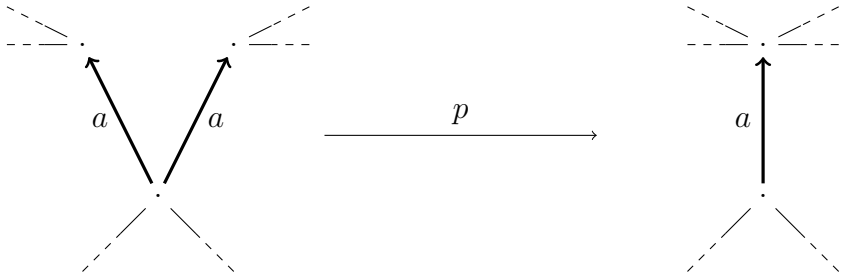


Figure 1.9: An example of a folding operation. In bold the edges involved in the folding.

Given a finite connected labeled graph $G$, we can successively apply folding operations to $G$ in order to get a sequence $G = G^0 \to G^1 \to ... \to G^l$. Notice that the number of edges decreases by 1 at each step, and thus the length of any such chain is bounded (by the number of the edges of $G$). The following proposition, although not explicitly stated in [Sta83], is a direct consequence.

**Proposition 1.2.10.** *Let $G$ be a finite connected labeled graph and let $G = G^0 \to G^1 \to ... \to G^m$ be a maximal sequence of folding operations. Also, fix a basepoint $* \in G$, inducing a basepoint $* \in G^i$ for $i = 0, ..., l$. Then we have the following:*

*(i) Each such sequence has the same length $m$ and the same final graph $G^m$.*

*(ii) Let $f^i : G^i \to R_n$ be the labeling map. Then the image of $f^i_* : \pi_1(G^i, *) \to \pi_1(R_n, *)$ is the same subgroup $H \leq F_n$ for every $i = 1, ..., m$.*

*(iii) For every $i = 1, ..., m$ there is a unique label-preserving map of pointed graphs $h^i : G^i \to \mathrm{cov}(H)$. The image $\mathrm{im}(h^i)$ is the same subgraph of $\mathrm{cov}(H)$ for every $i = 1, ..., m$.*

*(iv) The map $h^m$ is an embedding of $G^m$ as a subgraph of $\mathrm{cov}(H)$ and the subgraph $h^m(G^m)$ contains $\mathrm{core}_*(H)$. In particular $h^m_* : \pi_1(G^m, *) \to \pi_1(\mathrm{cov}(H), *)$ is an isomorphism and the map $f^m_* : \pi_1(G^m, *) \to F_n$ is injective.*

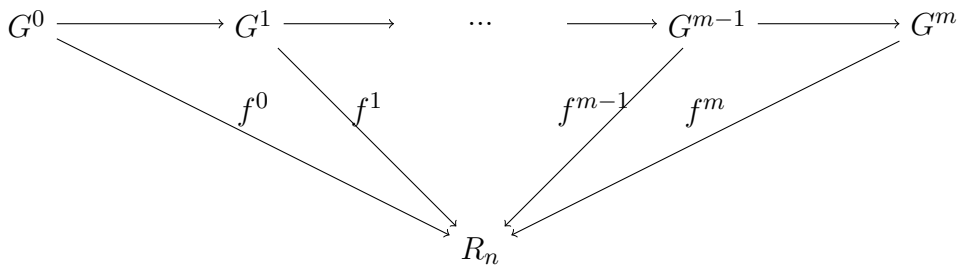


Figure 1.10: The diagram commutes.

*Proof.* It is easy to prove that the maps $f^i_* : \pi_1(G^i, *) \to \pi_1(R_n, *)$ and $f^{i+1}_* : \pi_1(G^{i+1}, *) \to \pi_1(R_n, *)$ have the same image. Now (ii) follows by induction.

We now consider the covering space $c : (\mathrm{cov}(H), *) \to (R_n, *)$. The labeling map $f^i : (G^i, *) \to (R_n, *)$ satisfies the condition $f^i_*(\pi_1(G^i, *)) \subseteq c_*(\pi_1(\mathrm{cov}(H), *))$ on the fundamental groups; by standard results about covering spaces, this implies the existence of a unique topological lifting of the map $f^i : (G^i, *) \to (R_n, *)$ to a map $h^i : (G^i, *) \to (\mathrm{cov}(H), *)$. The condition of being a lifting (i.e. of having $c \circ h^i = f^i$) is equivalent to the condition of $h^i$ being label-preserving.

It is immediate to prove that the images of the maps $h^i : G^i \to \mathrm{cov}(H)$ and $h^{i+1} : G^{i+1} \to \mathrm{cov}(H)$ are the same. Now (iii) follows by induction.

Consider the graph $G^m$ and observe that, for each vertex $v$ and for each $i = 1, ..., n$, there is at most one edge labeled $a_i$ going out of $v$ and one going into $v$. We can then glue branches of $\widetilde{R}_n$ to the graph $G^m$ as described in Section 1.2.3, in order to obtain a labeled graph $X$ which is a covering space of $R_n$ with fundamental group $H$. By the uniqueness of the pointed covering space, we have $X = \mathrm{cov}(H)$ we obtain that $h^m$ is an embedding of $G^m$ as a subgraph of $\mathrm{cov}(H)$ and that $G^m$ contains $\mathrm{core}_*(H)$. This proves (iv).

Finally, we observe that the final graph $G^m$ is the image of the unique map $h^0 : G \to \mathrm{cov}(H)$, and thus it doesn't depend on the chosen folding sequence. Also, the number $m$ of folding operations required is the difference between the number of edges of $G$ and the number of edges of $G^m$, so it doesn't depend on the chosen folding sequence either. This proves (i). □

**Definition 1.2.11.** *Let $G$ be a finite connected labeled graph. Define its* ***folded graph*** $\mathrm{fold}(G)$ *to be the labeled graph $G^m$ obtained from any maximal sequence of folding operations as in Proposition 1.2.10.*

It's sometimes important to be able to control the valence-1 vertices along a chain of folding operations; the following lemma helps us with this.

**Lemma 1.2.12.** *Let $(G, *)$ be a finite connected labeled graph. Let $G \to G^1 \to ... \to G^k$ be a chain of folding operations, and let $p_k : G \to G^k$ be the composition of the chain. Suppose that for some vertex $v$ of $G$, the vertex $p_k(v)$ of $G^k$ has valence one. Then all the edges at $v$ in $G$ have the same label and orientation (see figure 1.11).*

*Proof.* Suppose that $p_k(v)$ has valence one in $G^k$. Let $e, e'$ be edges of $G$ going out of $v$: we observe that $p_k(e), p_k(e')$ are both edges of $G^k$ going out of $p_k(v)$. But then they must coincide, since there is only one edge of $G^k$ going out of $p_k(v)$. In particular, $p_k(e), p_k(e')$ have the same label and orientation, and thus $e, e'$ have the same label and orientation too, since $p_k$ is label-preserving. □

**Corollary 1.2.13.** *Let $(G, *)$ be a finite connected labeled graph and let $G \to G^1 \to ... \to G^m$ be any maximal sequence of folding operations as in Proposition 1.2.10. Suppose for each vertex $v$ of $G$, there are two edges going out of $v$ with different labels, or with the same label but different orientations. Then there is no valence-1 vertex in any graph of the sequence.*

*Proof.* Suppose some graph $G^k$ contains a valence-1 vertex $u$. Let $p_k : G \to G^k$ be the map given by the composition of the folding operations: since $p_k$ is surjective, we can find a vertex $v$ of $G$ with $p_k(v) = u$. But then by Lemma 1.2.12, all the edges going out of $v$ have the same label and orientation, contradicting the hypothesis. □

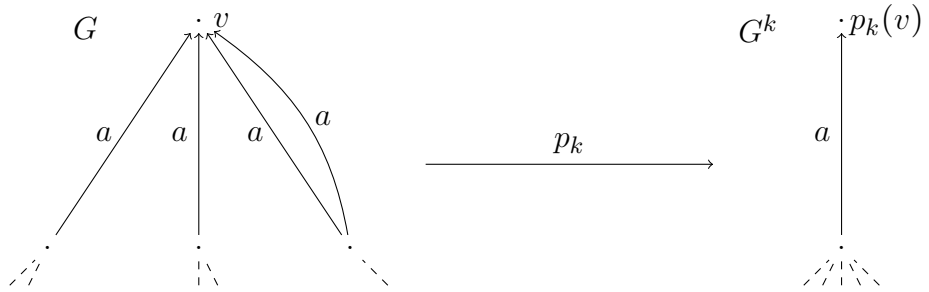


Figure 1.11: An example of a graph $G$ with a vertex $v$ that is a candidate to become a valence-1 vertex, after a few folding operations. Edges from $v$ to itself are not allowed. Multiple edges from $v$ to another vertex are allowed (but they must have the same label and orientation).

One of the most important consequences of Proposition 1.2.10 is that, given a finite set of generators for a subgroup $H \leq F_n$, we can explicitly construct its core graph $\text{core}_*(H)$, as we now explain.

Suppose we are given a finite set of reduced words $w_1, ..., w_k \in F_n$ of lengths $l_1, ..., l_k$, and let $H = \langle w_1, ..., w_k \rangle$. We can construct the graph $G$ given by a basepoint $*$ and pairwise disjoint loops $\gamma_1, ..., \gamma_k$ starting and ending at the basepoint. The loop $\gamma_i$ is subdivided into $l_i$ edges, labeled and oriented according to the letters of the word $w_i$, in such a way that, when going along $\gamma_i$, we read exactly the word $w_i$.

**Proposition 1.2.14.** *In the setting above, we have* $\mathrm{core}_*(H) = \mathrm{fold}(G)$.

*Proof.* Proposition 1.2.10 tells us that $\mathrm{fold}(G)$ is a finite subgraph of $\mathrm{cov}(H)$ and that it contains $\mathrm{core}_*(H)$. Corollary 1.2.13 tells us that $\mathrm{fold}(G)$ has no valence-1 vertex, except possibly for the basepoint. But a finite subgraph of $\mathrm{cov}(H)$, containing $\mathrm{core}_*(H)$, and with no valence-1 vertex except possibly the basepoint, must be equal to $\mathrm{core}_*(H)$. The conclusion follows. □

It follows that there is an algorithm that takes in input a finite set of generators for a subgroup $H \leq F_n$ and gives as output the graph $\mathrm{core}_*(H)$. In what follows, for algorithmic purposes, when we say that we are given a finitely generated subgroup $H$, we mean that we are given a finite set of generators for $H$; this means we can construct the finite graph $\mathrm{core}_*(H)$.

### 1.2.5 Applications

**Proposition 1.2.15.** *Every subgroup of* $F_n$ *is itself a free group.*

*Proof.* Let $H \leq F_n$ be a subgroup. Then we have $H = \pi_1(\mathrm{cov}(H), *)$ but $\mathrm{cov}(H)$ is a graph, and thus its fundamental group is a free group. □

**Proposition 1.2.16.** *A subgroup* $H \leq F_n$ *is finitely generated if and only if* $\mathrm{core}_*(H)$ *is finite.*

*Proof.* If $H$ is finitely generated then Proposition 1.2.14 tells us that $\mathrm{core}_*(H)$ is finite. If $\mathrm{core}_*(H)$ is finite, then $H = \pi_1(\mathrm{cov}(H), *) = \pi_1(\mathrm{core}_*(H), *)$ is the fundamental group of a finite graph, and thus is finitely generated. □

**Proposition 1.2.17.** *A subgroup* $H \leq F_n$ *has finite-index if and only if* $\mathrm{core}_*(H)$ *is a finite covering space of* $R_n$*. Moreover, if* $H$ *has finite index* $d$*, then* $H$ *is a free group of finite rank* $1 + d(n-1)$.

*Proof.* If $\mathrm{core}_*(H)$ is a finite covering space, then $\mathrm{core}_*(H) = \mathrm{cov}(H)$ is finite and thus $H$ has finite index. Conversely, if $H$ has finite index $d$, then $\mathrm{cov}(H)$ is a finite graph with $d$ vertices and $dn$ edges, and each vertex has valence $2n$. In particular $\mathrm{cov}(H)$ is a finite graph without valence-1 vertices, and thus $\mathrm{core}_*(H) = \mathrm{core}_*(\mathrm{cov}(H)) = \mathrm{cov}(H)$ and thus $\mathrm{core}_*(H)$ is a finite covering space of $R_n$. In that case, every maximal tree of $\mathrm{cov}(H)$ has $n-1$ edges, and the remaining $(d-1)n+1$ edges are the generators of $\pi_1(\mathrm{cov}(H), *) = H$. □

**Proposition 1.2.18** (Subgroup membership problem)**.** *There is an algorithm that, given a finitely generated free group* $H$ *and an element* $w \in F_n$*, determines whether* $w \in H$.

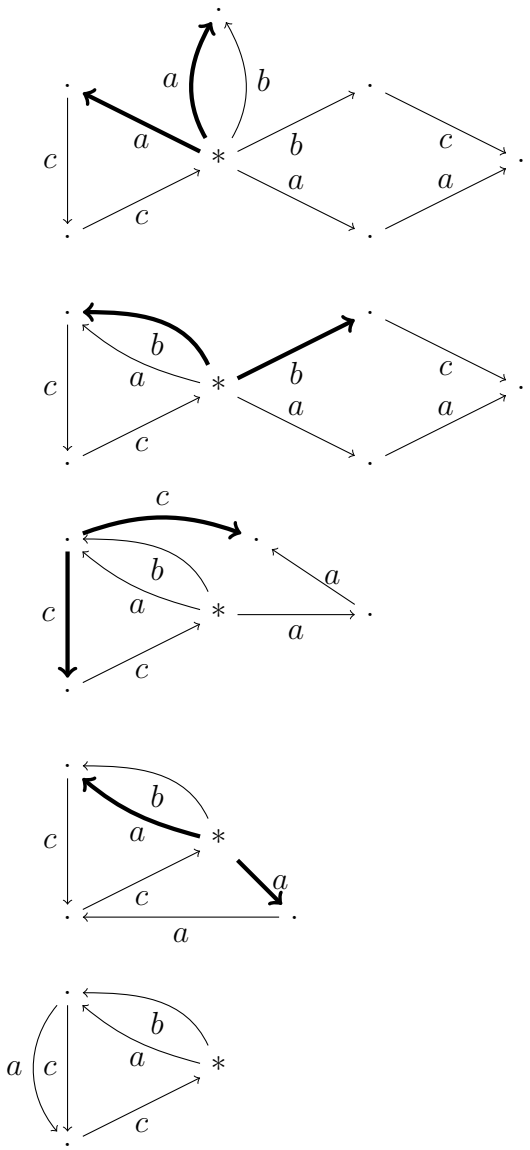


Figure 1.12: The algorithm to produce $\mathrm{core}_*(H)$ given a finite set of generators for $H$. In the figure we consider $F_3 = \langle a, b, c\rangle$ and $H = \langle acc, b\bar{a}, aa\bar{c}\bar{b}\rangle$. The first graph consists of a basepoint and three paths from the basepoint to itself; along the three paths we can read the generators of $H$. Then we apply a sequence of folding operations; at each step we can see in bold the two edges that are going to be identified at the next step. The last graph is $\mathrm{core}_*(H)$.

*Proof.* Let $\sigma : I_l \to R_n$ be the reduced path representing the element $w$. Consider the lifting $\widetilde{\sigma} : I_l \to \mathrm{cov}(H)$ that starts at the basepoint: we have that $w \in H$ if and only if $\widetilde{\sigma}$ ends at the basepoint too. Notice that the words that we read along $\sigma$ is the same that we read along $\widetilde{\sigma}$, i.e. the reduced word representing $w$.

Consider $\mathrm{core}_*(H)$, start at the basepoint, and read the reduced word for $w$. Each

time we read a letter, we move in $\text{core}_*(H)$ along the edge labeled with that letter (in the right direction); notice that there is at most one edge with the right label and orientation. There are three cases:

(i) At some point we aren't able to move, as there are no edges with the right label and orientation at the vertex we are in. This means that the path $\tilde{\sigma}$ enters one of the branches of $\tilde{R}_n$, as described in Section 1.2.3. Once a reduced path enters one such branch, there is no way it can come back: this means that the other endpoint of $\tilde{\sigma}$ can't be the basepoint, and thus $w \notin H$.

(ii) We are able to read the whole word $w$ and move accordingly, but the other endpoint of $\tilde{\sigma}$ isn't the basepoint. This means that $w \notin H$.

(iii) We are able to read the whole word $w$ and move accordingly, and the other endpoint of $\tilde{\sigma}$ is the basepoint. This means that $w \in H$. □

**Proposition 1.2.19.** *There is an algorithm that, given two finitely generated free groups $H, K \leq F_n$, determines whether $H = K$.*

*Proof.* It is enough to check whether $\text{core}_*(H) = \text{core}_*(K)$. □

**Theorem 1.2.20** (Marshall Hall)**.** *Let $H \leq F_n$ be finitely generated. Then there is a subgroup $H \leq K \leq F_n$ such that:*

*(i) There is a decomposition $K = H * H'$ for some subgroup $H' \leq K$.*

*(ii) $K$ has finite index in $F_n$.*

*Proof.* We take the graph $\text{core}_*(H)$ and we add (labeled) edges to it. The following property is true for $\text{core}_*(H)$: every vertex has at most one edge entering and at most one edge exiting for each label; we want this property to stay true every time we add an edge.

Fix a generator of $F_n$, let's say $a_1$. Suppose there is a vertex $v_0$ with no edge labeled $a_1$ entering. If $v_0$ has no edge labeled $a_1$ exiting, then we add an edge labeled $a_1$ going from $v_0$ to itself. If there is an edge labeled $a_1$ exiting from $v_0$, then we follow that edge to a vertex $v_1$; if $v_1$ has an edge labeled $a_1$ exiting, we follow that edge too, and so on, until we reach a vertex $v_k$ with no edge labeled $a_1$ exiting; we then add an edge labeled $a_1$ going from $v_k$ to $v_0$.

Using the procedure described in the preceding paragraph, add to $\text{core}_*(H)$ new oriented edges labeled $a_1$, until every vertex has exactly one edge labeled $a_1$ entering. In the same way, we can add edges labeled $a_1$ in such a way that every vertex has exactly one edge labeled $a_1$ exiting. Apply the same procedure for the other generators $a_2, ..., a_n$.

In the end we get a new graph $(Z, *)$, with the same vertices of $\text{core}_*(H)$, but with more edges: to be precise, in $Z$ every vertex has exactly one edge labeled $a_i$ entering, and one exiting, for every $i$. In other words, the map $(Z, *) \to (R_n, *)$ is a covering space of finite degree, corresponding to a finite index subgroup $K = \pi_1(Z, *)$ of $F_n$.

Fix any maximal tree for $\mathrm{core}_*(H)$ and observe that the generators for $K$ are exactly the generators of $H$, plus the generators given by the edges we added: this gives us a decomposition $K = H * H'$, where $H'$ is the subgroup generated by the edges that we added. □

**Theorem 1.2.21** (Intersection of subgroups)**.** *Let $H, H' \leq F_n$ be finitely generated subgroups. Then $H \cap H'$ is finitely generated. Moreover there is an algorithm that, given $\mathrm{core}_*(H)$ and $\mathrm{core}_*(H')$, computes $\mathrm{core}_*(H \cap H')$.*

*Proof.* We define a labeled graph $G$ as follows; this is the fibre product of the two labeling maps $\mathrm{core}_*(H) \to R_n$ and $\mathrm{core}_*(H') \to R_n$. The graph $G$ has a vertex $(v, v')$ for each couple of vertices $v$ of $\mathrm{core}_*(H)$ and $v'$ of $\mathrm{core}_*(H')$. The graph $G$ has an edge $(e, e')$ for each couple of edges $e$ of $\mathrm{core}_*(H)$ and $e'$ of $\mathrm{core}_*(H')$ with the same label; if $e$ goes from $v_0$ to $v_1$ (according to its orientation) and $e'$ from $v_0'$ to $v_1'$ (according to its orientation), then $(e, e')$ goes from $(v_0, v_0')$ to $(v_1, v_1')$ (with this orientation) and shares the same label as $e$ and $e'$. The graph $G$ has a basepoint $(*, *)$ given by the couple of basepoints of $\mathrm{core}_*(H)$ and $\mathrm{core}_*(H')$. There are also two natural label-preserving projection maps $p : G \to \mathrm{core}_*(H)$ and $p' : G \to \mathrm{core}_*(H')$.

Let $C$ be the connected component of $G$ containing the basepoint, let $f : C \to R_n$ be the labeling map and let $K = f_*(\pi_1(C, *)) \leq F_n$. By the definition we have that $C$ satisfies the hypothesis of Proposition 1.2.7, and thus $\widehat{C} = \mathrm{cov}(K)$ and in particular $\mathrm{core}_*(C) = \mathrm{core}_*(K)$. It remains to prove that $K = H \cap H'$, since that will mean that $\mathrm{core}_*(C) = \mathrm{core}_*(H \cap H')$ and we will be done.

The natural label-preserving projection map $p : G \to \mathrm{core}_*(H)$ restricts to a pointed label-preserving map $p : C \to \mathrm{core}_*(H)$; it follows that $K \leq H$. Similarly we have $K \leq H'$, and thus also $K \leq H \cap H$.

Let $w$ be a reduced word representing an element of $H \cap H'$. Let $\sigma : I_l \to \mathrm{core}_*(H)$ with $\sigma(0) = \sigma(1) = *$ be the unique reduced path such that that the word that we read along $\sigma$ is $w$; similarly, let $\sigma' : I_l \to \mathrm{core}_*(H')$ with $\sigma'(0) = \sigma'(1) = *$ be the unique reduced path such that that the word that we read along $\sigma'$ is $w$. Then we have a path $(\sigma, \sigma') : I_l \to G$ whose $j$-th edge is the couple given by the $j$-th edge if $\sigma$ and by the $j$-th edge of $\sigma'$: the image of this path is contained in $C$ and the word that we read along $(\sigma, \sigma')$ is $w$. This proves that $w \in K$; it follows that $H \cap H' \leq K$. □

## 1.3 Homomorphisms and kernels

Folding operations don't always preserve the rank of the graph. The folding operations that decrease the rank of the graph are the ones responsible for the kernel of homomorphisms between free groups. We now show that these operations can always be postponed until the end of a folding sequence. This provides a clean way of computing the kernel of a homomorphism between free groups.

### 1.3.1 Rank-preserving and non-rank-preserving folding

Let $G$ be a labeled graph and let $q : G \to G'$ be a folding operation.

**Definition 1.3.1.** *A Stallings folding $q : G \to G'$ is called **rank-preserving** if it is a homotopy equivalence.*

In that case, for every basepoint $* \in G$, the map $q : (G, *) \to (G', q(*))$ is a pointed homotopy equivalence and $q_* : \pi_1(G, *) \to \pi_1(G', *)$ is an isomorphism. Being rank-preserving is equivalent to the requirement that the endpoints that we are identifying are distinct (see also figure 1.13). In fact, the rank of the fundamental group of a finite connected graph is $E - V + 1$ where $E$ is the number of edges and $V$ is the number of vertices; during a folding operation, the number of edges always decreases by exactly one; if we identify two distinct vertices, then the number of vertices decreases by one too, and thus the rank is preserved; otherwise the number of vertices remains the same, and thus the rank decreases by one.

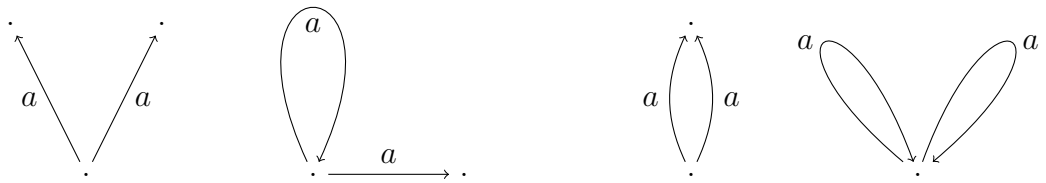


Figure 1.13: Examples of configurations where a folding operation is possible. The two examples on the left produce rank-preserving folding operations; the two examples on the right produce non-rank-preserving folding operations.

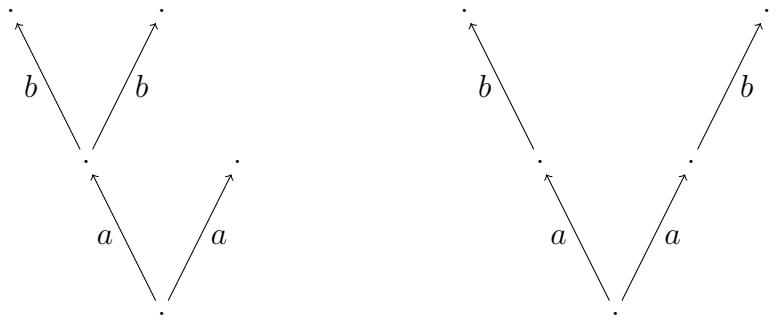


Figure 1.14: Above we have two examples of graphs, and in each of them we want to perform two folding operations (one involving $a$-labeled edges, and the other involving $b$-labeled edges). In the graph on the left, we can perform the two operations in any order. In the graph on the right, we are forced to perform the operation on the $a$-labeled edges first.

In a sequence of folding operations as in Proposition 1.2.10, it is not always possible to change the order of the operations; see for example figure 1.14. Informally, we could

say that certain folding operations are required before being able to perform other operations. The key observation is that the non-rank-preserving folding operations change the set of edges of $G$, but they do not change the set of vertices of $G$; as a consequence, they are not a requirement for any other operation. This can be made precise as follows.

**Proposition 1.3.2.** *Let $G$ be a finite connected labeled graph. Let $G = G^0 \to G^1 \to ... \to G^k$ be a maximal sequence of rank-preserving folding operations. Let $G^k \to G^{k+1} \to ... \to G^m$ be a maximal sequence of folding operations for $G^k$. Then we have the following:*

*(i) Each map in the first sequence is a homotopy equivalence; the map $G^0 \to G^k$ is a homotopy equivalence.*

*(ii) The second sequence only contains non-rank-preserving folding operations; the map $G^k \to G^m$ is an isomorphism on the set of vertices.*

*(iii) The concatenation of the two sequences produces a folding sequence as in Proposition 1.2.10. In particular $G^m = \text{fold}(G)$.*

*(iv) The numbers $k, m$ do not depend on the chosen sequences.*

*Remark.* This shows that the graph $G^k$ is essentially $\text{fold}(G)$, but with some edges repeated two or more times (see figure 1.15). The repeated edges (and their multiplicity) can depend on the chosen sequence of folding operations; the graph $G^k$ is not uniquely determined by $G$.

*Proof.* Part (i) is trivial.

For (ii), suppose the sequence $G^k \to ... \to G^m$ contains a rank-preserving folding operation, and let $j \geq k$ be the smallest integer such that $G^j \to G^{j+1}$ is rank-preserving; this means that there are two edges $e_1, e_2$ in $G^j$ with an endpoint $v$ in common, the other endpoints $v_1 \neq v_2$ distinct, and the same label and orientation. Let $p : G^k \to G^j$ be the composition of the sequence of folding operations $G^k \to ... \to G^j$: each of those operations is non-rank-preserving, and in particular it induces an isomorphism on the set of vertices. Thus we can take the vertices $p^{-1}(v)$ and $p^{-1}(v_1) \neq p^{-1}(v_2)$. Take any edge $e_3 \in p^{-1}(e_1)$ and $e_4 \in p^{-1}(e_2)$ and we have that in $G^k$ is it possible to fold $e_3$ and $e_4$, performing a rank-preserving folding operation. This gives a contradiction because the sequence of rank-preserving folding operations $G^0 \to ... \to G^k$ was maximal.

Part (iii) is trivial.

For part (iv), we observe the following: along the sequence $G^0 \to ... \to G^k$, at each step the number of vertices decreases by one, while along the sequence $G^k \to ... \to G^m$ the number of vertices is preserved. Thus $k$ is equal to the number of vertices of $G$ minus the number of vertices of $\text{fold}(G)$, regardless of the chosen sequence. By Proposition 1.2.10 the sum $m + k$ doesn't depend on the chosen sequence, and thus neither does $m$. □

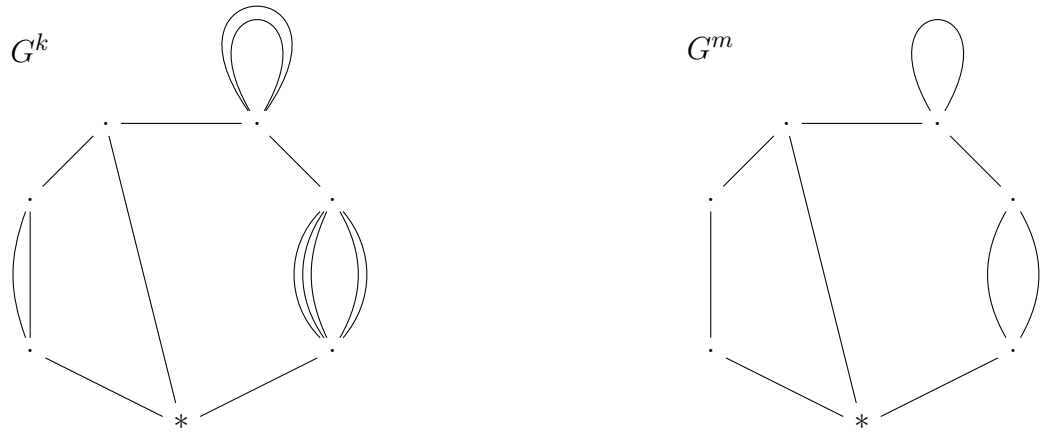


Figure 1.15: An example of the result of the folding procedure described in Proposition 1.3.2 (the labeling has been omitted). On the left the graph $G^k$, the result of the sequence of only rank-preserving folding operations. On the right the graph $G^m$ obtained from a sequence of both rank-preserving and non-rank-preserving operations.

### 1.3.2 The kernel of a homomorphism between free groups

Let $F_n = \langle a_1, ..., a_n \rangle$ and $F_r = \langle b_1, ..., b_r \rangle$ be finitely generated free groups.

**Theorem 1.3.3.** *Let $\varphi : F_r \to F_n$ be a homomorphism. Then there is a decomposition $F_r = F * F'$ such that $\varphi$ is injective on $F$ and trivial on $F'$. In particular, the kernel $\ker \varphi = \langle\langle F' \rangle\rangle$ is finitely generated as a normal subgroup of $F_r$.*

**Theorem 1.3.4.** *There is an algorithm that, given a homomorphism $\varphi : F_r \to F_n$ by means of the words $\varphi(b_1), ..., \varphi(b_r)$, determines two finite sets $M, N \subseteq F_r$ such that:*
*(i) $M \cup N$ is a basis for $F_r$.*
*(ii) $\varphi$ is injective on $\langle M \rangle$.*
*(iii) $\varphi$ is trivial on $\langle N \rangle$.*
*(iv) The length of $b_i$ as a reduced word in the basis $M \cup N$ is at most twice the length of $\varphi(b_i)$ as a reduced word in $a_1, ..., a_n$.*

*Proof of Theorems 1.3.3 and 1.3.4.* If we have $\varphi(b_r) = 1$, then we just put $b_r$ in $N$ and then we restrict our attention to the subgroup $\langle b_1, ..., b_{r-1} \rangle$. It is thus enough to prove the statement in the case where $\varphi(b_i) \neq 1$ for $i = 1, .., r$.

Let $R_n$ be the $n$-rose with edges labeled $a_1, ..., a_n$. Let $G$ be the labeled graph given by a basepoint $*$ and by paths $\gamma_i$ for $i = 1, ..., r$, such that the path $\gamma_i$ goes from the basepoint to itself and along $\gamma_i$ we can read the word $\varphi(b_i)$. Let $f : G \to R_n$ be the labeling map, inducing a map $f_* : \pi_1(G, *) \to \pi_1(R_n, *)$ between the fundamental groups. Let $\theta : F_r \to \pi_1(G, *)$ be the isomorphism sending the element $b_i \in F_r$ to the homotopy class $[\gamma_i]$, and observe that $f_* \circ \theta = \varphi$ as maps from $F_r$ to $\pi_1(R_n, *) = F_n$.

Let $G = G^0 \to ... \to G^k$ be a maximal sequence of rank-preserving folding operations and let $G^k \to ... \to G^m$ be a maximal sequence of folding operations for $G^k$, as in Proposition 1.3.2. The basepoint $* \in G$ induces a basepoint $* \in G^i$ for every $i = 0, ..., m$. Let $p : G \to G^k$ and $q : G^k \to G^m$ be the quotient maps given by the sequences of foldings. Let $f^k : G^k \to R_n$ and $f^m : G^m \to R_n$ be the labeling maps, and observe that the diagram of Figure 4.1 commutes.

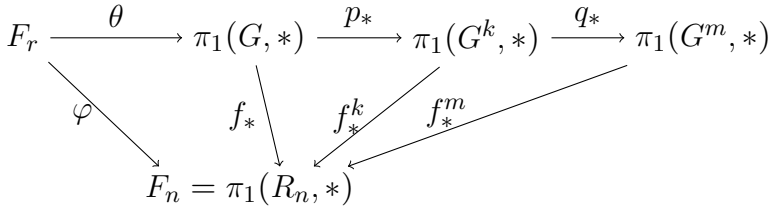


Figure 1.16: The diagram commutes.

Our plan now is to study the map $\varphi$ using the fact that $\varphi = f_*^m \circ q_* \circ p_* \circ \theta$. By Propositions 1.3.2 and 1.2.10 we have that $f_*^m$ is injective (and can thus be ignored) and the maps $\theta, p_*$ are isomorphisms. As we will see, the interesting map is $q_*$.

Since $p$ is a homotopy equivalence, the fundamental group of $G^k$ has rank $r$. Let $T$ be a maximal tree for $G^k$, and let $e_1, ..., e_r$ be the list of edges in $G^k \setminus T$ (each with its orientation coming from the labeling): these give a basis $[\sigma_1], ..., [\sigma_r]$ for the fundamental group $\pi_1(G^k, *)$, as defined in Proposition 1.1.13.

By Proposition 1.3.2, the map $q : G^k \to G^m$ is an isomorphism on the set of vertices (see figure 1.15); we consider the maximal tree $T$ for $G^k$, and we observe that $q\big|_T$ is an homeomorphism and $q(T)$ is a maximal tree for $G^m$. Let $d_1, ..., d_s$ be the list of edges in $G^m \setminus q(T)$ (each with its orientation coming from the labeling map): these give a basis $[\tau_1], ..., [\tau_s]$ for the fundamental group $\pi_1(G^m, *)$, according to Proposition 1.1.13.

The map $q_* : \pi_1(G^k, *) \to \pi_1(G^m, *)$ is now very easy to describe: we have

$$q_*([\sigma_i]) = \begin{cases} 1 & \text{if } q(e_i) \in q(T) \\ [\tau_j] & \text{if } q(e_i) = d_j \in G^m \setminus q(T) \end{cases}$$

For each $d_j \in G^m \setminus q(T)$ fix an index $i(j)$ such that $q(e_{i(j)}) = d_j$. Define the sets

$$N' = \{[\sigma_i] : q(e_i) \in q(T)\} \cup \{[\sigma_{i'}][\sigma_{i(j)}^{-1}] : q(e_{i'}) = d_j \text{ and } i' \neq i(j)\}$$

$$M' = \{[\sigma_{i(j)}] : j = 1, ..., s\}$$

and we observe that $M' \cup N'$ is a basis for $\pi_1(G^k, *)$ and that $q_*$ is injective on $\langle M' \rangle$ and zero on $\langle N' \rangle$. Thus we set $M = \theta^{-1}(p_*^{-1}(M'))$ and $N = \theta^{-1}(p_*^{-1}(N'))$ and this proves Theorem 1.3.3.

We observe that the construction of the graphs $G, G^k, G^m$ and of the maps $p, q$ is algorithmic. We use the bases $F_r = \langle b_1, ..., b_r \rangle$ and $\pi_1(G, *) = \langle [\gamma_1], ..., [\gamma_r] \rangle$ and $\pi_1(G^k, *) = \langle [\sigma_1], ..., [\sigma_r] \rangle$. The maps $\theta$ and $\theta^{-1}$ are of course explicit in those bases. The map $p_*$ can be algorithmically obtained too, because it is enough to look at the combinatorial paths $p \circ \gamma_i$ for $i = 1, ..., r$ and at when they cross the edges $e_1, ..., e_r$. In order to make explicit the map $p_*^{-1}$, we observe that each folding operation in the chain $G = G^0 \to ... \to G^k$ is a pointed homotopy equivalence, and it is easy to explicitly produce homotopy inverses $\alpha^i : G^i \to G^{i-1}$ for $i = 1, ..., k$, so that $p_*^{-1} = \alpha_*^1 \circ ... \circ \alpha_*^k$; for $i = 1, ..., r$, by looking at the path $\alpha^1 \circ ... \circ \alpha^k \circ \sigma_i$ we obtain a writing of $p_*^{-1}[\sigma_i]$ as a word in $[\gamma_1], ..., [\gamma_r]$. This shows that $M$ and $N$ can be built in an algorithmic way.

Each of $[\sigma_1], ..., [\sigma_r]$ can be written as a word of length at most 2 in the elements of $N'$. For $i = 1, ..., r$ let $l_i$ be the length of the word $\varphi(b_i)$ in the basis $a_1, ..., a_n$; the path $\gamma_i$ has length $l_i$, and so does the path $p \circ \gamma_i$; in particular $p \circ \gamma_i$ crosses the edges $e_1, ..., e_r$ at most $l_i$ times, and thus $p_*[\gamma_i]$ can be written as a word in $[\sigma_1], ..., [\sigma_r]$ of length at most $l_i$. Finally, each $b_i$ is a word of length one in $[\gamma_1], ..., [\gamma_r]$. This proves part (iv) of Theorem 1.3.4. □

### 1.3.3 Free groups are Hopfian

**Proposition 1.3.5** (Hopfian property)**.** *Let $\varphi : F_n \to F_n$ be a homomorphism from a finitely generated free group to itself. If $\varphi$ is surjective, then $\varphi$ is an isomorphism.*

*Proof.* By Theorem 1.3.3 we have a decomposition $F_n = F * F'$ with $\varphi\Big|_F$ injective and $\varphi\Big|_{F'}$ trivial. Since $F_n = F * F'$ we must have that $\text{rank}(F_n) = \text{rank}(F) + \text{rank}(F')$. Since $\varphi$ is surjective and $\varphi\Big|_{F'}$ trivial, we must have that $\varphi\Big|_F$ surjective. But we already knew that $\varphi\Big|_F$ was injective, and thus it must be an isomorphism, yielding that $\text{rank}(F) = \text{rank}(F_n)$. It follows that $\text{rank}(F') = 0$ and thus $\varphi = \varphi\Big|_{F'}$ is an isomorphism. □

For an alternative proof of Proposition 1.3.5, elementary and without any implicit use of folding, see Corollary 2.13.1 of [MKS04]. A very important corollary of Proposition 1.3.5 is the following:

**Corollary 1.3.6.** *Let $w_1, ..., w_n \in F_n$ be elements. If $w_1, ..., w_n$ generate all $F_n$ then $w_1, ..., w_n$ are a basis.*

*Proof.* Consider the homomorphism $\varphi : F_n \to F_n$ given by $\varphi(a_i) = w_i$ for $i = 1, ..., n$ and apply Proposition 1.3.5. □

## 1.4 Automorphisms of a free group

We here show that maximal trees can be pulled back along a rank-preserving folding operation $q : G \to G'$. Since a maximal tree gives a basis for the fundamental group of the graph $G$, this provides us two different bases for the same group $\pi_1(G) = \pi_1(G')$. These two bases aren't the same in general; instead, it turns out that they differ by an automorphism of a very specific kind, a *Whitehead automorphism*.

As every finite labeled graph can be written as a core graph, up to a finite sequence of folding operations, we show that every automorphism of $F_n$ can be written as a finite sequence of Whitehead automorphisms (up to substituting some elements of the basis with their inverses). This proves that Whitehead automorphisms form a finite set of generators for the group $\mathrm{Aut}(F_n)$.

### 1.4.1 Whitehead automorphisms

Let $F_n$ be a free group generated by $a_1, ..., a_n$.

**Definition 1.4.1.** *Let $a \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$ and let $A \subseteq \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\} \setminus \{a, \overline{a}\}$. Define the* ***Whitehead automorphism*** *$\varphi = (A, a)$ as the automorphism given by $a \mapsto a$ and*

$$a_j \mapsto \begin{cases} a_j & \text{if } a_j, \overline{a}_j \notin A \\ aa_j & \text{if } a_j \in A \text{ and } \overline{a}_j \notin A \\ a_j\overline{a} & \text{if } a_j \notin A \text{ and } \overline{a}_j \in A \\ aa_j\overline{a} & \text{if } a_j, \overline{a}_j \in A \end{cases}$$

*for $a_j \neq a, \overline{a}$.*

The letter $a$ will be called the **acting letter**, and the set $A$ will be the set of letters we **act on**; when we act on $a_j$, we can choose to multiply it by $a$ on the left or by $\overline{a}$ on the right (or both or none). Notice that our notation for Whitehead automorphisms is slightly different from the one you find in Lyndon and Schupp's book [LS01]: they choose to include the acting letter $a$ inside the set $A$, while we prefer not to do so.

In the literature, the word Whitehead automorphism also includes the automorphisms that induce a permutation of the set $\{a_1, \overline{a}_1, ..., a_n, \overline{a}_n\}$, i.e. automorphisms that permute the basis for the group (possibly inverting some of the generators). In order to avoid confusion, we refer to these automorphisms as *finite-order Whitehead automorphisms*, and to the ones of Definition 1.4.1 as *infinite-order Whitehead automorphisms*.

### 1.4.2 Rank-preserving folding and maximal trees

Let $G$ be a finite labeled graph and let $e_1, e_2$ be edges of $G$ with endpoints $v, v_1$ and $v, v_2$ respectively, and with the same label and orientation. Let $q : G \to G'$ be the folding operation that identifies $e_1$ and $e_2$. Suppose $q$ is rank-preserving (i.e. $v_1 \neq v_2$).

**Proposition 1.4.2** (Lifting maximal trees)**.** *In the above setting, suppose $T'$ is a maximal tree for $G'$. Then we have the following:*
*(i) If $q(e_1) = q(e_2)$ belongs to $T'$, then $T = q^{-1}(T')$ is a maximal tree for $G$.*
*(ii) If $q(e_1) = q(e_2)$ doesn't belong to $T'$, then exactly one between $q^{-1}(T') \setminus \{e_1\}$ and $q^{-1}(T') \setminus \{e_2\}$ is a maximal tree $T$ for $G$.*
*We say that $T$ is the **lifting** of $T'$ along the rank-preserving folding operation $q$.*

*Proof.* Recall that $\|\cdot\|_{\mathrm{e}}$ denotes the number of edges of a graph.
(i) Assume that $q(e_1) = q(e_2)$ belongs to $T'$. Let $T = q^{-1}(T')$. The rank-preserving folding operation $q : G \to G'$ induces a rank-preserving folding operation $q\big|_T : T \to T'$; it follows that $T$ is homotopy equivalent to $T'$, and in particular $T$ is a tree, and $\|T\|_{\mathrm{e}} = 1 + \|T'\|_{\mathrm{e}}$. By Lemma 1.1.14 we have

$$\|T\|_{\mathrm{e}} = 1 + \|T'\|_{\mathrm{e}} = 1 + \|G'\|_{\mathrm{e}} - \mathrm{rank}(G') = \|G\|_{\mathrm{e}} - \mathrm{rank}(G)$$

and thus $T$ is maximal.
(ii) Assume that $q(e_1) = q(e_2)$ doesn't belong to $T'$. Let $S = q^{-1}(T')$ and observe that $q$ induces an isomorphism $q\big|_S : S \to T'$, and in particular $S$ is a tree and $\|S\|_{\mathrm{e}} = \|T'\|_{\mathrm{e}}$.
Suppose $S$ contains two reduced paths $\sigma_1, \sigma_2$ connecting $v$ to $v_1, v_2$ respectively; then $q(\sigma_1), q(\sigma_2)$ are two distinct reduced paths in $T'$ connecting $q(v)$ to $q(v_1) = q(v_2)$, contradiction. Let $\sigma'$ be the unique reduced path in $T'$ going from $q(v)$ to $q(v_1) = q(v_2)$, and consider the lifting $\sigma$ in $S$: then $\sigma$ connects $v$ to exactly one of $v_1, v_2$, without loss of generality $v_2$.
Of course $S \cup \{e_2\}$ can't be a maximal tree for $G$, as it contains a loop given by $\sigma$ and $e_2$. Let $T = S \cup \{e_1\}$ and notice that $T$ is a tree, since $S$ is a tree and there is no path in $S$ connecting $v$ to $v_1$. By Lemma 1.1.14 we have

$$\|T\|_{\mathrm{e}} = 1 + \|S\|_{\mathrm{e}} = 1 + \|T'\|_{\mathrm{e}} = 1 + \|G'\|_{\mathrm{e}} - \mathrm{rank}(G') = \|G\|_{\mathrm{e}} - \mathrm{rank}(G)$$

and thus $T$ is maximal. □

By Proposition 1.1.13, choosing a maximal tree of a graph provides an identification of the fundamental group of the graph with a free group. We want to study how this identification changes when we perform a rank-preserving folding operation $q : G \to G'$.

Let $(G, *)$ be a finite connected labeled graph and let $q : G \to G'$ be a rank-preserving folding operation. Let $T'$ be a maximal tree for $G'$, and let $T$ be the maximal tree for $G$ obtained as lifting of $T'$ along $q$. Let $e_1, ..., e_n$ be the edges of $G \setminus T$, and notice that $G' \setminus T'$ consists of the $n$ pairwise distinct edges $q(e_1), ..., q(e_n)$.

Let $F = \langle x_1, ..., x_n \rangle$ be the free group with basis $x_1, ..., x_n$. Fix an arbitrary orientation on each $e_i$, inducing an orientation on $q(e_i)$, for $i = 1, ..., n$. Let $\varphi : F \to \pi_1(G, *)$ be the isomorphism that sends $x_i$ to the path that crosses $e_i$ exactly once, according to the chosen orientation, as in Proposition 1.1.13. Similarly, let $\varphi' : F \to \pi_1(G', *)$ be the isomorphism that sends $x_i$ to the path that crosses $q(e_i)$ exactly once, according to the induced orientation.

**Proposition 1.4.3.** *In the setting above, the map $(\varphi')^{-1} \circ q_* \circ \varphi : F \to F$ is a (infinite-order) Whitehead automorphism.*

*Proof.* If we fall in case (i) of Proposition 1.4.2, then it's immediate to see that $(\varphi')^{-1} \circ q_* \circ \varphi$ is the identity.

If we fall in case (ii) of Proposition 1.4.2, then the folding operation involves one edge $t$ of $T$, connecting vertices $v, v_1$ of $G$, and one edge $e_i$ of $G \setminus T$, for some $i \in \{1, ..., n\}$, connecting vertices $v, v_2$ of $G$. We assume that the chosen orientation on $e_i$ goes from $v$ to $v_2$, the other case being analogous. We also assume that the segment from the basepont to $v_1$ in $T$ contains $v$; the case where the segment from the basepoint to $v$ contains $v_1$ is analogous.

We set $a = x_i$ to be the acting letter of a Whitehead automorphism, and we want to define the set $A \subseteq \{x_1, ..., x_n, \overline{x}_1, ..., \overline{x}_n\} \setminus \{x_i, \overline{x}_i\}$ of letters we act on. For every $j = 1, ..., n$ with $j \neq i$, we consider the edge $e_j$ in $G$ and we call $u_{j,1}, u_{j,2}$ its endpoints, according to the orientation on $e_j$. If the unique reduced path from the basepoint to $u_{j,1}$ crosses the edge $t$ then we put $x_j$ in $A$, otherwise we don't; if the unique reduced path from the basepoint to $u_{j,2}$ crosses the edge $t$ then we put $\overline{x}_j$ in $A$, otherwise we don't.

This defines a Whitehead automorphism $\psi = (A, a)$ of $F$. A direct check on the paths in $G'$ proves that the two maps $\varphi' \circ \psi : F \to \pi_1(G', *)$ and $q_* \circ \varphi : F \to \pi_1(G', *)$ coincide. The conclusion follows. □

### 1.4.3 Generators for $\mathrm{Aut}(F_n)$

Let $F_n$ be a free group generated by $a_1, ..., a_n$.

**Theorem 1.4.4.** *The group $\mathrm{Aut}(F_n)$ is finitely generated. A finite set of generators is given by the Whitehead automorphisms.*

*Proof.* Let $\varphi : F_n \to F_n$ be an automorphism, and let $w_i$ be the reduced word representing $\varphi(a_i)$, for $i = 1, ..., n$. Let $G$ be the pointed labeled graph with one reduced path from the basepoint to itself for each of $w_1, ..., w_n$, as in Proposition

1.2.14. Consider a maximal folding sequence $G = G^0 \to ... \to G^m$ and observe that, since $\varphi$ is an automorphism, every operation in the folding sequence is rank-preserving and the last graph $G^m$ is the $n$-rose.

The $n$-rose $G^m$ has exactly one maximal tree $T^m$, given by the basepoint itself. Using Proposition 1.4.2 we can inductively define a maximal tree $T^i$ for $G^i$ for every $i = m, ..., 0$. The maximal tree $T^i$ provides a basis $w_1^i, ..., w_n^i$ for $\pi_1(G^i, *)$ and thus also for $F_n$. By Proposition 1.4.3 we have that the basis $w_1^i, ..., w_n^i$ differs from $w_1^{i+1}, ..., w_n^{i+1}$ by a Whitehead automorphism. But the basis $w_1^m, ..., w_n^m$ of the rose $G^m$ is just the standard basis $a_1, ..., a_n$, while the basis $w_1^0, ..., w_n^0$ for the graph $G$ is the basis $w_1, ..., w_n$, up to substituting some word with its inverse (since we can't be sure about the orientations).

Thus it is possible to pass from $a_1, ..., a_n$ to $w_1, ..., w_n$ by means of a sequence of Whitehead automorphisms (using only infinite-order Whitehead automorphisms, plus possibly a single finite-order Whitehead automorphism at the end of the sequence). The conclusion follows. □

We remark that infinite-order Whitehead automorphisms alone don't generated $\mathrm{Aut}(F_n)$. In fact, consider the abelianization map $\alpha : F_n \to \mathbb{Z}^n$ and observe that this induces a homomorphism $\alpha : \mathrm{Aut}(F_n) \to \mathrm{Aut}(\mathbb{Z}^n)$. Consider the homomorphism $D : \mathrm{Aut}(F_n) \to \mathbb{Z}/2\mathbb{Z}$ given by $D(\varphi) = \det(\alpha(\varphi)) \in \{1, -1\}$; the homomorphism $D$ is surjective and each infinite-order Whitehead automorphism belongs to the kernel of $D$. The same argument as in the proof of Theorem 1.4.4 yields that actually infinite-order Whitehead automorphisms generate all of $\ker D$, and thus the subgroup generated by infinite-order Whitehead automorphisms is an index-2 subgroup of $\mathrm{Aut}(F_n)$.

We point out that the group $\mathrm{Aut}(F_n)$ is also finitely presented, see Theorem 2.3.4.

# 2 | Primitive elements and Whitehead's algorithm

In this chapter we introduce the fundamental notion of primitive element in a free group and we provide two different algorithms to determine whether an element is primitive or not. The first algorithm, explained in Section 2.2, is the classical Whitehead's algorithm; in Sections 2.2, 2.3 and 2.4 we give an overview of three different proofs of the algorithm, which have been found over the years. The second algorithm, discussed in Section 2.5, has been proved more recently and has a completely different nature.

## 2.1 Primitive elements and free factors

Let $F_n$ be a free group with basis $a_1, ..., a_n$. There is a bijection between (ordered) bases for $F_n$ and automorphisms of $F_n$, the automorphism $\varphi : F_n \to F_n$ corresponding to the basis $\varphi(a_1), ..., \varphi(a_n)$. Thus, in order to understand the automorphism group $\mathrm{Aut}(F_n)$, it's fundamental to study the bases for $F_n$.

### 2.1.1 Primitive elements

Let $F_n$ be a free group generated by $a_1, ..., a_n$.

**Definition 2.1.1.** *An element $w \in F_n$ is called **primitive** if it is part of some basis.*

This means that there is an automorphism of $F_n$ that sends $a_1$ to $w$. Notice immediately that not every element is primitive, as the following examples illustrate:

*Example* 1. The element $a_1 a_2 \overline{a}_1 \overline{a}_2 \in F_n$ is not primitive. In fact, it belongs to the commutator subgroup $[F_n, F_n]$. The commutator subgroup is independent of the chosen basis, and since it doesn't contain any element of the standard basis, it can't contain any primitive element either.

*Example* 2. The element $a_1^2 \in F_n$ is not primitive. If $a_1^2, w_2, ..., w_n$ is a basis for $F_n$, then we look at the abelianization $\alpha : F_n \to \mathbb{Z}^n$. Since $a_1^2, w_2, ..., w_n$ generate $F_n$, we must have that $\alpha(a_1^2), \alpha(w_2), ..., \alpha(w_n)$ generate $\mathbb{Z}^n$. But $\alpha(a_1^2) = (2, 0, ..., 0)$ cannot be part of any set of $n$ generators for $\mathbb{Z}^n$.

*Example* 3. The element $a_1^2 a_2^3 \in F_2$ is not primitive. In fact, consider the surjective map $\alpha : F_2 \to S_3$ (the order-6 group of the permutations of three elements), where $a_1$ goes to any 2-cycle and $a_2$ goes to any 3-cycle. If $a_1^2 a_2^3, w$ is a basis for $F_2$, then $\alpha(a_1^2 a_2^3), \alpha(w)$ have to generate $S_3$. But $\alpha(a_1^2 a_2^3) = 1$ and $S_3$ can't be generated by just one element $\alpha(w)$.

In general, it can be useful to look at quotients of $F_n$; however, this doesn't provide an algorithmic criterion for determining whether an element is primitive or not.

### 2.1.2 Free factors

**Definition 2.1.2.** *A finitely generated subgroup $H \leq F_n$ is called **free factor** if any of the following equivalent conditions hold:*

*(i) There is a basis for $H$ which can be extended to a basis for $F$.*

*(ii) Every basis for $H$ can be extended to a basis for $F$.*

*(iii) There is a subgroup $H' \leq F$ such that $F = H * H'$.*

Notice that an element $w$ is primitive if and only if the subgroup $\langle w \rangle$ is a free factor. Thus the notion of free factor is a natural generalization of the notion of primitive element.

**Lemma 2.1.3.** *Let $(G, *)$ be a pointed graph and let $G'$ be a connected subgraph containing the basepoint $*$. Then the map $\pi_1(G', *) \to \pi_1(G, *)$ induced by the inclusion is injective, and $\pi_1(G', *)$ is a free factor in $\pi_1(G, *)$.*

*Proof.* Choose a maximal tree for $G'$ and extend it to a maximal tree for $G$. Apply Proposition 1.1.13 to obtain a basis for $\pi_1(G', *)$ together with an extension to a basis for $\pi_1(G, *)$. □

The following lemma is often useful when talking about free factors.

**Proposition 2.1.4.** *Let $K \leq F_n$ be a finitely generated subgroup, and let $H \leq F_n$ be a free factor. Then $H \cap K$ is a free factor in $K$.*

*Proof.* Suppose $H$ has rank $r$: without loss of generality, we can assume that $H = \langle a_1, ..., a_r \rangle \leq \langle a_1, ..., a_n \rangle = F_n$.
Let $G = \text{core}_*(K)$ be the basepointed labeled graph which represents $K$. A word $w$ belongs to $H \cap K$ if and only if it can be represented by a path inside $G$ which starts and ends at the basepoint, and which only crosses edges labeled with $a_1, ..., a_r$. Consider the subgraph $G' \subseteq G$ which is given by the union of the basepoint and of all such paths. Then $\pi_1(G', *)$ is exactly $H \cap K$.
But since $G'$ is a subgraph of $G$, we have that $\pi_1(G', *)$ is a free factor in $\pi_1(G, *)$, meaning that $H \cap K$ is a free factor in $K$, as desired. □

*Remark.* The above Proposition 2.1.4 can also be seen as a corollary of Kurosh subgroup theorem for free products. In particular, it remains true if we substitute the ambient free group $F_n$ with any group $G$ that decomposes as a free product $G = H * H'$.

The following two corollaries show how, when talking about primitive elements and free factors, it often makes sense to omit mention of the ambient group.

**Corollary 2.1.5.** *For $k < n$, consider the inclusion $F_k = \langle a_1, ..., a_k \rangle \leq \langle a_1, ..., a_n \rangle = F_n$, and let $w \in F_k$. Then $w$ is primitive in $F_k$ if and only if $w$ is primitive in $F_n$.*

**Corollary 2.1.6.** *For $k < n$, consider the inclusion $F_k = \langle a_1, ..., a_k \rangle \leq \langle a_1, ..., a_n \rangle = F_n$, and let $H \leq F_k$. Then $H$ is a free factor in $F_k$ if and only if $H$ is a free factor in $F_n$.*

Free factors behave well with respect to intersection.

**Corollary 2.1.7.** *Let $H, H' \leq F_n$ be free factors. Then $H \cap H'$ is a free factor in $F_n$.*

## 2.2 Whitehead's algorithm

It's not immediate, given an element of a finitely generated free group, that one can determine whether it is primitive or not. The first algorithm for this purpose was found by J.H.C. Whitehead in 1936.

The algorithm is quite simple: we take a word, and we apply to it all the possible Whitehead automorphisms (see Definition 1.4.1), trying to reduce its cyclic length; if we are able to do so, then we go on, until we eventually get a generator of $F_n$; if at some point we are forced to stop before getting a generator, then it means that the word isn't primitive.

Whitehead's original proof is based on the construction of a suitable 3-manifold, and on surgery suitably performed on certain paths and surfaces inside this 3-manifold, see [Whi36a]. We explain the details of this proof in Sections 2.2.3 and 2.2.4.

In this section we decided to provide a proof in the spirit of Whitehead's original argument, mainly for historical reasons; however, in the subsequent Sections 2.3 and 2.4 we are going to provide two other independent arguments, which are both more flexible that Whitehead's original one. Especially, the proof provided in Section 2.4 is remarkably short.

### 2.2.1 Whitehead graph and cut vertices

**Definition 2.2.1.** *Let $w$ be a cyclically reduced word. Define the **Whitehead graph** of $w$ as follows:*

*(i) We have $2n$ vertices labeled $a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n$.*

*(ii) For every pair of consecutive letters in $w$, we draw an (unoriented) arc from the inverse of the first letter to the second. We also draw an arc connecting the inverse of the last letter of $w$ to the first letter of $w$, as if they were adjacent.*

Notice that, $w$ being cyclically reduced, we never have any arc connecting a vertex to itself.

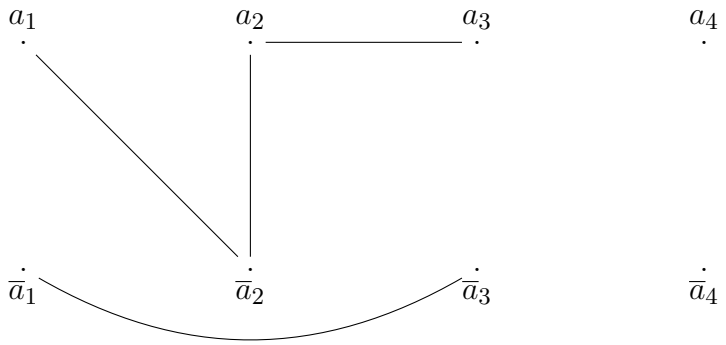


Figure 2.1: The Whitehead graph of $w = a_1a_2a_2\overline{a}_3$ in the free group $F_4 = \langle a_1, a_2, a_3, a_4\rangle$.

**Definition 2.2.2.** *Let $w$ be a cyclically reduced word. A vertex $a$ in the Whitehead graph of $w$ is called a **cut vertex** if it is non-isolated and at least one of the following two configurations happens:*

*(i) The connected component of $a$ doesn't contain $\overline{a}$.*

*(ii) The connected component of $a$ becomes disconnected if we remove $a$.*

A cut vertex in the Whitehead graph has a fundamental consequence: it provides us a Whitehead automorphism which makes the word shorter. In the proof of the following statement, it is put in evidence how the arcs of the Whitehead graph really represent letters appearing and canceling inside the word when we act with the automorphism. Recall that $|w|$ is the length of $w$ and $|w|_{\text{cyc}}$ is the length of the cyclic reduction of $w$.

**Theorem 2.2.3.** *Let $w$ be cyclically reduced, primitive and not a single letter. Suppose $a \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$ is a cut vertex for the Whitehead graph of $w$. Then there is a Whitehead automorphism $\varphi = (A, a)$ such that $|\varphi(w)|_{\text{cyc}} < |w|_{\text{cyc}}$.*

*Proof.* Let $w$ be cyclically reduced, and let $a$ be a cut vertex in its Whitehead graph. If the connected component of $a$ doesn't contain $\overline{a}$, then we take the set $A$ to be that connected component (excluding $a$ itself). Otherwise, take the connected component of $a$ and remove $a$ itself: we are left with at least two nonempty connected components, and at least one of these components doesn't contain $\overline{a}$; take $A$ to

be such a component. In both cases we consider the Whitehead automorphism $\varphi = (A, a)$. We look at what happens between two consecutive non-$a$ non-$\overline{a}$ letters in $w$ when we apply the automorphism $\varphi$.

Suppose we have two consecutive letters $w = ...bc...$ with $b, c \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\} \setminus \{a, \overline{a}\}$. Then we have an arc from $\overline{b}$ to $c$ in the Whitehead graph, so we are either acting on neither of $\overline{b}, c$ or on both of them. If we are not acting on them, then $\varphi(w) = ...bc...$ and the word is not affected between $b$ and $c$. If we are acting on both of them, then $\varphi(w) = ...b\overline{a}ac... = ...bc...$ and every $a$ and $\overline{a}$ which appears there immediately cancels, and again the word is not affected between $b$ and $c$.

Suppose we have in $w$ a segment of the form $w = ...ba^kc...$ with $b, c \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\} \setminus \{a, \overline{a}\}$ and $k \geq 1$ (the case $k \leq -1$ is analogous). This means that we have an arc from $\overline{a}$ to $c$, and thus we are not acting on $c$. If we are not acting on $\overline{b}$, then $\varphi(w) = ...ba^kc...$ and the word is not affected between $b$ and $c$. If we are acting on $\overline{b}$, then $\varphi(w) = ...b\overline{a}a^kc... = ...ba^{k-1}c...$ and the number of $a$ letters strictly decreases between $b$ and $c$.

To conclude we notice that, since there is at least one arc between $a$ and a vertex of $A$, at least one cancellation takes place, giving a (strict) decrease in the cyclic length of $w$, yielding Theorem 2.2.3. □

*Remark* 1. We observe that the argument of the proof of the above Theorem 2.2.3 yields also a slightly stronger result: every letter $a$ or $\overline{a}$ that appears, when applying the Whitehead automorphism $\varphi$ letter by letter to the word $w$, immediately cancels in the reduction process.

### 2.2.2 Whitehead's algorithm

The main theorem of Whitehead's paper [Whi36a] is the following:

**Theorem 2.2.4** (Whitehead, [Whi36a])**.** *Let $w$ be a cyclically reduced primitive word. Then the Whitehead graph of $w$ has a cut vertex.*

The above Theorem 2.2.4, together with Theorem 2.2.3, allows one to immediately deduce the following consequence.

**Theorem 2.2.5.** *Let $w$ be a cyclically reduced primitive word of length greater than 1. Then there is a Whitehead automorphism $\varphi$ such that $|\varphi(w)|_{\text{cyc}} < |w|_{\text{cyc}}$.*

As a corollary of Theorem 2.2.5 we have an algorithm that, given a word $w \in F_n$, determines whether the word is primitive or not. The algorithm runs as follows: we take our word $w$ and we look for a Whitehead automorphism reducing its cyclic length. If we find one, then we apply it to the words, and we reiterate the procedure on the new shorter word. If we obtain a word which is a single letter, then $w$ is primitive, as we have found a chain of Whitehead automorphisms whose composition

brings $w$ to a single letter. If at some point we aren't able to reduce the length of $w$ anymore with a Whitehead automorphism, and we don't have a single letter, then Theorem 2.2.5 implies that $w$ isn't primitive.

In the following Sections 2.2.3 and 2.2.4 we prove Theorem 2.2.5; we do this in line with the argument of Whitehead's original paper [Whi36a].

### 2.2.3 The fundamental group of $(S^2 \times S^1)^{\#n}$

We now explain how to reinterpret elements of $F_n$ and their Whitehead graphs as suitable paths inside a suitably chosen 3-manifold.

Let $M$ be a 3-manifold (with boundary). Take a disk $D^3$ embedded in $M$ and remove the interior of the disk: we get a new 3-manifold $N$ with one boundary component more than $M$ (which is the sphere $\partial D^3$). Such an operation will be called **digging a hole** in $M$.

Now we start with $S^3$. We dig two holes, with boundary two spheres $C^+$ and $C^-$. We glue $C^+$ and $C^-$ together to get a 3-manifold without boundary: if the orientation of the gluing map is chosen suitably, the result is diffeomorphic to $S^2 \times S^1$. In the same way, we can start with $S^3$ and dig $2n$ holes, with boundaries $C_1^+, ..., C_n^+, C_1^-, ..., C_n^-$. We then glue $C_i^+$ to $C_i^-$ as above. We get a 3-manifold $M$ without boundary, which is diffeomorphic to $(S^2 \times S^1)\#...\#(S^2 \times S^1)$, the connected sum of $n$ copies of $S^2 \times S^1$.

**Definition 2.2.6.** *A **basis** for $(S^2 \times S^1)^{\#n}$ is given by*
*(i) $n$ pairwise disjoint embedded spheres $\Gamma_1, ..., \Gamma_n$;*
*(ii) for each $\Gamma_i$ an orientation on the normal bundle $\nu(\Gamma_i)$;*
*such that, when we cut along the $\Gamma_i$s, we get $S^3$ with $2n$ holes.*

When we have a basis $\Gamma_1, ..., \Gamma_n$ and we cut along the $\Gamma_i$, we get $S^3$ with $2n$ holes. We call the boundaries of those holes $C_1^+, C_1^-, ..., C_n^+, C_n^-$, where $C_i^+, C_i^-$ are the boundary components obtained cutting along $\Gamma_i$, and the $+$ and $-$ are put according with the chosen orientation on $\nu(\Gamma_i)$.

The fundamental group of $(S^2 \times S^1)^{\#n}$ is isomorphic to $F_n$. A basis $\Gamma_1, ..., \Gamma_n$ gives us an explicit isomorphism as follows: the element $a_i$ of the basis of $F_n$ corresponds to any path $\alpha_i$ which goes from the basepoint to $C_i^-$, then crosses $\Gamma_i$ transversely in one point, and then goes back from $C_i^+$ to the basepoint. Thus $\alpha_i$ has exactly one intersection with $\Gamma_i$, and no intersection with $\Gamma_j$ for $j \neq i$.

Take a path $\alpha$, corresponding to a word $w$, which is transverse to the basis $\Gamma_1, ..., \Gamma_n$. Then we can cut along $\Gamma_1, ..., \Gamma_n$ in order to get $S^3$ with $2n$ holes, with boundaries $C_1^+, C_1^-, ..., C_n^+, C_n^-$. The curve $\alpha$ becomes a collection of arcs. We consider $C_i^+$ as a vertex labeled $a_i$, and $C_i^-$ as a vertex labeled $a_i^-$; we consider the arcs of $\alpha$ as edges connecting a vertex to another. We get a graph, which turns out to be exactly the Whitehead graph of $w$; this is made precise with the following lemma:

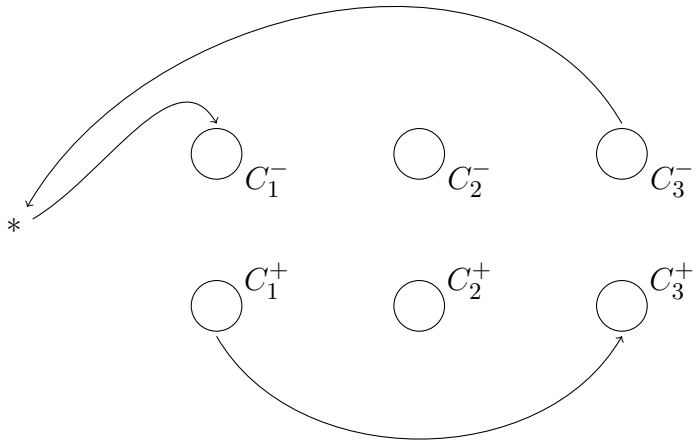


Figure 2.2: A path representing the word $w = \overline{a}_1 a_3$ in the free group $F_3 = \langle a_1, a_2, a_3 \rangle$.

**Lemma 2.2.7.** *Let $w \in F_n$ be cyclically reduced. Then $w$ can be represented by a path $\alpha$ (injective and transverse to each $\Gamma_i$) such that $\alpha$ never intersects the same $\Gamma_i$ two consecutive times in opposite direction. Moreover, if we take any such representative $\alpha$, and we cut along the basis $\Gamma_1, ..., \Gamma_n$, then we get exactly the Whitehead graph of $w$.*

*Proof.* Immediate from the definitions. □

Suppose we are given $(S^2 \times S^1)^{\#n}$ and a basis $\Gamma_1, ..., \Gamma_n$. Suppose also we are given a path $\theta : [0, 1] \to (S^2 \times S^1)^{\#n}$ such that $\theta$ has no intersection with any $\Gamma_i$ except at the two endpoints $\theta(0) \in \Gamma_1$ and $\theta(1) \in \Gamma_2$ and such that, in $\mathbb{R}^n$ with $2n$ holes, the path $\theta$ connects $C_1^+$ with $C_2^+$. Consider a small cylinder which goes along $\theta$; take the sphere $\Gamma_1$, cut a small hole near $\theta(0)$, and glue $\Gamma_1$ to the cylinder along that hole. Repeat the same procedure for $\Gamma_2$ and the other end of the cylinder. In the end we get a new sphere $\Gamma_1'$, given by $\Gamma_1$ plus the cylinder plus $\Gamma_2$, as in figure 2.3. Up to a small perturbation, we can make $\Gamma_1'$ to be disjoint from $\Gamma_1$ and $\Gamma_2$. It is immediate to prove the following fundamental result:

**Lemma 2.2.8.** *The family $\Gamma_1', \Gamma_2, \Gamma_3, ..., \Gamma_n$ is a basis for $M$. Moreover, let $\varphi : F_n \to F_n$ be the automorphism given by $a_2 \mapsto a_1 a_2$ and $a_i \mapsto a_i$ for $i \neq 2$. If $\alpha$ represents a word $w$ in the basis $\Gamma_1, ..., \Gamma_n$, then $\alpha$ represents $\varphi(w)$ in the basis $\Gamma_1', ..., \Gamma_n$.*

*Proof.* Immediate from the definitions. □

**Lemma 2.2.9.** *Let $\varphi : F_n \to F_n$ be an automorphism and let $\Gamma_1, ..., \Gamma_n$ be a basis for $(S^2 \times S^2)^{\#n}$. Then there is a basis $\Gamma_1', ..., \Gamma_n'$ such that, if the path $\alpha$ in the basis*

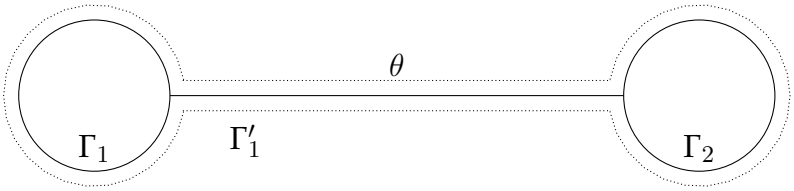


Figure 2.3: The construction of Lemma 2.2.8.

*$\Gamma_1, ..., \Gamma_n$ represents an element $w \in F_n$, then the same path $\alpha$ in the basis $\Gamma_1', ..., \Gamma_n'$ represents the element $\varphi(w) \in F_n$.*

*Proof.* By reiterating the construction of the above Lemma 2.2.8, we can obtain all Whitehead automorphisms. By Theorem 1.4.4, by reiterating the same construction we can obtain all automorphisms of $F_n$, as desired. □

### 2.2.4 Geometric proof of Whitehead's algorithm

This section is dedicated to the proof of Theorem 2.2.4. Let $w \in F_n \setminus \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$ be a primitive element; we want to prove that its Whitehead graph contains a cut vertex.

We start with the manifold $(S^2 \times S^1)^{\#n}$ and a basis $\Lambda_1, ...\Lambda_n$. We take a path $\alpha$ intersecting only $\Lambda_1$ once and transversely, so that $\alpha$ represents $a_1$. We also take a sphere $\Sigma$ which intersects $\alpha$ in exactly one point, and which is transverse to $\Lambda_1, ..., \Lambda_n$; for example, we may take $\Sigma$ to be a perturbation of $\Lambda_1$. Since $w$ is primitive, there is an automorphism $\varphi : F_n \to F_n$ with $\varphi(a_1) = w$. By Lemma 2.2.9 we can construct a basis $\Gamma_1, ..., \Gamma_n$ for $(S^2 \times S^1)^{\#n}$ such that $\alpha$ represents $w$ in this new basis. We can also assume that the new basis $\Gamma_1, ..., \Gamma_n$ is transverse to both $\alpha$ and $\Sigma$.

The intersection $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ consists of a (possibly empty) collection of simple pairwise disjoint curves $\sigma_1, ..., \sigma_m$. Each of these curves is *separating* on $\Sigma$, meaning that if we cut $\Sigma$ along a given $\sigma_j$, we get a disconnected surface.

In what follows, we will keep fixed the basis $\Gamma_1, ..., \Gamma_n$, and we will change the curve $\alpha$, but only by means of homotopies. However, we will need to modify the surface $\Sigma$ in a substantial way; we thus drop the assumption that $\Sigma$ is a sphere, allowing $\Sigma$ to be any closed orientable surface.

**Definition 2.2.10.** *A surface $\Sigma$ is called **suitable** for $\alpha$ if the following holds:*

*(i) $\Sigma$ is a closed orientable surface embedded in $M$.*

*(ii) $\Sigma$ is transverse to $\alpha$ and to $\Gamma_1, ..., \Gamma_n$.*

*(iii) $\Sigma$ intersects $\alpha$ in exactly one point, which doesn't belong to $\Gamma_1 \cup ... \cup \Gamma_n$.*

*(iv) The intersection $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ consists of a (possibly empty) collection of pairwise disjoint simple closed curves $\sigma_1, ..., \sigma_m$, such that each of these curves is*

*separating on* $\Sigma$.

**Lemma 2.2.11.** *Suppose that* $\alpha$ *never crosses the same* $\Gamma_i$ *two consecutive times in opposite direction. Suppose there is a suitable surface* $\Sigma$ *for* $\alpha$ *which is disjoint from* $\Gamma_1, ..., \Gamma_n$. *Then the Whitehead graph if* $w$ *has a cut vertex.*

*Proof.* Cut along $\Gamma_1, ..., \Gamma_n$ in order to get $S^3$ with $2n$ holes $C_1^+, C_1^-, ..., C_n^+, C_n^-$. The path $\alpha$ becomes the Whitehead graph of $w$, in virtue of lemma 2.2.7. The manifold $S^3$ with $2n$ holes is separated by $\Sigma$ into two connected components, which we call *interior* and *exterior* in some order. We know that we have a unique intersection point between $\alpha$ and $\Sigma$, and assume this on an arc of $\alpha$ going from $C$ to $C'$, with $C, C' \in \{C_1^+, C_1^-, ..., C_n^+, C_n^-\}$.
If $C$ has degree at least 2 in the Whitehead graph, then $C$ is a cut vertex (since it separates the interior of $\Sigma$ from the exterior). In the same way, if $C'$ has degree at least 2, then $C'$ is a cut vertex. If both $C$ and $C'$ have degree 1 and $C, C'$ are not inverses of each other, then both $C$ and $C'$ are cut vertices. If $C = C_i^+$ and $C' = C_i^-$ and they both have degree 1, then this forces $w = a_i$ or $w = \overline{a}_i$ (since adjacent to $a_i$ we can only have the letter is $a_i$ itself, on both sides). □

We now describe how to remove the intersection between $\Sigma$ and $\Gamma_1, ..., \Gamma_n$.

**Lemma 2.2.12.** *Suppose that* $\alpha$ *never crosses the same* $\Gamma_i$ *two consecutive times in opposite direction. Suppose* $\Sigma$ *is a suitable surface for* $\alpha$ *such that* $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ *is a collection of* $m \geq 1$ *simple closed curves. Suppose one of these curves* $\sigma$ *bounds on* $\Gamma_1$ *two disks* $D, D'$, *and one of those disks doesn't intersect* $\alpha$. *Then we can find a suitable surface* $\Sigma'$ *whose intersection with* $(\Gamma_1 \cup ... \cup \Gamma_n)$ *consists of* $m' < m$ *curves.*

*Proof.* Let's take the curve $\sigma$ in $\Sigma \cap \Gamma_1$, bounding two disks $D, D'$ on $\Gamma_1$. Suppose $D'$ doesn't intersect $\alpha$. We look at the curves in the intersection $\Sigma \cap D'$, and we take a curve which is innermost in $D'$, let's say $\sigma'$, bounding a smaller disk $D''$. We cut $\Sigma$ along $\sigma'$ to get two surfaces $\Sigma_0, \Sigma_1$, where $\Sigma_0$ contains the unique point of intersection $\Sigma \cap \alpha$.
We substitute the surface $\Sigma$ with the surface $\Sigma' = \Sigma_0 \cup D''$. The new surface $\Sigma'$ has exactly one point of intersection with $\alpha$. Also, we can push $D''$ slightly on one side of $\Gamma_1$, in order to eliminate the disk of intersection $D''$ from $\Sigma' \cap \Gamma_1$. We thus obtain a suitable surface $\Sigma'$ with strictly fewer curves in the intersection with $\Gamma_1 \cup ... \cup \Gamma_n$. □

**Lemma 2.2.13.** *Suppose that* $\alpha$ *never crosses the same* $\Gamma_i$ *two consecutive times in opposite direction. Suppose* $\Sigma$ *is a suitable surface for* $\alpha$ *such that* $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ *is a collection of* $m \geq 1$ *simple closed curves* $\sigma_1, ..., \sigma_m$. *A curve* $\sigma_j$ *belonging to* $\Gamma_i$ *will separate* $\Gamma_i$ *in two disks* $D, D'$: *suppose for every* $j = 1, ..., m$, *both the disks* $D$ *and* $D'$ *intersect* $\alpha$. *Then the Whitehead graph of* $w$ *has a cut vertex.*

*Proof.* Consider the curves $\sigma_1, ..., \sigma_m$ on $\Sigma$: by means of an innermost curve argument, we can assume that $\sigma_1$ separates $\Sigma$ in two surfaces $\Sigma_0, \Sigma_1$, and that $\Sigma_1$ doesn't contain any other $\sigma_j$, and doesn't contain the point of intersection with $\Sigma \cap \alpha$.

Cut along $\Gamma_1, ..., \Gamma_n$, in order to get $S^3$ with $2n$ holes $C_1^+, C_1^-, ..., C_n^+, C_n^-$. The surface $\Sigma_1$ becomes a surface inside $S^3$. Moreover $\Sigma_1$ has just one boundary component, given by (one of the two images of) $\sigma_1$; we can assume, without loss of generality, that such boundary component belongs to $C_1^+$.

The manifold $S^3$ with $2n$ holes is separated by $\Sigma_1$ into two connected components, which we call *interior* and *exterior* in some order. Moreover, from the hypothesis of the lemma, there is at least one arc of $\alpha$ going from $C_1^+$ to the interior and one going from $C_1^+$ to the exterior. But this means that if we remove the vertex $a_1$ from the Whitehead graph of $w$, the connected component of $a_1$ becomes disconnected. Thus, $a_1$ is a cut vertex, as desired. □

We finally describe how to modify $\alpha$ and $\Sigma$ in order to remain with a curve that never intersects any of $\Gamma_1, ..., \Gamma_n$ two consecutive times with opposite direction.

**Lemma 2.2.14.** *Suppose that $\alpha$ crosses $\Gamma_1$ two consecutive times, and in opposite directions; suppose also $\alpha$ has a suitable surface $\Sigma$. Then there is a path $\alpha'$ homotopic to $\alpha$, transverse to $\Gamma_1, ..., \Gamma_n$, with strictly less points of intersection with $\Gamma_1 \cup ... \cup \Gamma_n$, and with a suitable surface $\Sigma'$.*

*Proof.* We assume that, after cutting along $\Gamma_1, ..., \Gamma_n$, the path $\alpha$ has an arc $\overline{pq}$ going from $C_1^+$ to itself. For simplicity, we assume that the unique point of intersection $\alpha \cap \Sigma$ doesn't belong to $\overline{pq}$, the other case being similar.

Let's say the intersection $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ consists of $m$ pairwise disjoint simple closed curves $\sigma_1, ..., \sigma_m$. We cut along $\Gamma_1, ..., \Gamma_n$ in order to obtain $S^3$ with $2n$ holes. Each of $\sigma_1, ..., \sigma_m$ has two images after the cutting operation, and thus we end up with a collection of $2m$ pairwise disjoint simple closed curves $s_1, ..., s_{2m}$ belonging to $C_1^+ \cup C_1^- \cup ... \cup C_n^+ \cup C_n^-$. The surface $\Sigma$ is cut along $\sigma_1, ..., \sigma_m$, and becomes a union of $m+1$ pairwise disjoint surfaces with boundary $S_1, ..., S_{m+1}$ inside $S^3$ with $2n$ holes; each $S_i$ separates $S^3$ with $2n$ holes into two connected components.

Among $s_1, ..., s_{2m}$, we consider only the curves that belong to $C_1^+$ and separate $p$ from $q$, let's say they are $s_1, ..., s_k$ for some $0 \leq k \leq 2m$.

CASE $k = 0$. Suppose there is no curve separating $p$ and $q$ on $C_1^+$. Then we can substitute the arc $\overline{pq}$ of $\alpha$ with an arc belonging to $\Gamma_1$ and not intersecting $\Sigma$. We then push that new arc through $\Gamma_1$ in order to eliminate the two points $p, q$ from the intersection $\alpha \cap (\Gamma_1 \cup ... \cup \Gamma_n)$. We don't change the surface $\Sigma$, and we are done.

CASE $k = 1$. Suppose there is just one curve $s_1$ on $C_1^+$ separating $p$ and $q$. Then $s_1$ must belong to $S_j$ for some $j \in \{1, ..., m+1\}$. We notice that $S_j$ separates $S^3$ with $2n$ holes into two connected components, and that $p, q$ belong to different components, and the arc $\overline{pq}$ doesn't intersect $S_j$, contradiction.

CASE $k = 2$. Suppose there are exactly two curves $s_1, s_2$ on $C_1^+$ separating $p$ and $q$. If $s_1$ belongs to a surface $S_j$, then $s_2$ has to belong to the same surface $S_j$, since $S_j$ separates $S^3$ with $2n$ holes into two connected components. Let's say the curves $s_1, s_2$ on $S_j$ correspond to two curves $\sigma_1, \sigma_2$ on $\Sigma$: then this means that $\sigma_1, \sigma_2$ belong to the same connected component of $\Sigma \setminus (\sigma_3 \cup ... \cup \sigma_m)$.

We pick an arc $\gamma$ on $\Gamma_1$ that goes from $p$ to $q$ and intersects $\Sigma$ in exactly two points $p', q'$ belonging to $\sigma_1$ and $\sigma_2$ respectively. We substitute the arc $\overline{pq}$ of $\alpha$ with the arc $\gamma$, in order to obtain a curve $\alpha'$ homotopic to $\alpha$. However this creates new intersections with $\Sigma$. Thus we change $\Sigma$: we cut two small disks away from $\Sigma$, in correspondence of the two points $p', q'$, and then we add to $\Sigma$ a small "tunnel" which follows the arc $\overline{p'q'}$. This defines a surface $\Sigma'$ with is transverse to $\alpha, \Gamma_1, ..., \Gamma_n$ and which intersects $\alpha$ in exactly one point.

We need to check that all the simple closed curves in the intersection $\Sigma' \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ are separating. Recall that $\sigma_1, \sigma_2$ belong to the same connected component of $\Sigma \setminus (\sigma_3 \cup ... \cup \sigma_m)$: the "tunnel" that we added is a handle that got attached on $\Sigma$ with both ends in that connected component. Thus the curves $\sigma_3, ..., \sigma_m$ remain separating on $\Sigma'$. Finally, the two curves $\sigma_1, \sigma_2$ on $\Sigma$ give rise to a single curve $\overline{\sigma}$ on $\Sigma'$, and for how the tunnel is glued, it can be seen that this single curve is separating. The conclusion follows.

CASE $k \geq 3$. We proceed in a way similar to the case $k = 2$ above. Suppose there are curves $s_1, ..., s_k$ on $C_1^+$ separating $p$ and $q$, and suppose $s_1, ..., s_k$ are encountered in this order when going from $p$ to $q$ in $C_1^+$. We claim that $k$ must be even and it's possible to partition the curves $s_1, ..., s_k$ into pairs in such a way that:

(i) If $s_{i_1}, s_{i_2}$ are a pair, then they belong to the same $S_j$.

(ii) If $s_{i_1}, s_{i_2}$ are a pair and $s_{i_3}, s_{i_4}$ are a pair, and if $i_1 < i_3 < i_2$, then $i_1 < i_4 < i_2$.

Imagine we travel along $C_1^+$ from $p$ to $q$, in such a way that we cross $s_1, ..., s_k$ each once and in this order. Since each surface $S_j$ disconnects $S^3$ with $2n$ holes, we must have that each $S_j$ is crossed an even number of times: this immediately shows that $k$ is even and a partition into couples is possible, satisfying property (i). Given two surfaces $S_j, S_{j'}$, we have that $S_j$ separates $S^3$ with $2n$ holes into two connected components, and $S_{j'}$ must live in one of them. This extra property allows us to produce a partition satisfying property (ii) too.

Let's say the curves $s_1, ..., s_k$ correspond to curves $\sigma_1, ..., \sigma_k$ on $\Sigma$ (which are thus partitioned into pairs too): then this means that, if $\sigma_{i_1}, \sigma_{i_2}$ are a pair, then $\sigma_{i_1}, \sigma_{i_2}$ are in the same connected component of $\Sigma$ minus the others.

We pick an arc $\gamma$ on $\Gamma_1$ that goes from $p$ to $q$ and intersects $\Sigma$ in each of $\sigma_1, ..., \sigma_k$ exactly once and in this order. We substitute the arc $\overline{pq}$ of $\alpha$ with the arc $\gamma$, in order to obtain a curve $\alpha'$ homotopic to $\alpha$. This creates new intersections with $\Sigma$, thus we have to change $\Sigma$. For each pair $\sigma_{i_1}, \sigma_{i_2}$ we consider the two points of intersection if $\gamma$ with $\sigma_{i_1}$ and $\sigma_{i_2}$, and we remove two small disks from $\Sigma$ and add a "tunnel". We have to repeat this procedure for all the pairs in the partition, and property (ii)

of the partition allows us to make the tunnels nested, in such a way that allows us to define a non-self-intersecting new surface $\Sigma'$, given by $\Sigma$ with $k$ disks and $k/2$ tunnels added. This defines a surface $\Sigma'$ with is transverse to $\alpha, \Gamma_1, ..., \Gamma_n$ and which intersects $\alpha$ in exactly one point.

In the same way as in the case $k = 2$, it's easy to prove that each pair of curves $\sigma_{i_1}, \sigma i_2$ in $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ becomes a single separating curve in $\Sigma' \cap (\Gamma_1 \cup ... \cup \Gamma_n)$. The others curves in the intersection $\Sigma \cap (\Gamma_1 \cup ... \cup \Gamma_n)$ remain exactly the same in $\Sigma' \cap (\Gamma_1 \cup ... \cup \Gamma_n)$, and they also remain separating. The conclusion follows. □

We are finally ready to complete the proof of Theorem 2.2.4. As pointed out at the beginning of this Section 2.2.4, we have a curve $\alpha$ representing our word $w$ in the basis $\Gamma_1, ..., \Gamma_k$, and a surface $\Sigma$ suitable for $\alpha$. We apply repeatedly Lemma 2.2.14 in order to modify the curve $\alpha$ and the surface $\Sigma$, until $\alpha$ really becomes the Whitehead graph of $w$, according to Lemma 2.2.7. We then apply repeatedly Lemma 2.2.12 in order to reduce the number of intersections of $\Sigma$ with the basis $\Gamma_1, ..., \Gamma_n$: in this procedure, we either fall into the hypothesis of Lemma 2.2.13 at some point (in which case we are done), or we can continue the procedure until $\Sigma$ has no intersection with the basis $\Gamma_1, ..., \Gamma_n$ (in which case we are done by Lemma 2.2.11). This concludes the proof of Theorem 2.2.4.

## 2.3 Peak reduction

In this section we provide another independent proof of Theorem 2.2.5, based on the work of E. Rapaport, see [Rap58]. This technique turned out to be more flexible than the previous argument, and led to the proof of several other results and generalizations of Whitehead's algorithm; we outline some of these applications.

**Lemma 2.3.1** (Peak reduction, see [LS01])**.** *Let $u \in F_n$ and let $\sigma, \tau$ be Whitehead automorphisms. Suppose $|u|_{\text{cyc}} \leq |\sigma(u)|_{\text{cyc}} > |\tau(\sigma(u))|_{\text{cyc}}$. Then there are Whitehead automorphisms $\rho_1, ..., \rho_k$ for some $1 \leq k \leq 4$ such that*

*(i) $\rho_k \circ ... \circ \rho_1 = \tau \circ \sigma$.*

*(ii) For $i = 1, ..., k$ we have $|\rho_i \circ ... \circ \rho_1(u)|_{\text{cyc}} < |\sigma(u)|_{\text{cyc}}$.*

The proof of the lemma is a long and tedious list of cases, which can be found in [LS01] Lemma 4.18. In each case, one constructs by hands the chain $\rho_1, ..., \rho_k$ in such a way that at each step we get a word shorter than $\sigma(u)$. Given the lemma, the proof of Whitehead's algorithm is quite straightforward.

*Proof of Theorem 2.2.5.* Suppose $w$ is primitive. Then $w$ can be turned into a single letter $a_1$ by a finite chain of Whitehead automorphisms (we know this is possible because of Theorem 1.4.4), let's say $a_1 = \sigma_k ... \sigma_1(w)$.

We write in sequence the cyclic reductions of the words $w, \sigma_1(w), \sigma_2\sigma_1(w), ..., \sigma_k...\sigma_1(w)$. We take a word of the sequence with maximum cyclic length; if there are several words of maximum length, then we choose one which is adjacent to one which has strictly smaller cyclic length. We apply Lemma 2.3.1 to that word and to the two adjacent words, in order to modify the chain of words. The number of words in the list can increase, but the number of words of maximum cyclic length always decreases.

When the last word of maximum cyclic length disappears from the sequence, the maximum cyclic length of the sequence decreases. We then reiterate the procedure. We stop when the only word of maximum cyclic length is $w$; this means the first Whitehead automorphism of the chain shortens $w$, as required. □

Notice that Lemma 2.3.1 doesn't require the word to be primitive. In particular, it gives an algorithm to determine whether any two given elements of $F_n$ are in the same orbit under the action of $\mathrm{Aut}(F_n)$. We also remark that such an algorithm had already been provided by Whitehead in [Whi36b].

**Theorem 2.3.2.** *Let $u, v \in F_n$ and suppose there is an automorphism of $F_n$ sending $u$ to $v$. Then there are Whitehead automorphisms $\rho_1, ..., \rho_k$ such that*
*(i) $v = \rho_k \circ ... \circ \rho_1(u)$*
*(ii) For $i = 1, ..., k$ we have $|\rho_i \circ ... \circ \rho_1(u)|_{\mathrm{cyc}} \leq \max\{|u|_{\mathrm{cyc}}, |v|_{\mathrm{cyc}}\}$.*

In particular this gives an algorithm to check whether or not two given words $u, v$ can be obtained one from the other by means of automorphisms of $F_n$. It is enough to write down all the words of length at most $L = \max\{|u|_{\mathrm{cyc}}, |v|_{\mathrm{cyc}}\}$, and to apply all the possible Whitehead automorphisms to all these words: this partitions those words into the equivalence classes under the action of $\mathrm{Aut}(F_n)$, and in particular this tells us whether $u$ and $v$ are in the same orbit or not.

As pointed out by S. M. Gersten, the same peak reduction technique can be applied to a subgroup, in order to reduce the number of vertices of the core graph of the covering space associated to the subgroup. To be precise, the following statement holds:

**Theorem 2.3.3** ([Ger84])**.** *Let $H_1, ..., H_r, K_1, ..., K_r \leq F_n$ be finitely generated subgroups and suppose there is an automorphism $\theta : F_n \to F_n$ such that $\theta(H_i) = K_i$ for $i = 1, ..., r$. Then there are Whitehead automorphisms $\rho_1, ..., \rho_s$ such that $K_i = \rho_s \circ ... \circ \rho_1(H_i)$ for $i = 1, ..., r$ and such that, for each $j = 1, ..., s$, we have*

$$\sum_{i=1}^{r} \|\mathrm{core}(\rho_j \circ ... \circ \rho_1(H_i))\|_{\mathrm{e}} \leq \max\{\sum_{i=1}^{r} \|\mathrm{core}(H_i)\|_{\mathrm{e}}, \sum_{i=1}^{r} \|\mathrm{core}(K_i)\|_{\mathrm{e}}\}$$

Notice that we are considering the unpointed core graphs and not the pointed ones. In particular, this gives an algorithm that allows us to determine whether a subgroup is a free factor, completely analogous to the algorithm for primitive words.

**Theorem 2.3.4.** *The group* $\mathrm{Aut}(F_n)$ *is finitely presented.*

*Proof.* We already know from Theorem 1.4.4 that Whitehead automorphisms are generators for $\mathrm{Aut}(F_n)$. We claim that the relations are the equalities $\tau\sigma = \rho_k...\rho_1$ produced by Lemma 2.3.1; these are a finite number, since $1 \leq k \leq 4$ and for each Whitehead automorphism we only have a finite number of choices.

Suppose we are given a chain of Whitehead automorphisms $\rho_1, ..., \rho_k$ such that $\rho_k...\rho_1 = \mathrm{id}$ and we want to prove that $\rho_k...\rho_1$ is a product of conjugates of our relations. If we apply $\rho_1, ..., \rho_k$ to the family of letters $a_1, ..., a_n$, we get back the letters $a_1, ..., a_n$: we apply repeatedly Theorem 2.3.3 to modify this chain of automorphisms. At each step we modify the chain according to one of the substitutions described in the cases of the proof of Lemma 2.3.1; this means that we are multiplying $\rho_k...\rho_1$ by a conjugate of one of the relations. In the end we are able to get a new chain $\sigma_1, ..., \sigma_h$ such that $\mathrm{id} = \sigma_1 = \sigma_2\sigma_1 = ... = \sigma_h...\sigma_1$, meaning that $\sigma_1 = ... = \sigma_h = \mathrm{id}$. This means that $\rho_k...\rho_1$ can be obtained as product of conjugates of our relations, as desired. □

## 2.4 Whitehead's algorithm and folding

We here provide a third proof of Theorem 2.2.4, due to H. Wilton, see Lemma 2.10 of [Wil18], and independently to M. Heusener and R. Weidmann, see [HW19]. This is a surprisingly short argument based on folding of graphs; this reasoning will also be of fundamental importance in the subsequent Chapter 3.

*Proof of Theorem 2.2.4.* Let $w$ be a cyclically reduced word which is primitive. We will assume that $w$ contains all the letters $a_1, ..., a_n$ at least once: otherwise, if $w$ only contains the letters $a_1, ..., a_k$, then we can just apply the same argument in the free factor $\langle a_1, ..., a_k\rangle \leq \langle a_1, ..., a_n\rangle = F_n$ (using Corollary 2.1.5).

Since $w$ is primitive, we can take a basis $w = w_1, w_2, ..., w_n$ of reduced words. We can build the graph $G$ given by a basepoint $*$, together with a path $p_i$ for each $w_i$: the path $p_i$ goes from $*$ to $*$, and contains an edge for each letter appearing in $w_i$ (in such a way that, moving around the path $p_i$, we read exactly the word $w_i$). Let $G(w)$ denote the subgraph of $G$ given by the only cycle corresponding to the generator $w$.

We now apply a sequence of folding operations to the graph $G$, in order to get a sequence $G \to G' \to ... \to G^{m-1} \to G^m$ as in Proposition 1.2.10: each map $G^i \to G^{i+1}$ consists of a single folding operation, and no further folding operation can be applied to $G^m$. Since $w_1, ..., w_n$ is a basis, we have that $G^m$ is the standard $n$-rose $R_n$. Since $w_1$ is cyclically reduced and $w_2, ..., w_n$ are reduced, Corollary 1.2.13 yields that no graph $G^i$ contains any valence-1 vertex. A folding operation can decrease the rank of the fundamental group, but it can't increase it; since $\pi_1(G)$ has the same rank as $\pi_1(R_n)$, we must have that each folding operation is rank-preserving.

We now look at the graph $G^{m-1}$: it doesn't contain any valence-1 vertex, and a single rank-preserving folding operation sends it to the standard $n$-rose. It is quite easy to see that $G^{m-1}$ has to be of the form described in Figure 2.4 below, for some $1 \leq \alpha \leq \beta \leq n$ with $\alpha < n$ (up to permutation of the letters, and up to substitution of some letter with its inverse).

We have a map of graphs $f : G(w) \to G^{m-1}$ preserving the orientations and the labels on the edges (given by the inclusion $G(w) \to G$ followed by the sequence of foldings). Suppose we have two adjacent letters $w = ...yz...$: this means that $G(w)$ contains two edges labeled $y$ and $z$ with a common endpoint $u$. We either have $f(u) = v$ or $f(u) = v'$, meaning that $\overline{y}$ and $z$ are either both in $\{a_1, \overline{a}_1, a_2, \overline{a}_2, ..., a_\alpha, \overline{a}_\alpha, a_{\alpha+1}, ..., a_\beta\}$ or both in $\{\overline{a}_1, \overline{a}_{\alpha+1}, ..., \overline{a}_\beta, a_{\beta+1}, \overline{a}_{\beta+1}, ..., a_n, \overline{a}_n\}$. This tells us that, if we remove the vertex $\overline{a}_1$ from the Whitehead graph of $w$, we get the disjoint union $V \sqcup V'$ of two separate graphs: $V$ with vertices $a_1, a_2, \overline{a}_2, ..., a_\alpha, \overline{a}_\alpha, a_{\alpha+1}, ..., a_\beta$ and $V'$ with vertices $\overline{a}_{\alpha+1}, ..., \overline{a}_\beta, a_{\beta+1}, \overline{a}_{\beta+1}, ..., a_n, \overline{a}_n$.

If the image $f(G(w)) \subseteq G^{m-1}$ crosses both the edges labeled $a_1$, then in the Whitehead graph of $w$ we have that $\overline{a}_1$ is connected to at least one vertex in $V$ and to one vertex in $V'$; this means $\overline{a}_1$ is a cut vertex (because it satisfies condition (ii) of Definition 2.2.2). If $f(G(w))$ crosses the edge labeled $a_1$ with distinct endpoints, but not the other, then $\overline{a}_1$ is connected to $V'$ but not to $V$; and again $\overline{a}_1$ is a cut vertex (because it satisfies condition (i) of Definition 2.2.2). If $f(G(w))$ doesn't cross the arc labeled $a_1$ with distinct endpoints, then we make use of the assumption that $w$ contains every letter at least once; we get that $f(G(w))$ has to contain the edge $a_{\alpha+i}$ for some $1 \leq i \leq \beta$; this gives that any of $a_{\alpha+i}, \overline{a}_{\alpha+i}$ is a cut vertex for the Whitehead graph of $w$ (because they satisfy condition (i) of Definition 2.2.2). □

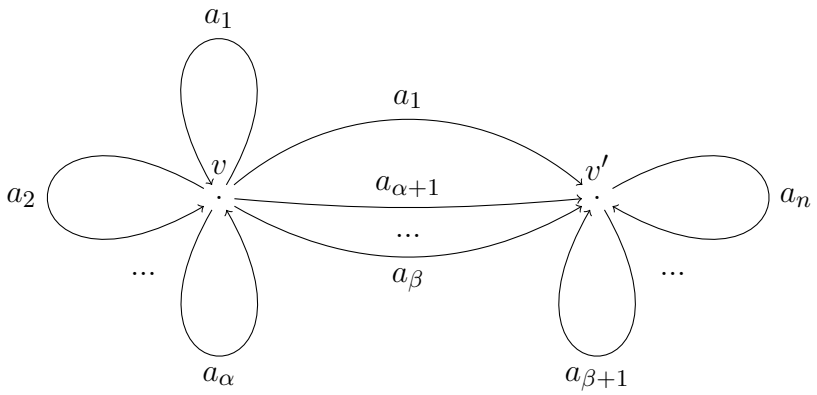


Figure 2.4: The generic graph $G^{m-1}$. This contains exactly one edge with each label, except for the two edges labeled $a_1$. Those two edges have to be folded in order to obtain the $n$-rose.

## 2.5 Another algorithm for primitivity

Recently M. Bestvina and M. Bridson, and independently D. Puder, found another algorithm which allows us to test whether a subgroup is a free factor (and, in particular, whether an element is primitive) (see [Pud13]). This algorithm is completely different from Whitehead's original one, since it's based only on operations performed on the core graph of the subgroup.

### 2.5.1 Immediate quotients

Let $G$ be a labeled graph and let $u, v$ be distinct vertices of $G$. Identify those two vertices to a single vertex and call $G_1$ the resulting labeled graph. The quotient map $q : G \to G_1$ is label-preserving.

**Definition 2.5.1.** *The quotient map $q : G \to G_1$ is called **immediate quotient**.*

In terms of fundamental group, an immediate quotient means adding a new free generator. This additional generator is gained without the need to add any new edge: the set of edges of $G$ is in bijection with the set of edges of $G_1$. Immediate quotients commute with Stallings folding, as we now explain.

**Lemma 2.5.2** (Pull-back of immediate quotients)**.** *Let $G$ be a labeled graph. Let $p : G \to G'$ be a folding operation and let $q' : G' \to G'_1$ be an immediate quotient. Then there is a commutative diagram as in Figure 2.5, where $q$ is an immediate quotient and $p_1$ is a folding operation; the diagram isn't unique in general. For each such diagram, the map $p_1$ is rank-preserving if and only if the map $p$ is.*

*Proof.* If $q'$ identifies two distinct vertices $u', v'$ of $G'$, then we take any two vertices $u, v$ of $G$ with $p(u) = u'$ and $p(v) = v'$, and notice that the choice of $u, v$ needs not to be unique. We define $q$ to be the quotient of $G$ that identifies $u$ with $v$. The lemma follows from the definitions. □

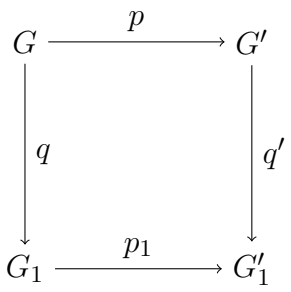


Figure 2.5: The horizontal maps $p, p_1$ are folding operations, the vertical maps $q, q'$ are immediate quotients. The diagram commutes.

### 2.5.2 Relative graphs and folding

The next proposition illustrates how folding operations on a bigger graph $G$ can generate immediate quotients on a subgraph $K$.

**Proposition 2.5.3.** *Let $G$ be a labeled graph and let $K$ be a subgraph. Let $p : G \to G'$ be a folding operation. Let $K' = p(K)$ and let $p_K = p\big|_K : K \to K'$. Then exactly one of the following cases takes place:*

*(i) $p_K$ is the identity.*

*(ii) $p_K$ is a folding operation. In this case $p_K$ is rank-preserving if and only if $p$ is.*

*(iii) $p_K$ is an immediate quotient. In this case $p$ is forced to be rank-preserving.*

*Proof.* Let $e_1, e_2$ be the two edges of $G$ identified by $p$, with endpoints $v, v_1$ and $v, v_2$ respectively.

If $v_i$ belongs to $G \setminus K$ for some $i \in \{1, 2\}$, then $e_i$ belongs to $G \setminus K$ too, and $p_K$ is the identity. So we can assume that both $v_1, v_2$ belong to $K$.

If $v_1 = v_2$ then $p$ is non-rank-preserving; if at most one of $e_1, e_2$ belongs to $K$, then $p_K$ is the identity; if both $e_1, e_2$ belong to $K$, then $p_K$ is a non-rank-preserving folding.

If $v_1 \neq v_2$ then $p$ is rank-preserving; if at most one of $e_1, e_2$ belongs to $K$, then $p_K$ is an immediate quotient; if both $e_1, e_2$ belong to $K$, then $p_K$ is a rank-preserving folding. □

For a finite connected graph $G$ and a connected subgraph $K$, we define the **relative rank** to be $\mathrm{rank}(G, K) = \mathrm{rank}(G) - \mathrm{rank}(K)$ and notice that this is always non-negative.

It's useful to keep track of the numbers $\mathrm{rank}(G), \mathrm{rank}(K), \mathrm{rank}(G, K)$ when applying the above Proposition 2.5.3 to finite connected labeled graphs. Notice that $\mathrm{rank}(G)$ either remains the same (if $p$ is rank-preserving) or decreases by one (if $p$ is non-rank-preserving). Notice also that $\mathrm{rank}(K)$ either remains the same (if $p_K$ is the identity or a rank-preserving-folding) or decreases by one (if $p_K$ is a non-rank-preserving folding) or increases by one (if $p_K$ is an immediate quotient). An analysis of the cases of Proposition 2.5.3 shows that $\mathrm{rank}(G, K)$ either remains the same or decreases by one.

### 2.5.3 The immediate quotients algorithm

**Theorem 2.5.4** (Puder, [Pud13])**.** *Let $H \leq F_n$ be a finitely generated subgroup. Then the following are equivalent:*

*(i) $H$ is a free factor.*

*(ii) It's possible to perform immediate quotients on $\mathrm{core}_*(H)$ such that, after a sequence of rank-preserving folding operations, we end up with a 1-vertex graph.*

To be precise, the 1-vertex graph that we obtain has one petal for each label that appears at least once in $\text{core}_*(H)$, and the number of immediate quotients to be performed on $\text{core}_*(H)$ is the number of petals minus $\text{rank}(H)$.

*Proof.* (i) $\Rightarrow$ (ii) Suppose $H$ is a free factor and let $w_1, ..., w_h, u_1, ..., u_{n-h}$ be a basis for $F_n$ such that $w_1, ..., w_h$ is a basis for $H$. Consider the graph $\text{core}_*(H)$, and add $n-h$ extra closed paths starting and ending at the basepoint, such that while going along those paths we read the words $u_1, ..., u_{n-h}$. Call $G$ the resulting graph, so that $\text{core}_*(H)$ is a subgraph of $G$ and $G \setminus \text{core}_*(H)$ consists of the $n-h$ additional paths.

Since $w_1, ..., w_h, u_1, ..., u_{n-h}$ generates $F_n$, by Proposition 1.2.10 there is a sequence of folding operations $G \to ... \to R_n$ from $G$ to $R_n$. Since $\text{rank}(G) = n$ we must have that all the operations in the sequence are rank-preserving.

We now look at the pair of graphs $(G, \text{core}_*(H))$ and we apply Proposition 2.5.3: we obtain a sequence of rank-preserving folding operations and immediate quotients that goes from $\text{core}_*(H)$ to a subgraph of $R_n$ (to be precise, to the subgraph given by all the edges whose label appears on some edge of $\text{core}_*(H)$).

By an iterated application of Proposition 2.5.2, we can move all the immediate quotient operations to the beginning of the sequence; notice that all the subsequent folding operations remain rank-preserving.

(ii) $\Rightarrow$ (i) Suppose there are $k$ immediate quotients to be performed on $\text{core}_*(H)$ such that, taking the resulting graph to be $L$, there is a sequence $L \to ... \to R$ of rank-preserving folding operations from $L$ to a 1-vertex graph $R$.

Let $u_i, v_i$ be the two vertices of $\text{core}_*(H)$ that are identified in the $i$-th immediate quotient, for $i = 1, ..., k$. Let $p_i, q_i$ be reduced combinatorial paths in $\text{core}_*(H)$ going from the basepoint to $u_i, v_i$ respectively, and let $c_i, d_i$ be the words that we read along $p_i, q_i$ respectively. Take $\text{core}_*(H)$ and add a path from the basepoint to itself, such that along the path we read the word $c_i\overline{d}_i$: if we fold this extra path on $\text{core}_*(H)$ (folding half of it on $p_i$ and half of it on $q_i$), after a few rank-preserving folding operations we obtain exactly the graph $\text{core}_*(H)$ with the two vertices $u_i, v_i$ identified.

Let $G$ be the graph obtained from $\text{core}_*(H)$ by adding $k$ paths from the basepoint to itself, such that on the $i$-th path we read the word $c_i\overline{d}_i$. We can perform rank-preserving folding operations on $G$ in order to obtain the graph $\text{core}_*(H)$ with the $k$ immediate quotients performed. We can the perform further rank-preserving folding operations in order to get the rose $R$.

A basis for the fundamental group of the graph $G$ is given by a basis for $H$ together with the words $c_1\overline{d}_1, ..., c_k\overline{d}_k$. Since all the foldings in the sequence from $G$ to $R$ are rank-preserving, the two fundamental groups are isomorphic, and thus $H$ is a free factor in $\pi_1(R, *)$. Of course $\pi_1(R, *)$ is a free factor in $F_n$, and the conclusion follows from Corollary 2.1.6. □

Given a finitely generated subgroup $H \leq F_n$, there is a finite number of possible

immediate quotients that we can perform on $\mathrm{core}_*(H)$. For each finite set of immediate quotients of $\mathrm{core}_*(H)$, we can fold as in Proposition 1.2.10 and we can determine whether we obtain a rose, and whether the folding operations that we use are all rank-preserving. Thus we obtain the following:

**Corollary 2.5.5.** *The above process gives an algorithm that decides whether a finitely generated subgroup $H \leq F_n$ is a free factor or not.*

Another important feature of Theorem 2.5.4 is the following: in order to obtain, from a basis for $H$, a basis for $\pi_1(R, *)$, we add words $c_i \overline{d}_i$ whose length is bounded in terms of the number of edges of $\mathrm{core}_*(H)$. Recall that $\|\cdot\|_{\mathrm{e}}$ denotes the number of edges of a finite graph. We have the following:

**Corollary 2.5.6.** *Let $H \leq F_n$ be a free factor with $\mathrm{rank}(H) = h$. Then there are elements $w_1, ..., w_{n-h} \in F_n$ such that $F_n = H * \langle w_1, ..., w_{n-h} \rangle$ and such that $|w_i| \leq 2\|\mathrm{core}_*(H)\|_{\mathrm{e}}$.*

# 3 | A fine property of Whitehead's algorithm

Whitehead's theorem grants us that, if $w$ is cyclically reduced primitive word of length greater than 1 in a finitely generated free group $F_n$, then there is a Whitehead automorphism $\varphi = (A, a)$ such that $|\varphi(w)|_{\text{cyc}} < |w|_{\text{cyc}}$. This inequality gives information about the "global" length of $\varphi(w)$, but it doesn't tell us anything about how or where exactly the cancellations take place inside the word. In this chapter we introduce a fine cancellation property of Whitehead's algorithm, proving that the automorphism can be chosen in a way that allows to control the cancellations in $\varphi(w)$ in a very precise way. This improves the above inequality from "global" to one that holds "locally everywhere".

In Section 3.1 we introduce the *fine property* for primitive words. We show how this extra property allows one to prove, for example, a new criterion that characterizes free factors of $F_n$ in terms of their primitive words. In Section 3.2 we show how the fine property can be generalized to free factors. This allows us to prove a relative version of Whitehead's algorithm.

In Section 3.3 we introduce the free factor complex of a free group, and we show how the fine property allows us to obtain an algorithm to determine whether or not two vertices in the free factor complex have distance $d$ for $d = 1, 2, 3$, as well as $d = 4$ in a special case. The existence of an algorithm for distance $d = 4$ in the general case is an open problem.

In Section 3.4 we introduce the notion of *echelon subgroup* of a free group. We make use of the fine property to provide an algorithm that determines whether a given subgroup of $F_n$ is echelon; this answers a question of A. Rosenmann [Ros13]. We also provide a counterexample showing that the intersection of two echelon subgroups needs not to be echelon; this answers another question of Rosenmann [Ros13].

## 3.1 The fine property for primitive words

In this section we investigate how cancellations take place when we apply a Whitehead automorphism to a word. We prove a fine cancellation property for Whitehead's algorithm for primitive words. We make use of this property to obtain a new

equivalent condition for a subgroup $H$ of a free group $F_n$ to be a free factor: we have that $H$ is a free factor in $F_n$ if and only if every element which is primitive in $H$ is also primitive in $F_n$.

### 3.1.1 A cancellation property of Whitehead automorphisms

Let $F_n$ be a free group generated by $a_1, ..., a_n$. Let $w = b_1...b_l$ be a cyclically reduced word with $b_j \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$. Let $\varphi = (A, a)$ be a Whitehead automorphism. We can substitute each $b_j$ with the sequence $\varphi(b_j)$ (which is either $b_j$ or $ab_j$ or $b_j\overline{a}$ or $ab_j\overline{a}$): this produces a new word $\varphi(b_1)...\varphi(b_l)$ which represents the element $\varphi(w)$; this word will not be cyclically reduced in general. In what follows, when we speak of the free or cyclic reduction, we mean any sequence of moves in which an adjacent pair of letters $a_i\overline{a}_i$ or $\overline{a}_ia_i$ is replaced by the empty word.

**Lemma 3.1.1.** *Let $w = b_1...b_l$ be a cyclically reduced word, and let $\varphi = (A, a)$ be a Whitehead automorphism. Then, in the process of cyclic reduction for the sequence $\varphi(b_1)...\varphi(b_l)$, no letter different from $a$ gets cancelled.*

*Proof.* For simplicity of notation, we prove the proposition only for the free reduction process; the proof for the cyclic reduction process is completely analogous.

Fix a process of free reduction for $\varphi(b_1)...\varphi(b_l)$. Suppose some cancellation takes place, involving a letter which is not $a$ nor $\overline{a}$. Consider the first such cancellation. Suppose this cancellation involves a letter from the block $\varphi(b_j)$ and one from the block $\varphi(b_k)$, with $j < k$. Then we must have $b_k = \overline{b}_j$, and either all the letters in between are $a$, or all of them are $\overline{a}$. This means that the word $w$ has the form either $...b_ja^d\overline{b}_j...$ or $...b_j\overline{a}^d\overline{b}_j...$ for some $d \geq 0$. Also, since the writing $b_1...b_l$ was reduced, we must have $d > 0$.

We assume $w$ has the form $...b_ja^d\overline{b}_j...$, the other case being completely analogous. If $\overline{b}_j \notin A$, then the sequence $\varphi(b_1)...\varphi(b_l)$ has the form $...b_ja^d\overline{b}_j...$, and at least one $a$ letter survives between $b_j$ and $\overline{b}_j$, and thus $b_j$ is not allowed to cancel with $\overline{b_j}$. If $\overline{b}_j \in A$, then the sequence $\varphi(b_1)...\varphi(b_l)$ has the form $...b_j\overline{a}a^da\overline{b}_j...$, and again we see that at least one $a$ letter survives between $b_j$ and $\overline{b}_j$, and thus $b_j$ is not allowed to cancel with $\overline{b}_j$. This contradiction completes the proof. □

### 3.1.2 The fine property

Theorem 2.2.5 tells us that, given a cyclically reduced primitive word $w \in F_n$ which is not a single letter, there is a Whitehead automorphism reducing its cyclic length.

**Theorem 3.1.2** (Fine property, [Asc21])**.** *The automorphism in Theorem 2.2.5 can be chosen in such a way that every $a$ or $\overline{a}$ letter, which is added when we apply $\varphi$ to $w$ letter by letter, immediately cancels (in the cyclic reduction process).*

*Proof.* This was already observed in Remark 1 at the end of the proof of Theorem 2.2.3. □

Let me give an example to illustrate the property of Theorem 3.1.2. Let $w = ababa\bar{b}c$ in $F_3 = \langle a, b, c\rangle$ and consider the Whitehead automorphism $\varphi : F_3 \to F_3$ given by $a \mapsto a\bar{b}$ and $b \mapsto b$ and $c \mapsto c$ (meaning that the letter $b$ acts on the set of letters $\{\overline{a}\}$); notice that this automorphism doesn't come from a cut vertex in the Whitehead graph of $w$. The word becomes $\varphi(w) = aaa\overline{b}\overline{b}c$: this is shorter, meaning that the automorphism $\varphi$ would be suitable for Theorem 2.2.5 applied to the word $w$. But $\varphi$ doesn't satisfy Theorem 3.1.2, because a letter $\bar{b}$ appears between the last $a$ and the $c$, and it doesn't cancel.

*Remark* 2. By looking carefully at the proof of Theorem 2.2.3, we notice that it is also possible to count the number of cancellations that take place, by just looking at the Whitehead graph of $w$ and at the Whitehead automorphism $\varphi = (A, a)$; this is also shown in Proposition 2.2 of [PW14]. To be precise, if $c(a, A)$ is the number of edges of the Whitehead graph of $w$ that connect the vertex $a$ to a vertex of $A$, then we have $|\varphi(w)|_{\text{cyc}} = |w|_{\text{cyc}} - c(a, A)$.

### 3.1.3 Recognizing free factors from their primitive elements

Let $H \leq F_n$ be any subgroup. If $H$ is a free factor, then of course every element which is primitive in $H$ has to be primitive in $F_n$. Here we make use of the fine property of Theorem 3.1.2 to prove that the converse holds too: if $H$ is not a free factor, then there is an element which is primitive in $H$ but not in $F_n$. This section is completely dedicated to the proof of the existence of such a witness.

**Theorem 3.1.3** ([Asc21])**.** *Let $H \leq F_n$ be a finitely generated subgroup. Suppose that every element which is primitive in $H$ is also primitive in $F_n$. Then $H$ is a free factor.*

*Proof.* The proof proceeds by induction on the rank of the subgroup. For the base step, we notice that for a subgroup of rank 1 the statement is trivially true. For the inductive step, suppose we know the statement to be true for subgroups of rank $k$, and we want to prove it for subgroups of rank $k + 1$.

Take a subgroup $H = \langle w_1, ..., w_{k+1}\rangle$ of rank $k + 1$, meaning that $w_1, ..., w_{k+1}$ is a basis for $H$, and suppose that every element $v$ which is primitive in $H$ is also primitive in $F_n$. We consider the subgroup $H' = \langle w_1, ..., w_k\rangle$, and we notice that every element $v$ which is primitive in $H'$ is also primitive in $H$, and thus is primitive in $F_n$. Then $H'$ has rank $k$ and satisfies the hypothesis of the theorem, so by inductive hypothesis we get that $H'$ is a free factor. So we can take an automorphism $\theta : F_n \to F_n$ with $\theta(w_i) = a_i$ for $i = 1, ..., k$. Instead of proving that $H$ is a free factor in $F_n$, we prove that $\theta(H)$ is a free factor in $F_n$; but $\theta(H) = \langle a_1, ..., a_k, \theta(w_{k+1})\rangle$, so it

is enough to prove the statement of Theorem 3.1.3 for subgroups $H$ of the form $H = \langle a_1, ..., a_k, w\rangle$.

The statement in the case of subgroups of the form $H = \langle a_1, ..., a_k, w\rangle$ will be proved by induction on the length of $w$. The base step where $w$ has length one is trivial.

We observe that, if the first (or the last) letter of $w$ is $b \in \{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$, then we can define the shorter word $w' = \overline{b}w$ (or $w\overline{b}$). But then $H = \langle a_1, ..., a_k, w\rangle = \langle a_1, ..., a_k, w'\rangle$, and so we are done by inductive hypothesis. Thus in the following we assume this is not the case.

We consider the word $v = a_1 w a_1 \widetilde{w}$ where $\widetilde{w}$ is any word in the letters $\{a_1, ..., a_k\}$ with the following properties:

(i) The Whitehead graph of $\widetilde{w}$ contains at least one edge joining each pair of distinct vertices in $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$.

(ii) When we write $a_1 w a_1 \widetilde{w}$ we get a cyclically reduced word, without any cancellation needed.

For example we may take $\widetilde{w} = (a_1 a_1)(a_2 a_2)...(a_k a_k) \prod_{1 \leq i < j \leq k} a_1(a_i a_j)(a_i \overline{a}_j)$, where the factors in the product are ordered lexicographically (but any ordering works).

The word $v$ is primitive in the subgroup $H$, so by the hypothesis it has to be primitive in $F$, and in particular we can take an automorphism $\varphi$ satisfying Theorems 2.2.5 and 3.1.2. We now look at how $\varphi$ acts on the letters $a_1, ..., a_k$ and on the word $w$.

CASE 1: Suppose the acting letter $a$ is different from $a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k$.

Then $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ is either contained in $A$ or disjoint from $A$. This is because the vertices $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ are all pairwise connected in the Whitehead graph of $v$.

SUBCASE 1.1: Suppose $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ is disjoint from $A$.

This means that $\varphi(a_i) = a_i$ for $i = 1, ..., k$ and $\varphi\langle a_1, ..., a_k, w\rangle = \langle a_1, ..., a_k, \varphi(w)\rangle$.

If we apply $\varphi$ to $v$ letter by letter, and then we reduce the resulting word, we have that only $a$-letters and $\overline{a}$-letters can cancel. This implies that the reduced word representing $\varphi(v) = a_1 \varphi(w) a_1 \widetilde{w}$ has $a_1$ as initial segment and $a_1 \widetilde{w}$ as final segment (and in particular it's also cyclically reduced). This has to be strictly shorter than the reduced word representing $v = a_1 w a_1 \widetilde{w}$: we can remove the common initial segment $a_1$ and the common final segment $a_1 \widetilde{w}$, and we deduce that the reduced word representing $\varphi(w)$ has to be strictly shorter than the reduced word representing $w$, and thus we are done by inductive hypothesis.

SUBCASE 1.2: $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ is contained in $A$.

This means that $\varphi(a_1) = a a_1 \overline{a}, ..., \varphi(a_k) = a a_k \overline{a}$.

Moreover we have $\varphi\langle a_1, ..., a_k, w\rangle = a\langle a_1, ..., a_k, \overline{a}\varphi(w)a\rangle\overline{a}$.

As in subcase 1.1, we have that the reduced word representing $\varphi(v) = a a_1(\overline{a}\varphi(w)a)a_1 \widetilde{w}\overline{a}$ has $a a_1$ as initial segment and $a_1 \widetilde{w}\overline{a}$ as final segment, and thus its cyclic reduction is

$a_1(\overline{a}\varphi(w)a)a_1\widetilde{w}$ and has $a_1$ as initial segment and $a_1\widetilde{w}$ as final segment. This has to be strictly shorter than the reduced word representing $v = a_1wa_1\widetilde{w}$: we can remove the common initial segment $a_1$ and the common final segment $a_1\widetilde{w}$, and we deduce that the reduced word representing $\varphi(w)$ has to be strictly shorter than the reduced word representing $w$, and thus we are done by inductive hypothesis.

CASE 2: Suppose the acting letter $a$ is one of $a_2, ..., a_k, \overline{a}_2, ..., \overline{a}_k$.

Then $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ is disjoint from $A$, because all the vertices $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\} \setminus \{a, \overline{a}\}$ are connected to $\overline{a}$ in the Whitehead graph of $v$.

Now we have $\varphi(a_i) = a_i$ for $i = 1, ..., k$ and $\varphi\langle a_1, ..., a_k, w\rangle = \langle a_1, ..., a_k, \varphi(w)\rangle$.

We proceed exactly as in subcase 1.1. Since $a \neq a_1, \overline{a}_1$, we deduce that the reduced word representing $\varphi(v) = a_1\varphi(w)a_1\widetilde{w}$ has $a_1$ as initial segment and $a_1\widetilde{w}$ as final segment. This has to be strictly shorter than the reduced word representing $v = a_1wa_1\widetilde{w}$: we can remove the common initial and final segments, and we obtain that the reduced word representing $\varphi(w)$ is strictly shorter than the reduced word representing $w$, and we are again done by inductive hypothesis.

CASE 3: Suppose $a = \overline{a}_1$ (the case $a = a_1$ is completely analogous).

As in case 2, we get that $\{a_1, ..., a_k, \overline{a}_1, ..., \overline{a}_k\}$ is disjoint from $A$, and thus $\varphi(a_i) = a_i$ for $i = 1, ..., k$ and $\varphi\langle a_1, ..., a_k, w\rangle = \langle a_1, ..., a_k, \varphi(w)\rangle$.

Reasoning as in the previous cases, we obtain that the reduced word representing $\varphi(v) = (a_1\varphi(w))a_1\widetilde{w}$ is cyclically reduced and has $a_1\widetilde{w}$ as final segment. Since this has to be strictly shorter than the reduced word representing $v = a_1wa_1\widetilde{w}$, we can deduce that the reduced word representing $a_1\varphi(w)$ has to be strictly shorter than the reduced word representing $a_1w$. Now some care is needed.

If $a_1$ is an initial segment of the reduced word representing $a_1\varphi(w)$, then we obtain that the reduced word representing $\varphi(w)$ is strictly shorter than the reduced word representing $w$, and we are done by inductive hypothesis.

Otherwise, let $t$ be the first letter of $w$, and we must have $t \in A$. We look at the arcs between $a = \overline{a}_1$ and $A$ in the Whitehead graph of $v$: we certainly have at least one arc from $\overline{a}_1$ to $t$.

SUBCASE 3.1: If we have more than one arc from $\overline{a}_1$ to $t$, or if we have any other arc from $\overline{a}_1$ to $A$, then each of those arcs give cancellations when applying $\varphi$ to $w$ letter by letter and reducing the resulting word. We already knew that $a_1\varphi(w)$ was strictly shorter than $a_1w$, but now we got at least one additional cancellation, so we are able to deduce that $a_1\varphi(w)$ is strictly shorter than $w$. Now we observe that $\varphi\langle a_1, ..., a_k, w\rangle = \langle a_1, ..., a_k, a_1\varphi(w)\rangle$, and we are done by inductive hypothesis.

SUBCASE 3.2: Suppose we only have one arc from $\overline{a}_1$ to $t$, and no other arc from $\overline{a}_1$ to $A$. If the vertex $t$ has degree at least 2, then we use $t$ instead of $\overline{a}_1$ as cut vertex for the Whitehead graph of $v$, and we end up in case 1, and we are done. If the

vertex $t$ has degree 1, then the letter $t$ appears exactly once inside $w$; in this case we have the automorphism $\eta : F_n \to F_n$ which keeps all the letters fixed, except for $t \mapsto w$; this gives $\eta\langle a_1, ..., a_k, t\rangle = \langle a_1, ..., a_k, w\rangle$, showing that the subgroup is a free factor, as desired. □

We proved that Theorem 3.1.3 holds for a subgroup $H$ which is finitely generated, but that hypothesis can easily be waived. In fact, for a subgroup of rank greater than the rank of $F_n$ (and, as a consequence, for a subgroup of infinite rank), the hypothesis of Theorem 3.1.3 can't hold.

## 3.2 The fine property for free factors

In this section we generalize the fine property introduced in Section 3.1 to subgroups: we prove that, if a subgroup $H$ is a free factor, then there is a Whitehead automorphism $\varphi = (A, a)$ such that $\text{core}(\varphi(H))$ can be obtained from $\text{core}(H)$ by means of a quotient map, that takes some of the edges of $\text{core}(H)$ and collapses each of them to a single point. In particular, this allows us to control the action of $\varphi$ on all the subgroups of $H$ at the same time.

### 3.2.1 Whitehead graph of a subgroups

Let $F_n$ be a free group generated by $a_1, ..., a_n$.

**Definition 3.2.1.** *Let $G$ be a labeled graph and let $v \in G$ be a vertex. Define the **letters at** $v$ to be the subset $L(v) \subseteq \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$ of the labels of the edges coming out of $v$.*

In other words, we have $a_i \in L(v)$ if and only if $G$ contains an edge labeled $a_i$ coming out of $v$, and $\overline{a}_i \in L(v)$ if and only if $G$ contains an edge labeled $a_i$ going into $v$.

**Definition 3.2.2.** *Let $G$ be a labeled graph. Define the **Whitehead graph** of $G$ as follows:*

*(i) We have $2n$ vertices labeled $a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n$.*

*(ii) For every vertex $v \in G$ and for every pair $b, c \in L(v)$ of distinct letters at $v$, we draw an (unoriented) arc from $b$ to $c$ in the Whitehead graph.*

This means that the Whitehead graph of $G$ contains an edge for every (legal) turn in $G$. Notice that the Whitehead graph contains a complete subgraph with vertex set $L(v)$ for every vertex $v \in G$; moreover, the Whitehead graph is exactly the union of these complete subgraphs.

Observe that, if $p : G \to G'$ is a folding operation, then the Whitehead graphs of $G$ and $G'$ satisfy the following: if two vertices $b, c$ in the Whitehead graph of $G$ are

connected by at least one edge, then the same two vertices $b, c$ are connected by at least one edge in the Whitehead graph of $G'$ as well (even though the number of edges connecting $b$ and $c$ might change).

We define the Whitehead graph of a nontrivial finitely generated subgroup $H \leq F_n$ to be the Whitehead graph of $\text{core}(H)$. When the subgroup $H$ is generated by a single word $H = \langle w \rangle$, the Whitehead graph of $H$ coincides with the Whitehead graph of the cyclic reduction of $w$; in this sense, this notion of Whitehead graph is a generalization of the previous Definition 2.2.1. We can also define the notion of cut vertex for the Whitehead graph of a subgroup exactly as in Definition 2.2.2.

### 3.2.2 Whitehead automorphisms and subdivision of graphs

We are now interested in how the core graph of a subgroup changes when we apply a Whitehead automorphism. We are thus going to describe an operation which we call **subdivision**, which we perform on a labeled graph. In what follows, we work with a fixed Whitehead automorphism $\varphi = (A, a)$ with $a \in \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$. We adopt the convention that an edge labeled with $c \in \{\overline{a}_1, ..., \overline{a}_n\}$ and with a certain orientation is the same as an edge labeled with $\overline{c}$ and with opposite orientation.

Let $G$ be a labeled graph. Choose an edge $e \in G$, oriented and labeled with a letter $b \in \{a_1, ..., a_n\}$. If $b, \overline{b} \notin A$ we do not change the edge $e$. If $b \in A$ and $\overline{b} \notin A$, then we subdivide $e$ in two edges, and to the first we give the label $a$ and the orientation of $e$, and to the second we give the label $b$ and the same orientation. If $b \notin A$ and $\overline{b} \in A$, then we subdivide $e$ in two edges, and to the first we give the label $b$ and the same orientation of $e$, and to the second we give the label $a$ and opposite orientation. If $b, \overline{b} \in A$ then we perform both transformations on $e$, as in Figure 3.1.

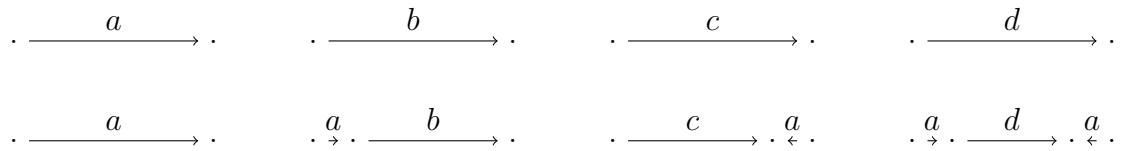


Figure 3.1: The effect of the subdivision operation on the single edges. Here $F_4 = \langle a, b, c, d \rangle$ and $\varphi = (\{b, \overline{c}, d, \overline{d}\}, a)$, meaning that $\varphi(a) = a$ and $\varphi(b) = ab$ and $\varphi(c) = c\overline{a}$ and $\varphi(d) = ad\overline{a}$. Above we see the edges before the subdivision, while below we see them after the subdivision.

We apply the subdivision operation to each edge of $G$, in order to obtain another labeled graph, which will be called **$\varphi$-subdivided graph**, which we denote by $\text{subd}_\varphi(G)$.

Notice that every vertex of $G$ is in a natural way also a vertex of $\text{subd}_\varphi(G)$. If we have a vertex $v \in G$ and a letter $b \in L(v) \cap A$, then the subdivision operation on the corresponding edge will create a new vertex $u$ together with an edge labeled $a$ going

from $v$ to $u$: we say that the vertex $u$ is a **new vertex near** $v$. For every vertex of $\mathrm{subd}_\varphi(G)$, it is either a vertex of $G$, or a new vertex near a unique $v \in G$.

To a pointed labeled graph $G$ we associate the subgroup $H \leq F_n$ given by the image $H = \pi_1(f)(\pi_1(G, *))$, where $f : G \to R_n$ is the labeling map. It is immediate to see that, if $G$ is associated with the subgroup $H$, then $\mathrm{subd}_\varphi(G)$ is associated with the subgroup $\varphi(H)$. This observation yields the following (which is essentially the same as Lemma 2 in [CG10]):

**Proposition 3.2.3.** *Let $H \leq F_n$ be a non-trivial finitely generated subgroup and let $\varphi = (A, a)$ be a Whitehead automorphism. Then we have the following:*

*(i)* $\mathrm{core}_*(\varphi(H)) = \mathrm{core}_*(\mathrm{fold}(\mathrm{subd}_\varphi(\mathrm{core}_*(H))))$

*(ii)* $\mathrm{core}(\varphi(H)) = \mathrm{core}(\mathrm{fold}(\mathrm{subd}_\varphi(\mathrm{core}(H))))$

*Proof.* (i) For the labeled graph $\mathrm{core}_*(H)$ with labeling map $f : \mathrm{core}_*(H) \to R_n$, we have that $f_*(\pi_1(\mathrm{core}_*(H), *)) = H$. After the subdivision operation, if we call $g : \mathrm{subd}_\varphi(\mathrm{core}_*(H)) \to R_n$ the labeling map, we have that $g_*(\pi_1(\mathrm{subd}_\varphi(\mathrm{core}_*(H)), *)) = \varphi(H)$. In particular, Proposition 1.2.10 tells us that $\mathrm{fold}(\mathrm{subd}_\varphi(\mathrm{core}_*(H)))$ can be embedded in $\mathrm{cov}(\varphi(H))$ as a subgraph containing $\mathrm{core}_*(\varphi(H))$. It follows that

$$\mathrm{core}_*(\mathrm{fold}(\mathrm{subd}_\varphi(\mathrm{core}_*(H)))) = \mathrm{core}_*(\varphi(H))$$

as desired.

(ii) Analogous to (i). □

We are also interested in having a detailed description of how the folding operations take place. Some partial description is already given in Lemma 3 of [CG10], but we provide the more precise technical Lemma 3.2.4, which is a generalization of Lemma 3.1.1.

**Lemma 3.2.4.** *Let $H \leq F_n$ be a non-trivial finitely generated subgroup and let $\varphi = (A, a)$ be a Whitehead automorphism. Consider the graph* $\mathrm{subd}_\varphi(\mathrm{core}(H))$*. For every vertex $v \in \mathrm{core}(H)$, fold together all the edges of* $\mathrm{subd}_\varphi(\mathrm{core}(H))$ *going out of $v$ and labeled with $a$. Then, after these folding operations, no further folding operation is possible. In particular, for every folding sequence starting from* $\mathrm{subd}_\varphi(\mathrm{core}(H))$*, the sequence contains only rank-preserving folding operations involving edges labeled $a$.*

*Proof.* Let $G = \mathrm{core}(H)$ and denote with $L(v)$ the set of letters at the vertex $v$ in $G$. For every vertex $v \in G$, take all the edges in $\mathrm{subd}_\varphi(G)$ labeled $a$ and going out of $v$, and fold them all together; call the resulting graph $G'$. Our aim is to show that no folding operation is possible on $G'$. This is equivalent to showing that, for every vertex $u \in G'$ and for every letter, there is at most one edge with that label going out of $u$.

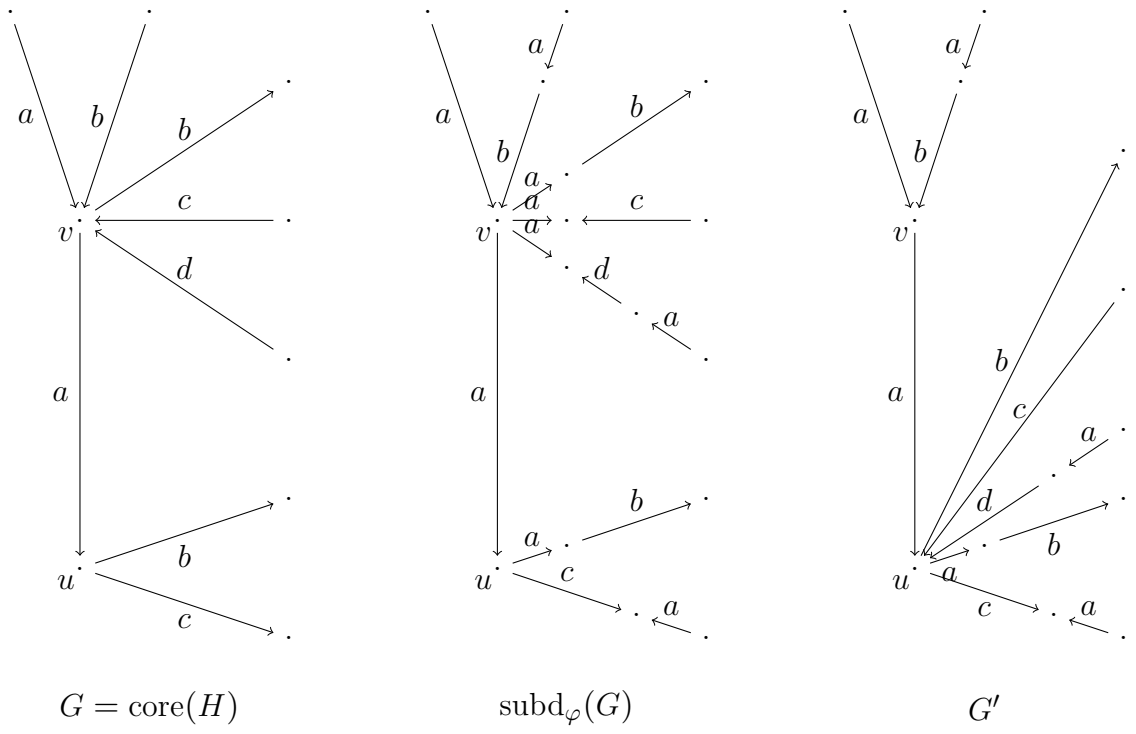


Figure 3.2: A local picture of the graph $G$ and how it changes during the proof of Lemma 3.1.1 (with a focus on case 2). Here $F_4 = \langle a, b, c, d \rangle$ and $\varphi = (\{b, \overline{c}, d, \overline{d}\}, a)$ is the same as in Figure 3.1. We see a portion of the starting graph $G$, its subdivision, and the corresponding portion of $G'$.

Suppose we have a vertex $v \in G$ with $L(v) \cap A = \emptyset$. Then no new vertex is created near $v$ in the subdivision operation.

Suppose we have a vertex $v \in G$ with $L(v) \cap A \neq \emptyset$ and $a \in L(v)$: this means that $G$ contains an edge labeled $a$ going from $v$ to some vertex $u$ of $G$. Then every new vertex, which is created near $v$ with the subdivision operation, gets identified with $u$ in $G'$.

Suppose we have a vertex $v \in G$ with $L(v) \cap A \neq \emptyset$ and $a \notin L(v)$. Then all the new vertices, which are created near $v$ with the subdivision operation, are identified together into a single vertex.

Thus $G'$ contains only two types of vertices: vertices $u$ that are also vertices of $G$, and new vertices $u_1$ near a vertex $u \in G$ with $L(u) \cap A \neq \emptyset$ and $a \notin L(u)$.

CASE 1: Suppose we have a vertex $u \in G$ with $\overline{a} \notin L(u)$. Since $\overline{a} \notin L(u)$, we have that the vertex $u \in \mathrm{subd}_\varphi(G)$ does not get identified with any other vertex during the folding operations that produce $G'$. The edges going out of $u \in G$ with label in $L(u) \cap A^c$ give edges going out of $u \in G'$ with the same label. The edges going out of $u \in G$ with label in $L(u) \cap A$ give edges labeled $a$ going out of $u \in \mathrm{subd}_\varphi(G)$; all the edges labeled $a$ and going out of $u \in \mathrm{subd}_\varphi(G)$ are folded together in $G'$,

meaning that there is at most one edge labeled $a$ going out of $u \in G'$. Thus in this case the vertex $u \in G'$ has at most one edge with each label going out of it.

CASE 2: Suppose we have a vertex $u \in G$ with $\overline{a} \in L(u)$. This means that $G$ contains an edge labeled $a$ going from $v$ to $u$ (see also Figure 3.2). It is possible that the subdivision operation creates new vertices near $v$; the folding operation identifies all such vertices with $u$. Thus, for every letter $l \in L(v) \cap A$, the vertex $u \in G'$ has one edge labeled $l$ going out of it. The edges going out of $u$ with label in $L(u) \cap A^c$ give edges going out of $u \in G'$ with the same label. Notice that $L(v) \cap A$ and $L(u) \cap A^c$ are disjoint, so we can not get two vertices with the same label in this way. As in case 1, the edges going out of $u \in G$ with label in $L(u) \cap A$ give edges labeled $a$ going out of $u \in \mathrm{subd}_\varphi(G)$; all the edges labeled $a$ going out of $u \in \mathrm{subd}_\varphi(G)$ are folded together in $G'$, meaning that there is at most one edge labeled $a$ going out of $u \in G'$. Thus in this case the vertex $u \in G'$ has at most one edge with each label going out of it.

CASE 3: Suppose we have a vertex $u \in G$ with $L(u) \cap A \neq \emptyset$ and $a \notin L(u)$. This means that all the vertices which are created near $u$ fold together into a single vertex $u_1 \in G'$. Each edge going out of $u \in G$ with label in $L(u) \cap A$ gives one edge going out of $u_1 \in G'$ with the same label. The vertex $u_1$ also has one edge labeled $\overline{a}$ going out of it, and notice that $\overline{a} \notin L(u) \cap A$. It follows that the vertex $u_1 \in G'$ has at most one edge with each label going out of it.

Since we examined each vertex of $G'$, we conclude that no folding operation is possible on $G'$. We observe that, by Proposition 1.2.10, every maximal folding sequence starting from $\mathrm{subd}_\varphi(G)$ will end with the graph $G'$: it follows that every such folding sequence can contain only rank-preserving folding operations (since $\mathrm{rank}(G') = \mathrm{rank}(G)$), and each folding operation in each such sequence can only involve edges labeled $a$ (since, for every other label $b$, the graphs $G$ and $G'$ have the same number of edges with label $b$). The conclusion follows. □

### 3.2.3 Whitehead's algorithm for subgroups

We are now ready to state the analogues of Theorems 2.2.4, 2.2.3 and 3.1.2 for free factors. Recall that $\|\cdot\|_{\mathrm{e}}$ denotes the number of edges of a finite graph, and that $L(v)$ is as introduced in Definition 3.2.1.

**Theorem 3.2.5.** *Let $H \leq F_n$ be a free factor and suppose* $\mathrm{core}(H)$ *has more than one vertex. Then the Whitehead graph of $H$ contains a cut vertex.*

**Theorem 3.2.6** (See also Theorem 2.3.3)**.** *Let $H \leq F_n$ be a free factor and suppose* $\mathrm{core}(H)$ *has more than one vertex. Then there is a Whitehead automorphism $\varphi$ such that*

$$\|\mathrm{core}(\varphi(H))\|_{\mathrm{e}} < \|\mathrm{core}(H)\|_{\mathrm{e}}$$

**Theorem 3.2.7** ([Asc21]). *The automorphism $\varphi = (A, a)$ in Theorem 3.2.6 can be chosen in such a way that, at each vertex $v$ of* $\mathrm{core}(H)$*, exactly one of the following configurations takes place:*

*(i)* $L(v) \cap A = \emptyset$.

*(ii)* $L(v) \subseteq A$.

*(iii)* $a \in L(v)$ *and* $L(v) \subseteq A \cup \{a\}$.

*Remark.* Case (i) means that we do not act on any of the letters at $v$. Case (ii) means that we act on all the letters at $v$. Case (iii) means that we act on all the letters at $v$ except for $a$.

*Remark.* Notice that, if $\overline{a} \in L(v)$, then $v$ necessarily falls into case (i).

For the proof of Theorem 3.2.5 we proceed as in Section 2.4.

*Proof of Theorem 3.2.5.* Let $H$ be a free factor such that $\mathrm{core}(H)$ has more than one vertex. Up to conjugation, we can assume that the basepoint of $\mathrm{core}_*(H)$ belongs to $\mathrm{core}(H)$. We also assume that $\mathrm{core}(H)$ contains each letter $a_1, ..., a_n$ at least once; otherwise, if $\mathrm{core}(H)$ only contains the letters $a_1, ..., a_k$, then we can just apply the same argument in the free factor $\langle a_1, ..., a_k \rangle \leq \langle a_1, ..., a_n \rangle = F_n$ (using Corollary 2.1.6).

Since $H$ is a free factor, we can take a basis for $H$ and add reduced words $w_1, ..., w_r$ in order to make it a basis for $F_n$. Take the graph $\mathrm{core}(H)$ (that, as assumed before, contains the basepoint) and add $r$ paths from the basepoint to itself, corresponding to the words $w_1, ..., w_r$, in order to get a graph $G$. Then, apply a sequence of folding operations $G \to G^1 \to ... \to G^m$ until no further folding operation is possible, as in Proposition 1.2.10. Since $\langle H, w_1, ..., w_r \rangle = F_n$, we must have that $G^m = R_n$ is the standard $n$-rose. Using Corollary 1.2.13, we can see that no graph in the sequence contains any valence-1 vertex. Also, since $\pi_1(G)$ has the same rank as $\pi_1(R_n)$, we must have that each folding operation is rank-preserving.

Thus $G^{m-1}$ has no valence-1 vertex, and produces the standard $n$-rose with just one rank-preserving folding operation. It is easy to see that $G^{m-1}$ has to be of the form described in Figure 2.4, for some $1 \leq \alpha \leq \beta \leq n$ with $\alpha < n$ (up to permutation of the letters, and up to substitution of some letter with its inverse) (and the two edges labeled $a_1$ are the ones to be folded in order to obtain the $n$-rose).

We have a map of graphs $f : \mathrm{core}(H) \to G^{m-1}$ which preserves orientations and labels of edges. The image of $f(\mathrm{core}(H)) \subseteq G^{m-1}$ contains each letter at least once, meaning that it has to cross at least one of the edges connecting $v$ to $v'$ (see Figure 2.4). If it crosses the edge labeled $a_1$, then $\overline{a}_1$ is a cut vertex for the Whitehead graph of $H$. If it doesn't cross the edge labeled $a_1$, then it has to cross the edge $a_{\alpha+i}$ (for some $1 \leq i \leq \beta - \alpha$), and thus any of $a_{\alpha+i}, \overline{a}_{\alpha+i}$ is a cut vertex for the Whitehead graph of $H$. □

Theorems 3.2.6 and 3.2.7 are a consequence of Theorem 3.2.5.

*Proof of Theorems 3.2.6 and 3.2.7.* Let $a$ be a cut vertex in the Whitehead graph of $H$.

If the connected component of $a$ does not contain $\overline{a}$, then we take the set $A$ to be that connected component (excluding $a$ itself). Otherwise, take the connected component of $a$ and remove $a$ itself: we remain with at least two nonempty connected components, and at least one of these components does not contain $\overline{a}$; take $A$ to be such a component. We consider the Whitehead automorphism $\varphi = (A, a)$.

Take a vertex $v$ in $\mathrm{core}(H)$, and notice that the letters in $L(v)$ are vertices of a complete subgraph of the Whitehead graph of $H$. Thus $L(v)$ has to be contained either in $A \cup \{a\}$ or in $A^c$. This yields the trichotomy of Theorem 3.2.7.

We now examine more in detail what happens in each of the three cases. The folding takes place according to Lemma 3.2.4. For each vertex $v$ of $\mathrm{core}(H)$, we look at the vertices which are created near $v$ in $\mathrm{subd}_\varphi(\mathrm{core}(H))$.

Case (i): $L(v) \subseteq A^c$. This means no new vertex is created near $v$. The total number of vertices remains unchanged.

Case (ii): $L(v) \subseteq A$. This means that, for every edge with endpoint $v$, a new vertex is created near $v$. All these new vertices are then folded together into a vertex $v_1$. The vertex $v$ becomes a valence-1 vertex, and can thus be removed from the graph. Thus we lose the vertex $v$ and we gain the vertex $v_1$ in the core graph: the total number of vertices is unchanged.

Case (iii): $a \in L(v)$ and $L(v) \subseteq A \cup \{a\}$. This means that $\mathrm{core}(H)$ contains an edge $e$ labeled $a$ going from $v$ to $u$. For every other edge with endpoint $v$, a new vertex is created near $v$. All these new vertices are then folded together with the vertex $u$. The vertex $v$ becomes a valence-1 vertex, and can thus be removed from the graph. The total number of vertices decreases by 1.

In each of the cases (i), (ii) and (iii), the number of vertices and edges of the core graph does not increase. Also, since the Whitehead graph contains at least an edge between $a$ and $A$, we have that case (iii) happens at least once, giving a strict decrease in the number of vertices and edges. This yields Theorem 3.2.6. □

*Remark.* We notice that, if $\mathrm{core}(H)$ has rank $r$, the number of edges of $\mathrm{core}(H)$ is the number of vertices plus $r - 1$. The same holds for $\mathrm{core}(\varphi(H))$, which has rank $r$ too. Thus the decrease in the number of vertices is the same as the decrease in the number of edges.

*Remark.* One may try to look for generalization of Remark 2: we would like to compute the decrease in the number of edges of $\mathrm{core}(H)$ by just looking the Whitehead graph of $H$; this is unfortunately not easy, because the valence of certain vertices of $H$ comes to play a role. Let $S$ be the set of vertices of $\mathrm{core}(H)$ that fall into case (iii) of the trichotomy of Theorem 3.2.7: then we have that $\|\mathrm{core}(\varphi(H))\|_\mathrm{e} = \|\mathrm{core}(H)\|_\mathrm{e} - |S|$, as we will show later in Lemma 3.2.10. Let $c(a, A)$ be the number of edges of the Whitehead graph of $H$ that connect the vertex $a$ to a vertex of $A$: the number

$c(a, A)$ can in general be different from $|S|$; in fact $c(a, A)$ and $S$ are related by $c(a, A) = \sum_{v \in S}(d(v) - 1)$ where $d(v)$ denotes the degree of a vertex $v$ of $\mathrm{core}(H)$.

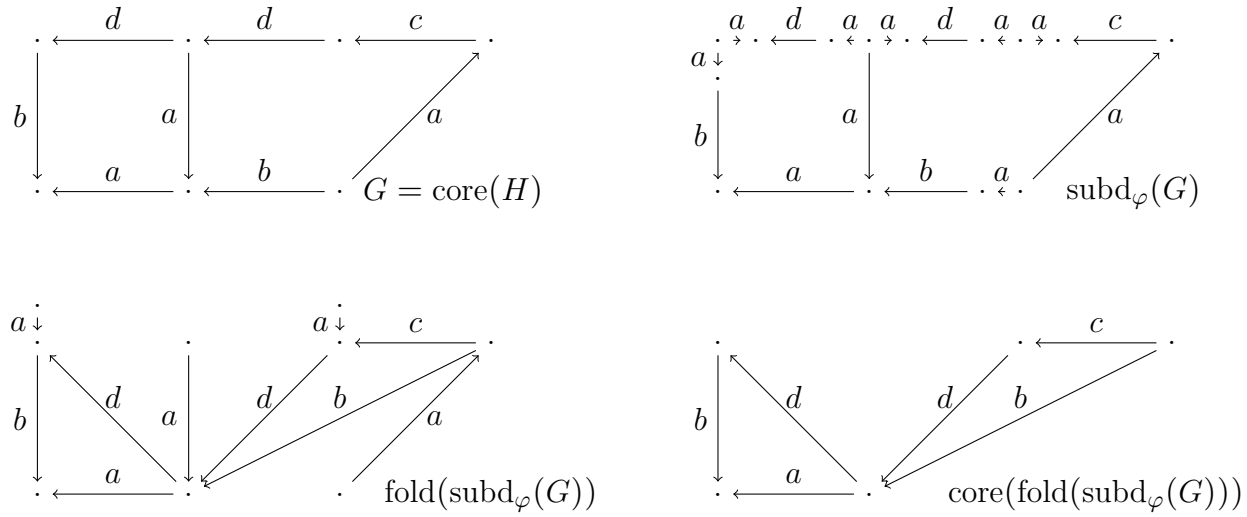


Figure 3.3: Here $F_4 = \langle a, b, c, d\rangle$ and we consider the free factor $H = \langle db\overline{a}^2, x\overline{b}acd\rangle$. The Whitehead transformation $\varphi = (\{b, \overline{c}, d, \overline{d}\}, a)$ satisfies the trichotomy of Theorem 3.2.7 for the graph $\mathrm{core}(H)$. In the figure we start with $\mathrm{core}(H)$, we subdivide it, we fold the result and we remove the valence-1 vertices: the result is $\mathrm{core}(\varphi(H))$. We observe that $\mathrm{core}(\varphi(H))$ can be obtained from $\mathrm{core}(H)$ in the following way: take the vertices of $\mathrm{core}(H)$ which fall in case (iii) of the trichotomy, and collapse to a point the $a$-edges at those vertices; Theorem 3.2.9 below makes this formal.

### 3.2.4 The quotient map

In the following, we use the notation $L(v)$ as introduced in Definition 3.2.1.

**Definition 3.2.8.** *Let $H \leq F_n$ be a finitely generated non-trivial subgroup and let $\varphi = (A, a)$ be a Whitehead automorphism. We say that the action of $\varphi$ on $H$ is **fine** if for each vertex $v \in \mathrm{core}(H)$, exactly one of the following configurations takes place:*

*(i) $L(v) \cap A = \emptyset$.*

*(ii) $L(v) \subseteq A$.*

*(iii) $a \in L(v)$ and $L(v) \subseteq A \cup \{a\}$.*

*Remark.* This is exactly the property given by the trichotomy of Theorem 3.2.7.

**Theorem 3.2.9.** *Let $H \leq F_n$ be a finitely generated non-trivial subgroup and let $\varphi = (A, a)$ be a Whitehead automorphism such that the action of $\varphi$ on $H$ is fine. Then $\mathrm{core}(\varphi(H))$ can be seen as a quotient of $\mathrm{core}(H)$ by means of a quotient map*

$q : \mathrm{core}(H) \to \mathrm{core}(\varphi(H))$ *that collapses some edges to points whilst preserving the labels and orientations on the other edges. The edges that* $q$ *collapses are exactly the* $a$*-edges going out of a vertex* $v \in G$ *that falls into case (iii) of the trichotomy of Definition 3.2.8.*

*Proof.* By Proposition 3.2.3 we have that $\mathrm{core}(\varphi(H)) = \mathrm{core}(\mathrm{fold}(\mathrm{subd}_{\varphi}(\mathrm{core}(H))))$. We examine cases (i), (ii), (iii) of the trichotomy of Definition 3.2.8 to see how $\mathrm{core}(H)$ changes when we perform the operations of subdivision, folding, and removal of valence-1 vertices.

Let $v$ be a vertex in $\mathrm{core}(H)$ which falls into case (i). Then we have that no new vertex is created near $v$, and the core graph remains unchanged.

Let $v$ be a vertex in $\mathrm{core}(H)$ which falls into case (ii). Then we have that, for each edge at $v$, a new vertex is created near $v$. All these new vertices fold together into a new vertex $v_1$, and $v$ becomes a valence-1 vertex and is thus removed from the core graph. The vertex $v_1$ takes the place of the vertex $v$, and the core graph doesn't change.

Let $v$ be a vertex in $\mathrm{core}(H)$ which falls into case (iii). We consider the unique edge $e$ labeled $a$ and going from $v$ to another vertex $u$ of $\mathrm{core}(H)$. For every other edge at $v$, we have that a new vertex is created near $v$. All these new vertices are then folded together and with $u$, and the vertex $v$ becomes a valence-1 vertex, and is thus removed from the core graph. The effect on the core graph is exactly the same as collapsing the edge $e$ to a single point.

The conclusion follows. □

The above Theorem 3.2.9 has several interesting consequences.

**Lemma 3.2.10.** *Let* $H \leq F_n$ *be a non-trivial finitely generated subgroup, and let* $\varphi = (A, a)$ *be a Whitehead automorphism such that the action of* $\varphi$ *on* $H$ *is fine. If case (iii) takes places for exactly* $p \geq 1$ *vertices* $v \in \mathrm{core}(H)$*, then* $\|\mathrm{core}(\varphi(H))\|_{\mathrm{e}} = \|\mathrm{core}(H)\|_{\mathrm{e}} - p$*. If case (iii) never happens, then* $\mathrm{core}(\varphi(H)) = \mathrm{core}(H)$ *and the restriction of* $\varphi$ *to* $H$ *is the conjugation by some element* $u \in F_n$*.*

*Proof.* Immediate from the above discussion. □

We now show that the trichotomy of Definition 3.2.8 has a nice behaviour when we pass to subgroups.

**Lemma 3.2.11.** *Let* $K \leq H \leq F_n$ *be non-trivial finitely generated subgroups, and let* $\varphi = (A, a)$ *be a Whitehead automorphism. If the action of* $\varphi$ *on* $H$ *is fine, then the action of* $\varphi$ *on* $K$ *is fine.*

*Proof of Lemma 3.2.11.* It is enough to notice that for every vertex $u \in \mathrm{core}(K)$ we have that its image $\widehat{i}(u) = v \in \mathrm{core}(H)$ satisfies $L(u) \subseteq L(v)$. Since the vertex $v$ satisfies the trichotomy of Definition 3.2.8, so does $u$. □

*Remark.* In the hypothesis of Lemma 3.2.11, we have that, for each subgroup $K \leq H$, the automorphism $\varphi$ either strictly decreases the size of $\text{core}(K)$, or it acts on $K$ as a conjugation by an element of $F_n$. We can actually be more precise than just that. Consider the map $q : \text{core}(H) \to \text{core}(\varphi(H))$, and let $G \subseteq \text{core}(H)$ be the subgraph given by the union of all the edges which are not collapsed by $q$. Observe that the inclusion $i : K \to H$ induces a locally injective label-preserving map of graphs $\widehat{i} : \text{core}(K) \to \text{core}(H)$. Then, $\varphi$ acts on $K$ as a conjugation automorphism if and only if $\widehat{i}(\text{core}(K)) \subseteq G$.

We conclude this section with a technical lemma which will be useful to us later. Let again $K \leq H \leq F_n$ be finitely generated non-trivial subgroups. Let $i : K \to H$ be the inclusion, consider the map of graphs $\widehat{i} : \text{core}(K) \to \text{core}(H)$ and consider the subgraph $\widehat{i}(\text{core}(K)) \subseteq \text{core}(H)$. For an automorphism $\varphi : F_n \to F_n$, let $j : \varphi(K) \to \varphi(H)$ be the inclusion, let $\widehat{j} : \text{core}(\varphi(K)) \to \text{core}(\varphi(H))$ be the corresponding map of graphs, and consider the subgraph $\widehat{j}(\text{core}(\varphi(K))) \subseteq \text{core}(\varphi(H))$.

**Lemma 3.2.12.** *Let $K \leq H \leq F_n$ be non-trivial finitely generated subgroups. Let $\varphi = (A, a)$ be a Whitehead automorphism such that the action of $\varphi$ on $K$ is fine. Then, with the above notation, we have $\|\widehat{j}(\text{core}(\varphi(K)))\|_e \leq \|\widehat{i}(\text{core}(K))\|_e$.*

*Proof.* Let $q : \text{core}(K) \to \text{core}(\varphi(K))$ be as in Theorem 3.2.9. Suppose we have two edges $e, e'$ in $\text{core}(K)$ such that $\widehat{i}(e) = \widehat{i}(e')$, and suppose $q$ does not collapse either of $e, e'$. Then the two edges $q(e), q(e')$ of $\text{core}(\varphi(K))$ are sent to the same edge $\widehat{j}(q(e)) = \widehat{j}(q(e'))$ of $\text{core}(\varphi(H))$. It follows that $\widehat{j}(\text{core}(\varphi(K)))$ has at most as many edges as $\widehat{i}(\text{core}(K))$, as desired. □

*Remark.* Lemma 3.2.12 becomes false if we try to count the number of vertices, instead of counting the number of edges.

### 3.2.5 A relative version of Whitehead's algorithm

**Theorem 3.2.13.** *Let $F_n = \langle a_1, ..., a_n \rangle$ and consider the free factor $\langle a_1, ..., a_k \rangle$ for $1 \leq k \leq n-1$. Let $w \in F_n$ be primitive and not a single letter. Suppose there is an automorphism $\theta : F_n \to F_n$ such that $\theta(\langle a_1, ..., a_k \rangle) = \langle a_1, ..., a_k \rangle$ and $\theta(w) = a_{k+1}$. Then there is a Whitehead automorphism $\varphi = (A, a)$ such that:*

*(i) $\varphi(a_i) = a_i$ for $i = 1, ..., k$.*

*(ii) $|\varphi(w)| < |w|$.*

*(iii) Every letter $a$, which is added to $w$ when applying $\varphi$ to $w$ letter-by-letter, immediately cancels (in the free reduction process).*

*Remark.* Notice that the word $w$ is not required to be cyclically reduced. In (ii) we mean the length and not the cyclic length. In (iii) we consider the free reduction process and not the cyclic reduction process.

For the proof, we need the following straightforward lemma:

**Lemma 3.2.14.** *Consider the inner automorphism $\gamma_a(w) = aw\overline{a}$ of $F_n$. Then for every Whitehead automorphism $(A, a)$, the identity $(A, a) = \gamma_a \circ (A^c \setminus \{a, \overline{a}\}, \overline{a})$ holds.*

*Proof of Theorem 3.2.13.* Consider the free factor $H = \langle a_1, ..., a_k \rangle * \langle w \rangle$. Notice that $\mathrm{core}(H)$ consists of $\mathrm{core}_*(w)$ together with $k$ edges from the basepoint to itself, labeled with the letters $a_1, ..., a_k$ (and here it is important that $k \geq 1$). We apply Theorems 3.2.6 and 3.2.7 to $H$ in order to get a Whitehead automorphism $\varphi = (A, a)$. If the basepoint of $\mathrm{core}(H)$ would fall into case (ii) or (iii) of the trichotomy of Theorem 3.2.7, then we apply Lemma 3.2.14 and consider the Whitehead automorphism $\varphi = (A^c \setminus \{a, \overline{a}\}, \overline{a})$ instead. Then $\varphi$ satisfies all of the desired properties. □

We observe that Theorem 3.2.13 can also be generalized to subgroups (and the proof is the same, so will be omitted).

**Theorem 3.2.15.** *Let $F_n = \langle a_1, ..., a_n \rangle$ and consider the free factor $\langle a_1, ..., a_k \rangle$ for $1 \leq k \leq n-1$. Let $H \leq F_n$ be a free factor of rank $r \geq 1$ and suppose that $\mathrm{core}_*(H)$ has at least two vertices. Suppose there is an automorphism $\theta : F_n \to F_n$ such that $\theta(\langle a_1, ..., a_k \rangle) = \langle a_1, ..., a_k \rangle$ and $\theta(H) = \langle a_{k+1}, ..., a_{k+r} \rangle$. Then there is a Whitehead automorphism $\varphi = (A, a)$ such that:*

*(i) $\varphi(a_i) = a_i$ for $i = 1, ..., k$.*

*(ii) $\|\mathrm{core}_*(\varphi(H))\|_e < \|\mathrm{core}_*(H)\|_e$.*

*(iii) The trichotomy of Theorem 3.2.7 holds at each vertex $v \in \mathrm{core}_*(H)$. Moreover, the basepoint always falls into case (i) of the trichotomy.*

## 3.3 The complex of free factors

Given a closed surface $M$ we want to study its fundamental group and its mapping class group, i.e. the group of diffeomorphisms of the surface up to isotopy. In order to study the mapping class group, several spaces have been introduced: perhaps the most important are the Teichmuller space, the space of marked hyperbolic metrics on $M$, and the curve complex, a flag simplicial complex whose vertices are the essential simple closed curves in $M$. There is an action of the mapping class group on them and this accounts for all of the symmetries. Thus the study of these spaces allows one to obtain information about the structure of the mapping class group of $M$.

Instead of a closed surface, we consider a finite graph $G$, and we want to study its fundamental group, which is a finitely generated free group $F_n$, and the group of homotopy equivalences of the graph to itself up to homotopy, which is isomorphic to the group $\mathrm{Out}(F_n)$ of the outer automorphisms of $F_n$. In analogy with the case of surfaces, it's possible to introduce several spaces associated to $F_n$, and on which

the group $\mathrm{Out}(F_n)$ acts on. Perhaps the most important are Culler-Vogtman Outer Space, see [CV86], which can be thought as the space of volume-1 metrics on reduced finite graphs with fundamental group $F_n$, and the free factor complex, which is the analogue of the curve complex.

Many results are known for the curve complex of a surface, and in particular it has been proved that it's hyperbolic and that there is an algorithm to compute the distance between two given vertices. The same questions for the free factor complex turn out to be much more difficult. The free factor complex has been proved to be hyperbolic, see [BF14], and recently a rigidity result has been obtained, stating that the isomorphisms of the complex are exactly the ones induced by the automorphisms of the free group, see [BB22]. It is very hard to prove that the free factor complex is unbounded (and the proof is non-constructive, meaning that it doesn't produce explicit examples of points at big distance in the free factor complex). The question of determining whether there is an algorithm to compute the distance between two given vertices remains open (even just for distance four).

### 3.3.1 The complex of free factors

Let $F_n$ be a free group generated by $a_1, ..., a_n$. For a subgroup $H \leq F_n$, we denote by $[H]$ the conjugacy class of that subgroup.

Define a simplicial complex $\mathrm{FF}_n$ as follows. We have a vertex for each conjugacy class $[H]$ of free factors $H \leq F_n$, with the exception of 1 and $F_n$ itself. We have an edge between $[H_0]$ and $[H_1]$ whenever there is an inclusion $H_0' \leq H_1'$ between two representatives $H_0' \in [H_0]$ and $H_1' \in [H_1]$ of the two classes (and in that case, we have a unique edge between those vertices). More generally, we have a $k$-simplex with vertices $[H_0], [H_1], ..., [H_k]$ if and only if there are inclusions $H_0' \leq H_1' \leq ... \leq H_k'$ between some representatives $H_i' \in [H_i]$ for $i = 0, ..., k$. This in particular means that our complex is a flag complex.

**Definition 3.3.1.** *Define the* ***complex of free factors*** *to be the simplicial complex* $\mathrm{FF}_n$ *defined above.*

It is shown in [BF14] that $\mathrm{FF}_n$ is connected. We would like to determine whether there is an algorithm that, given two vertices of $\mathrm{FF}_n$, gives as output their distance in a finite time, where distance is combinatorial distance in the 1-skeleton $\mathrm{FF}_n^1$. We here furnish algorithms for distances $1, 2, 3$, and also an algorithm for distance 4 when one of the free factors has rank $n - 1$.

### 3.3.2 Distance one

It is easy to check whether two conjugacy classes of free factors $[H], [K]$ are at distance 1 or not. Assume $\mathrm{rank}(H) \geq \mathrm{rank}(K)$. We look for representatives $H' \in [H]$ and

$K' \in [K]$ with an inclusion $K' \leq H'$. This is equivalent to looking for a locally injective map of graphs $\mathrm{core}(K) \to \mathrm{core}(H)$. Each such map, if it exists, is uniquely determined by the image of a given vertex; thus we only have to deal with a finite number of cases.

### 3.3.3 Distance two

We denote with $G \sqcup G'$ the disjoint union of the two graphs $G$ and $G'$.

**Proposition 3.3.2.** *Let $H, K \leq F_n$ be non-trivial free factors, and suppose that $\mathrm{core}(H) \sqcup \mathrm{core}(K)$ contains at least one edge with each label $a_i \in \{a_1, ..., a_n\}$. Suppose there are free factors $H' \in [H]$ and $K' \in [K]$ and $J \neq F_n$ such that $H', K' \leq J$. Then there is a Whitehead automorphism $\varphi$ such that*

$$\|\mathrm{core}(\varphi(H)) \sqcup \mathrm{core}(\varphi(K))\|_e < \|\mathrm{core}(H) \sqcup \mathrm{core}(K)\|_e$$

*Proof.* Since $J$ is a free factor, by a recursive application of Theorems 3.2.6 and 3.2.7, we obtain a chain of Whitehead automorphisms $\varphi_1, ..., \varphi_l$ such that $\mathrm{core}(\varphi_l \circ ... \circ \varphi_1(J))$ is a rose with labels only in $\{a_1, ..., a_{n-1}\}$. By Lemmas 3.2.11 and 3.2.10, we have that either

$$\mathrm{core}(\varphi_1(H)) \sqcup \mathrm{core}(\varphi_1(K)) = \mathrm{core}(H) \sqcup \mathrm{core}(K)$$

or

$$\|\mathrm{core}(\varphi_1(H)) \sqcup \mathrm{core}(\varphi_1(K))\|_e < \|\mathrm{core}(H) \sqcup \mathrm{core}(K)\|_e$$

If $\mathrm{core}(\varphi_1(H)) \sqcup \mathrm{core}(\varphi_1(K)) = \mathrm{core}(H) \sqcup \mathrm{core}(K)$ then we repeat the reasoning with $\varphi_2$ instead of $\varphi_1$; and so on. If $\mathrm{core}(\varphi_l \circ ... \circ \varphi_1(H)) \sqcup \mathrm{core}(\varphi_l \circ ... \circ \varphi_1(K)) = \mathrm{core}(H) \sqcup \mathrm{core}(K)$ then we have a contradiction, since $\mathrm{core}(\varphi_l \circ ... \circ \varphi_1(H)) \sqcup \mathrm{core}(\varphi_l \circ ... \circ \varphi_1(K))$ only contains edges with the labels $\{a_1, ..., a_{n-1}\}$ while $\mathrm{core}(H) \sqcup \mathrm{core}(K)$ contains edges with all possible labels by hypothesis. So we can take the smallest $m$ such that $\mathrm{core}(\varphi_m(H)) \sqcup \mathrm{core}(\varphi_m(K)) \neq \mathrm{core}(H) \sqcup \mathrm{core}(K)$, and the Whitehead automorphism $\varphi_m$ satisfies the thesis. □

Let $[H], [K]$ be conjugacy classes of non-trivial free factors. We want to check (i) whether or not there are representatives with non-trivial intersection and (ii) whether or not there are representatives contained in a common proper free factor.

For (i) we use Theorem 1.2.21: there are representatives $H' \in [H]$ and $K' \in [K]$ with non-trivial intersection if and only if one of the connected components of the pullback of the two graphs $\mathrm{core}(H)$ and $\mathrm{core}(K)$ contains a non-trivial cycle.

For (ii) we apply Proposition 3.3.2 repeatedly. If $\mathrm{core}(H) \sqcup \mathrm{core}(K)$ contains only edges with labels from $\{a_1, ..., a_{n-1}\}$ then they are at distance two; otherwise we look for a Whitehead transformation which strictly reduces the number of vertices of $\mathrm{core}(H) \sqcup \mathrm{core}(K)$: if we find it then we apply it and reiterate the reasoning, otherwise it means that there are no representatives contained in a common free factor.

### 3.3.4 Distance three

Given two conjugacy classes of free factors $[H], [K]$, we want to check whether there are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $I, J$ such that $H', J \leq I$ and $J \leq K'$ (there is also a symmetric check to do, but it is completely analogous).

**Definition 3.3.3.** *Define the finite oriented graph $\Theta = \Theta(H, K)$ as follows.*

*(i) We have one vertex for each pair of connected core folded labeled graphs $(A, B)$ such that $\|A\|_e \leq \|\text{core}(H)\|_e$ and $\|B\|_e \leq \|\text{core}(K)\|_e$.*

*(ii) Suppose we are given vertices $(A, B), (C, D)$ of $\Theta$ and a Whitehead automorphism $\varphi$ such that $C = \text{core}(\text{fold}(\text{subd}_\varphi(A)))$ and $D$ is isomorphic to a subgraph of $\text{core}(\text{fold}(\text{subd}_\varphi(B)))$. Then we add an edge going from $(A, B)$ to $(C, D)$, and we label that edge with the automorphism $\varphi$.*

**Proposition 3.3.4.** *Let $[H], [K]$ be conjugacy classes of free factors. Then the following are equivalent:*

*(i) There are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $I, J$ such that $H', J \leq I$ and $J \leq K'$.*

*(ii) There is a path in $\Theta$ going from a vertex of the form $(\text{core}(H), B)$ for some subgraph $B$ of $\text{core}(K)$ to a vertex of the form $(A', B')$ such that $A' \sqcup B'$ only uses labels from a proper subset $P \subset \{a_1, ..., a_n\}$.*

*Proof.* (i) $\Rightarrow$ (ii) Suppose there are non-trivial free factors $I, J$ such that $H', J \leq I$ and $J \leq K'$. The inclusion $j : J \to K'$ gives a locally injective map $\widehat{j} : \text{core}(J) \to \text{core}(K)$, and thus a subgraph $\widehat{j}(\text{core}(J)) \subseteq \text{core}(K)$: the pair $(\text{core}(H), \widehat{j}(\text{core}(J)))$ is a vertex of $\Theta$. If $\text{core}(I)$ contains only edges with labels from a proper subset of $\{a_1, ..., a_n\}$, then we see that the pair of graphs $(\text{core}(H), \widehat{j}(\text{core}(J)))$ only contains edges with those labels too, and we are done.

If $\text{core}(I)$ contains at least one edge with each label, then by Theorems 3.2.6 and 3.2.7 there is a Whitehead automorphism $\varphi$ such that $\|\text{core}(\varphi(I))\|_e < \|\text{core}(I)\|_e$, and such that for each vertex in $\text{core}(I)$ the trichotomy of Theorem 3.2.7 holds. In particular, by Lemmas 3.2.10, 3.2.11, 3.2.12, we have that the number of edges of $\text{core}(H)$ and of $\widehat{j}(\text{core}(J))$ doesn't increase either. This means that the pair $(\text{core}(\varphi(H)), \widehat{j}(\text{core}(\varphi(J))))$ is a vertex of $\Theta$, and that $\Theta$ contains an edge labeled $\varphi$ and going from $(\text{core}(H), \widehat{j}(\text{core}(J)))$ to $(\text{core}(\varphi(H)), \widehat{j}(\text{core}(\varphi(J))))$ (here we are using Proposition 3.2.3).

We now reiterate the same reasoning. By Theorems 3.2.6 and 3.2.7, we can take a finite sequence of Whitehead automorphisms $\varphi_1, ..., \varphi_l$ such that $\varphi_i$ strictly reduces the number of edges of $\text{core}(\varphi_{i-1} \circ ... \circ \varphi_1(I))$, such that for each vertex of $\text{core}(\varphi_{i-1} \circ ... \circ \varphi_1(I))$ the trichotomy of Theorem 3.2.7 holds, and such that $\varphi_l \circ ... \circ \varphi_1(I)$ only contains edges with labels from a proper subset $P \subset \{a_1, ..., a_n\}$. Then this produces a path in $\Theta$ with vertices $(\text{core}(\varphi_i \circ ... \circ \varphi_1(H)), \widehat{j}(\text{core}(\varphi_i \circ ... \circ \varphi_1(J))))$ and which

goes from the pair $(\mathrm{core}(H), \widehat{j}(\mathrm{core}(J)))$ to a pair containing only edges with labels in $P$, and we are done.

(ii) $\Rightarrow$ (i) Suppose there is a path in $\Theta$ with vertices $(A_1, B_1), ..., (A_l, B_l)$ and with an edge labeled $\varphi_i$ going from $(A_i, B_i)$ to $(A_{i+1}, B_{i+1})$, such that $A_1 = \mathrm{core}(H)$ and $B_1$ is a subgraph of $\mathrm{core}(K)$, and such that $A_l, B_l$ only contain edges with labels in a propert subset $P \subset \{a_1, ..., a_n\}$. Then we fix basepoints in $A_l$ and $B_l$ and we set $\psi = \varphi_1^{-1} \circ ... \circ \varphi_l^{-1}$: we get a segment of length three in $\mathrm{FF}_n^1$ connecting $[H]$ and $[K]$, with vertices $[H] = [\psi(\pi_1(A_l))]$ and $[I] = [\psi(\langle a_1, ..., a_{n-1}\rangle)]$ and $[J] = [\psi(\pi_1(B_l))]$ and $[K]$. □

Since the graph $\Theta$ is finite, there is an algorithm that tells us whether such a path in $\Theta$ exists or not. This provides an algorithm to check whether two vertices of $\mathrm{FF}_n$ are at distance three or not.

### 3.3.5 About distance four

We would like to check whether two conjugacy classes of free factors $[H], [K]$ are at distance at most four in $\mathrm{FF}_n$. In order to achieve this, we need to check two conditions:

1. Whether or not there are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $J_1, J_2, J_3$ such that $J_1 \leq H'$ and $J_1, J_3 \leq J_2$ and $J_3 \leq K'$.

2. Whether or not there are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $J_1, J_2, J_3$ such that $H', J_2 \leq J_1$ and $J_2, K' \leq J_3$.

We here provide an algorithm to check condition 1. Notice that, in the particular case when $\mathrm{rank}(H) = n - 1$, condition 2 reduces to checking distance three; it follows that, when one of the free factors has rank $n - 1$, we have an algorithm to check whether they are at distance four or not.

**Definition 3.3.5.** *Define the finite oriented graph $\Omega = \Omega(H, K)$ as follows.*

*(i) We have one vertex for each pair of connected core folded labeled graphs $(A, B)$ such that $\|A\|_e \leq \|\mathrm{core}(H)\|_e$ and $\|B\|_e \leq \|\mathrm{core}(K)\|_e$.*

*(ii) Suppose we are given vertices $(A, B), (C, D)$ of $\Omega$ and a Whitehead automorphism $\varphi$ such that $C$ is isomorphic to a subgraph of $\mathrm{core}(\mathrm{fold}(\mathrm{subd}_\varphi(A)))$ and $D$ is isomorphic to a subgraph of $\mathrm{core}(\mathrm{fold}(\mathrm{subd}_\varphi(B)))$. Then we add an edge going from $(A, B)$ to $(C, D)$, and we label that edge with the automorphism $\varphi$.*

**Proposition 3.3.6.** *Let $[H], [K]$ be conjugacy classes of free factors. Then the following are equivalent:*

*(i) There are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $J_1, J_2, J_3$ such that $J_1 \leq H'$ and $J_1, J_3 \leq J_2$ and $J_3 \leq K'$.*

*(ii) There is a path in $\Omega$ going from a vertex of the form $(A, B)$ for some subgraph $A$ of $\mathrm{core}(H)$ and some subgraph $B$ of $\mathrm{core}(K)$ to a vertex of the form $(A', B')$ such that $A' \sqcup B'$ only uses labels from a proper subset $P \subset \{a_1, ..., a_n\}$.*

*Proof.* (i) $\Rightarrow$ (ii) Suppose there are representatives $H' \in [H]$ and $K' \in [K]$ and non-trivial free factors $J_1, J_2, J_3$ such that $J_1 \leq H'$ and $J_1, J_3 \leq J_2$ and $J_3 \leq K'$. By means of Theorems 3.2.6 and 3.2.7, we take a chain of Whitehead automorphisms $\varphi_1, ..., \varphi_l$ such that $\varphi_{i+1}$ strictly reduces the number of edges of $\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(J_2))$, and such that the trichotomy of Theorem 3.2.7 holds too. By Lemmas 3.2.10, 3.2.11, 3.2.12, we have that this produces a path $(A_i, B_i)$ in $\Omega$, where $A_i$ is the image the map $\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(J_1)) \to \mathrm{core}(\varphi_i \circ ... \circ \varphi_1(H))$ induced by the inclusion $J_1 \leq K'$, and $B_i$ is the image of the map $\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(J_3)) \to \mathrm{core}(\varphi_i \circ ... \circ \varphi_1(K))$ induced by the inclusion $J_3 \leq K'$. The starting point $(A_1, B_1)$ of the path is given by two subgraphs of $\mathrm{core}(H)$ and $\mathrm{core}(K)$ respectively, and the endpoint $(A_l, B_l)$ has the property that $A_l \sqcup B_l$ only contains edges with labels from a proper subset $P \subset \{a_1, ..., a_n\}$.
(ii) $\Rightarrow$ (i) Suppose there is a path $(A_1, B_1), ..., (A_l, B_l)$ in $\Omega$ with an edge from $(A_i, B_i)$ to $(A_{i+1}, B_{i+1})$ labeled $\varphi_i$, and such that $A_1, B_1$ are subgraphs of $\mathrm{core}(H), \mathrm{core}(K)$ respectively, and $A_l \sqcup B_l$ contains only edges with labels in a proper subset $P \subset \{a_1, ..., a_n\}$. Then we fix basepoints in $A_l$ and $B_l$, we set $\psi = \varphi_1^{-1} \circ ... \circ \varphi_l^{-1}$, and we produce the free factors $J_1 = \psi(\pi_1(A_l))$ and $J_2 = \psi(\langle P \rangle)$ and $J_3 = \psi(\pi_1(B_l))$. For these free factors, there are representatives $H' \in [H]$ and $K' \in [K]$ such that $J_1 \leq H'$ and $J_1, J_3 \leq J_2$ and $J_3 \leq K'$, as desired. □

Since the graph $\Omega$ is finite, we obtain an algorithm to check condition 1.

## 3.4 Echelon subgroups of a free group

Given an automorphism $\varphi$ of a finitely generated free group $F_n$, one of the most important objects to study is the subgroup $\mathrm{Fix}(\varphi) = \{w \in F_n : \varphi(w) = w\}$ of its fixed points. In [Ger87] Gersten showed that $\mathrm{Fix}(\varphi)$ is always finitely generated. Bestvina and Handel introduced the notion of train-track maps, see [BH92], providing a much deeper insight on the structure of $\mathrm{Fix}(\varphi)$, and proving that $\mathrm{rank}(\mathrm{Fix}(\varphi)) \leq n$. More generally, it turns out that fixed-point subgroups behave very well with respect to intersections: in [DV96] W. Dicks and E. Ventura introduce the notion of *inert subgroup*, i.e. a subgroup $H \leq F_n$ such that $\mathrm{rank}(H \cap K) \leq \mathrm{rank}(K)$ for every other subgroup $K \leq F_n$, and they show that the subgroup $\mathrm{Fix}(\varphi)$ is inert, generalizing the result of Bestvina and Handel. An algorithm that given the automorphism $\varphi$ computes $\mathrm{Fix}(\varphi)$ was found later, see [BM12]; the converse problem, of determining whether a given subgroup $H \leq F_n$ is the fixed-point subgroup of some automorphism, remains open. Some partial results can be found in the literature, based on the train

track maps of Bestvina and Handel, but they only go in one direction, showing that if $H$ is a fixed-point subgroup then $H$ has to be of a certain form, see [MV04].

Recently, A. Rosenmann introduced the notion of echelon subgroup of a free group, see Definitions 3.4.6 and 3.4.8, in analogy with the concept of matrix in echelon form in linear algebra. It follows immediately from the normal form of Martino and Ventura [MV04] that fixed-point subgroups of automorphisms of $F_n$ are echelon. In [Ros13] Rosenmann shows that echelon subgroups are inert, i.e. enjoy the same property of good behaviour with respect to intersections as fixed-point subgroups do. This places the property of being echelon as an intermediate property in the chain

$$\text{fixed-point subgroup of an automorphism} \Rightarrow \text{echelon} \Rightarrow \text{inert}$$

In his paper [Ros13] Rosenmann asks whether there is an algorithm that determines whether a given subgroup is echelon or not. We here give an affirmative answer to that question, and in Section 3.4.3 we provide such an algorithm. In Section 3.4.4 we prove, by means of a counterexample, that the intersection of two echelon subgroups needs not to be echelon; this answers in the negative to another question of Rosenmann.

### 3.4.1 The free factor support of a subgroup

Let $F_n$ be a free group with basis $a_1, ..., a_n$.

**Proposition 3.4.1.** *Let $H \leq F_n$ be a subgroup. Then there is a unique free factor $B \leq F_n$ containing $H$ of minimum rank (the free factor $B$ is minimum by inclusion among the free factors containing $H$).*

*Proof.* Let $B \leq F_n$ be a free factor containing $H$ and such that $B$ has minimum rank among the free factors containing $H$. Let now $B' \leq F_n$ be any other free factor containing $H$: by Corollary 2.1.7 we have that $B \cap B' \leq F_n$ is a free factor containing $H$. But by Proposition 2.1.4 we have that $B \cap B'$ is a free factor in $B$ too, implying that $\text{rank}(B \cap B') \leq \text{rank}(B)$. But $B$ has minimum rank among the free factors containing $H$, and thus we must have $\text{rank}(B \cap B') = \text{rank}(B)$, which gives $B \cap B' = B$. Thus $B$ is minimum by inclusion among the free factors containing $H$ (and in particular, it is the unique free factor of minimum rank containing $H$). □

**Definition 3.4.2.** *For a subgroup $H \leq F_n$ define the **free factor support** of $H$ to be the free factor $B = \text{ffs}(H)$ given by Proposition 3.4.1.*

Given a non-trivial finitely generated subgroup $H \leq F_n$, it's possible to algorithmically compute the free factor $\text{ffs}(H)$, as we now explain.

**Proposition 3.4.3.** *Let $H \leq F_n$ be a non-trivial finitely generated subgroup. Suppose the edges of $\text{core}(H)$ are labeled with at least $\text{rank}(\text{ffs}(H)) + 1$ different letters. Then there is a Whitehead automorphism $\varphi$ such that $\|\text{core}(\varphi(H))\|_e < \|\text{core}(H)\|_e$.*

*Proof.* Let $B = \mathrm{ffs}(H)$. By an iterated application of Theorems 3.2.6 and 3.2.7, we can find a sequence of Whitehead automorphisms $\varphi_1, ..., \varphi_k$ such that the following holds:

(i) The action of $\varphi_i$ on $\varphi_{i-1} \circ ... \circ \varphi_1(B)$ is fine for $i = 1, ..., k$.

(ii) We have $\|\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(B))\|_e < \|\mathrm{core}(\varphi_{i-1} \circ ... \circ \varphi_1(B))\|_e$ for $i = 1, ..., k$.

(iii) $\mathrm{core}(\varphi_k \circ ... \circ \varphi_1(B))$ is a one-vertex graph with $\mathrm{rank}(B)$ edges.

From (i) and from Lemma 3.2.11, we deduce that the action of $\varphi_i$ on $\varphi_{i-1} \circ ... \circ \varphi_1(H)$ is fine for $i = 1, ..., k$. From (iii) we obtain that $\mathrm{core}(\varphi_k \circ ... \circ \varphi_1(H))$ contains edges with at most $\mathrm{rank}(B)$ different labels, and thus must be different from $\mathrm{core}(H)$. From (ii) and from Lemma 3.2.10, we have that for each $i = 1, ..., k$ either

$$\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(H)) = \mathrm{core}(\varphi_{i-1} \circ ... \circ \varphi_1(H))$$

or

$$\|\mathrm{core}(\varphi_i \circ ... \circ \varphi_1(H))\|_e < \|\mathrm{core}(\varphi_{i-1} \circ ... \circ \varphi_1(H))\|_e$$

In particular, we can take $l \geq 1$ to be the smallest such that $\mathrm{core}(\varphi_l \circ ... \circ \varphi_1(H)) \neq \mathrm{core}(\varphi_{l-1} \circ ... \circ \varphi_1(H))$. Then we have that $\mathrm{core}(\varphi_{l-1} \circ ... \circ \varphi_1(H)) = \mathrm{core}(H)$ and the Whitehead automorphism $\varphi = \varphi_l$ satisfies the thesis. □

**Theorem 3.4.4.** *There is an algorithm which takes as input a non-trivial finitely generated subgroup $H \leq F_n$ and gives as output the subgroup* $\mathrm{ffs}(H)$.

*Algorithm.* We look for a Whitehead automorphism $\varphi$ such that $\|\mathrm{core}(\varphi(H))\|_e < \|\mathrm{core}(H)\|_e$; since there is only a finite number of Whitehead automorphisms, we can to this in a bounded amount of time. If we find such $\varphi$, then we replace $H$ with $\varphi(H)$, and reiterate the process. Since the number of edges of $\mathrm{core}(H)$ is strictly decreasing, we eventually stop. We end up with a sequence of Whitehead automorphisms $\varphi_1, ..., \varphi_k$ such that $\|\mathrm{core}(\varphi_k \circ ... \circ \varphi_1(H))\|_e$ can't be reduced by means of a Whitehead automorphism. Let $S \subseteq \{a_1, ..., a_n\}$ be the set of all the letters which appear as labels of at least one edge of $\mathrm{core}(\varphi_k \circ ... \circ \varphi_1(H))$. Up to conjugation, we have $\varphi_k \circ ... \circ \varphi_1(H) \leq \langle S \rangle$ and thus $\mathrm{ffs}(\varphi_k \circ ... \circ \varphi_1(H)) \leq \langle S \rangle$. If the inclusion is strict, then $\mathrm{rank}(\mathrm{ffs}(\varphi_k \circ ... \circ \varphi_1(H))) < |S|$, and we can apply Proposition 3.4.3 and find a Whitehead automorphism reducing the number of edges of $\mathrm{core}(\varphi_k \circ ... \circ \varphi_1(H))$, contradiction. This shows that $\mathrm{ffs}(\varphi_k \circ ... \circ \varphi_1(H)) = \langle S \rangle$ and thus $\mathrm{ffs}(H) = \varphi_1^{-1} \circ ... \circ \varphi_k^{-1}(\langle S \rangle)$, giving an explicit basis for $\mathrm{ffs}(H)$. □

### 3.4.2 Echelon subgroups

Let $F_n$ be a free group with basis $a_1, ..., a_n$. Let $H \leq F_n$ be a finitely generated subgroup and denote $r = \mathrm{rank}(H)$.

**Definition 3.4.5.** *A **flag** for $H$ is a chain of free factors $B_1 \leq ... \leq B_r \leq F_n$ such that*

$$\mathrm{rank}(H \cap B_i) = i \qquad \text{for } i = 1, ..., r.$$

*A flag for $H$ is called* ***minimal*** *if it satisfies* $\mathrm{ffs}(H \cap B_i) = B_i$ *for* $i = 1, ..., r$.

**Definition 3.4.6.** *An* ***echelon basis*** *for $H$ is a (ordered) basis $b_1, ..., b_n$ for $F_n$ such that*

$$\mathrm{rank}(H \cap \langle b_1, ..., b_i \rangle) \leq \mathrm{rank}(H \cap \langle b_1, ..., b_{i-1} \rangle) + 1 \qquad \text{for } i = 1, ..., n.$$

The above definition takes inspiration from the notion of matrix in echelon form, coming from linear algebra. The following lemma shows that a subgroup admits an echelon basis if and only if it admits a (minimal) flag.

**Lemma 3.4.7.** *Let $H \leq F_n$ be a finitely generated subgroup with* $\mathrm{rank}(H) = r$. *Then the following are equivalent:*

*(i) $H$ has an echelon basis.*

*(ii) $H$ has a flag.*

*(iii) $H$ has a minimal flag.*

*Proof.* Denote $r = \mathrm{rank}(H)$.

(i)⇒(ii) Let $b_1, ..., b_n$ be an echelon basis for $H$. By Proposition 2.1.4 we have that $H \cap \langle b_1, ..., b_{j-1} \rangle$ is a free factor in $H \cap \langle b_1, ..., b_j \rangle$, implying that $\mathrm{rank}(H \cap \langle b_1, ..., b_{j-1} \rangle) \leq \mathrm{rank}(H \cap \langle b_1, ..., b_j \rangle)$. Thus we have a non-decreasing sequence of integers $0 \leq \mathrm{rank}(H \cap \langle b_1 \rangle) \leq \mathrm{rank}(H \cap \langle b_1, b_2 \rangle) \leq ... \leq \mathrm{rank}(H \cap \langle b_1, ..., b_{n-1} \rangle) \leq r$, and since $H$ is echelon we have that at each step the sequence increases by at most one. For each $i = 1, ..., r$ we can thus choose an index $j(i)$ such that $\mathrm{rank}(H \cap \langle b_1, ..., b_{j(i)} \rangle) = i$. Define the free factor $B_i = \langle b_1, ..., b_{j(i)} \rangle$, and notice that $B_1 \leq ... \leq B_{r-1} \leq B_r$ and $\mathrm{rank}(H \cap B_i) = i$ for $i = 1, ..., r$.

(ii)⇒(iii) Let $B_1 \leq ... \leq B_r \leq F_n$ be a flag for $H$. We define the free factor $B'_i = \mathrm{ffs}(H \cap B_i)$ for $i = 1, ..., r$. Since $B_i$ is a free factor containing $H \cap B_i$, we must have $B'_i \leq B_i$, and thus also $H \cap B'_i \leq H \cap B_i$. Since both $H$ and $B'_i$ contain $H \cap B_i$, we must have $H \cap B_i \leq H \cap B'_i$. It follows that $H \cap B'_i = H \cap B_i$, and in particular $\mathrm{ffs}(H \cap B'_i) = B'_i$ and $\mathrm{rank}(H \cap B'_i) = i$ for $i = 1, ..., r$. To conclude, notice that from the definition it's immediate that $B'_1 \leq ... \leq B'_r$. Thus the free factors $B'_1, ..., B'_r$ are a minimal flag for $H$.

(iii)⇒(i) Let $B_1 \leq ... \leq B_r \leq F_n$ be a minimal flag for $H$. Choose an ordered basis for $B_1$, extend it to an ordered basis for $B_2$, and so on. We obtain an ordered basis for $F_n$, and it's easy to see that this is an echelon basis for $H$. □

Notice that, if $B_1 \leq ... \leq B_r$ is a minimal flag for $H$, then we have $\mathrm{rank}(H \cap B_r) = \mathrm{rank}(H)$. But by Proposition 2.1.4 we have that $H \cap B_r$ is a free factor in $H$, and thus this implies $H \cap B_r = H$, yielding $H \leq B_r$ and $\mathrm{ffs}(H) = B_r$.

Notice that the proof of Lemma 3.4.7 is constructive, thanks to Theorem 3.4.4. This means that, from an algorithmic point of view, it is equivalent to have a basis with respect to which $H$ is echelon, or a (minimal) flag for $H$; in fact, from one we can compute the other.

**Definition 3.4.8.** *We say that a finitely generated subgroup $H \leq F_n$ is* ***echelon*** *if it admits an echelon basis.*

The above Lemma 3.4.7 gives the following corollary, stating that the property of being echelon is somewhat independent on the ambient free group (but not entirely, see Proposition 3.4.16).

**Corollary 3.4.9.** *Let $H \leq F_n$ be a finitely generated subgroup. Let $A \leq F_n$ be a free factor containing $H$. Then $H$ is echelon in $F_n$ if and only if $H$ is echelon in $A$.*

### 3.4.3 An algorithm to recognize echelon subgroups

Let $F_n$ be a free group with basis $a_1, ..., a_n$. The aim of this section is to provide an algorithm that, given a finitely generated subgroup $H \leq F_n$, tells us whether $H$ is echelon or not, and in case of an affirmative answer also computes an echelon basis for $H$. This answers a question asked by A. Rosenmann in [Ros13].

**Lemma 3.4.10.** *Let $K \leq H \leq F_n$ be finitely generated subgroups. Suppose $K$ is a free factor in $H$ with* $\mathrm{rank}(K) = \mathrm{rank}(H) - 1$. *Let $i : K \to H$ be the inclusion map and let $\widehat{i} : \mathrm{core}(K) \to \mathrm{core}(H)$ be the induced map between their core graphs. Then $\widehat{i}$ is either injective or surjective.*

*Proof.* Up to conjugation, we can assume that the basepoint of $\mathrm{core}_*(K)$ belongs to $\mathrm{im}(\widehat{i})$. Let $K' = \pi_1(\mathrm{im}(\widehat{i}), *)$ and notice that by Lemma 2.1.3 we have that $K' \leq H$ is a free factor; moreover we have an inclusion $K \leq K'$, induced by the map $\widehat{i} : \mathrm{core}_*(K) \to \mathrm{core}_*(K')$. Suppose $\widehat{i}$ isn't surjective: then we have $\mathrm{rank}(K') \leq \mathrm{rank}(H) - 1 = \mathrm{rank}(K)$. But for free factors of $H$, the inclusion $K \leq K'$ and the inequality $\mathrm{rank}(K') \leq \mathrm{rank}(K)$ force $K' = K$. It follows that $\mathrm{core}(K) = \mathrm{core}(K') = \mathrm{im}(\widehat{i})$ is a subgraph of $\mathrm{core}(H)$, and thus $\widehat{i}$ is injective, as desired. □

The algorithm we provide will be recursive in nature: the key step for the recursion is given by the following Lemma 3.4.11.

**Lemma 3.4.11.** *Let $H \leq F_n$ be a non-trivial finitely generated subgroup. Then the following are equivalent:*

*(i) $H$ is echelon.*

*(ii) There is a free factor $A \leq \mathrm{ffs}(H)$ such that* $\mathrm{rank}(H \cap A) = \mathrm{rank}(H) - 1$ *and $H \cap A$ is echelon.*

*Proof.* Let $r = \mathrm{rank}(H)$.

(i)⇒(ii). Suppose $H$ is echelon, and let $B_1 \leq ... \leq B_r \leq F_n$ be a minimal flag for $H$, see Lemma 3.4.7. In particular we must have $H \leq B_r$ and $\mathrm{ffs}(H) = B_r$. Define $A = B_{r-1}$ and we have that $A \leq \mathrm{ffs}(H)$ and $\mathrm{rank}(H \cap A) = r - 1$. We

have to prove that $H \cap A$ is echelon: but we notice that the chain of free factors $B_1 \leq ... \leq B_{r-1} \leq F_n$ is a flag for $H$, and thus we are done by Lemma 3.4.7.

(ii)⇒(i). Suppose we have a free factor $A \leq \text{ffs}(H)$ such that $\text{rank}(H \cap A) = r - 1$ and such that $H \cap A$ is echelon. By Lemma 3.4.7 we can find a flag $B_1 \leq ... \leq B_{r-1}$ such that $\text{rank}((H \cap A) \cap B_i) = i$ and $\text{ffs}((H \cap A) \cap B_i) = B_i$ for $i = 1, ..., r-1$. We define $B_r = \text{ffs}(H)$ and we notice that $B_{r-1} = \text{ffs}(H \cap A \cap B_{r-1}) \leq \text{ffs}(H) = B_r$, so that we have a chain of free factors $B_1 \leq ... \leq B_{r-1} \leq B_r$. Since $A$ is a free factor, for $i = 1, ..., r-1$ we have $B_i = \text{ffs}(H \cap A \cap B_i) \leq A$ and thus $H \cap A \cap B_i = H \cap B_i$, implying that $\text{rank}(H \cap B_i) = i$. Of course we also have $\text{rank}(H \cap B_r) = r$. The conclusion follows by Lemma 3.4.7. □

We are now looking for free factors $A \leq \text{ffs}(H)$ such that $\text{rank}(H \cap A) = \text{rank}(H) - 1$. In order to do this, we make use of Theorem 3.2.7 and of Lemmas 3.2.11 and 3.2.12, which allow us to look instead for paths inside the finite graphs $\Lambda_r(E)$, defined as follows.

**Definition 3.4.12.** *For $r, E \geq 1$ integers, define the finite oriented graph $\Lambda_r(E)$ as follows.*

*(i) We have one vertex for each connected core folded labeled graph $G$ with $\text{rank}(G) = r$ and $\|G\|_e \leq E$.*

*(ii) Suppose we are given two vertices $G, G'$ of $\Lambda_r(E)$ and a Whitehead automorphism $\varphi$ such that $\|G'\|_e \leq \|G\|_e$ and $\pi_1(G', *'), \varphi(\pi_1(G, *))$ are conjugated subgroups of $F_n$ (and this is independent on the chosen basepoints). Then we add an edge going from $G$ to $G'$, and we label that edge with the automorphism $\varphi$.*

**Proposition 3.4.13.** *Let $H \leq F_n$ be a non-trivial finitely generated subgroup with $\text{ffs}(H) = F_n$. Let $r = \text{rank}(H)$ and let $E = \|\text{core}(H)\|_e$. Then $H$ is echelon if and only if there are vertices $G_1$ of $\Lambda_r(E)$ and $G_2, G_3$ of $\Lambda_{r-1}(E)$ such that the following conditions hold:*

*(i) There is a path in $\Lambda_r(E)$ from $\text{core}(H)$ to $G_1$.*

*(ii) $G_2$ is a subgraph of $\text{core}(\varphi(\pi_1(G_1, *)))$ for some Whitehead automorphism $\varphi$ (this doesn't depend on the choice of the basepoint).*

*(iii) There is a path in $\Lambda_{r-1}(E)$ from $G_2$ to $G_3$.*

*(iv) The edges of $G_3$ use only labels from a proper subset $S \subset \{a_1, ..., a_n\}$.*

*(v) The group $\pi_1(G_3, *)$ is echelon (this doesn't depend on the choice of the basepoint).*

*Proof.*

(⇒) Suppose first that $H$ is echelon, and we want to find $G_1, G_2, G_3$ satisfying (i)-(v). By Lemma 3.4.11, let $A \leq F_n$ be a free factor such that $\text{rank}(H \cap A) = r - 1$ and $H \cap A$ is echelon. By an iterated application of Theorems 3.2.6 and 3.2.7, we find a sequence of Whitehead automorphisms $\varphi_1, ..., \varphi_l$ such that for every $k = 0, ..., l - 1$

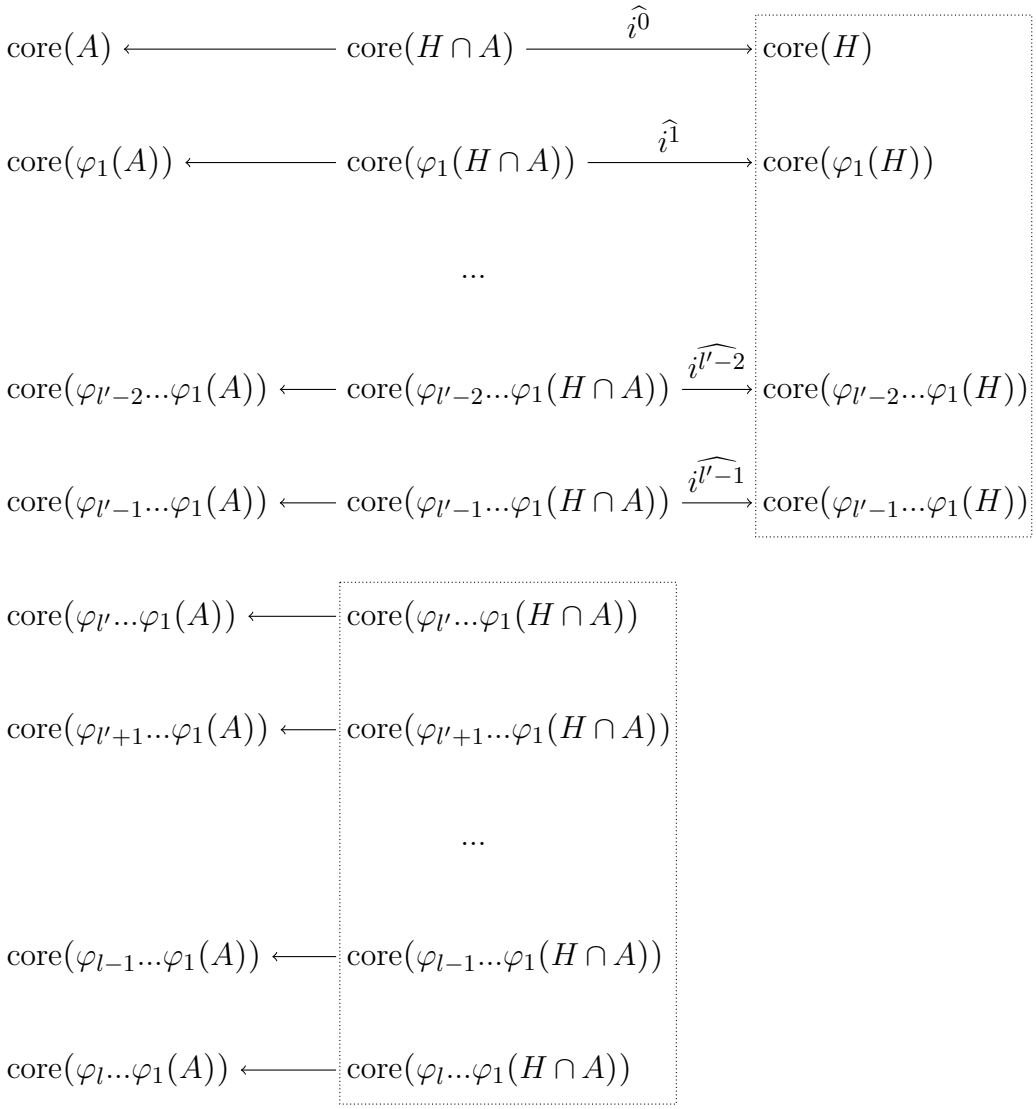


Figure 3.4: A diagram of how the situation in the proof of Proposition 3.4.13 changes as we apply the sequence of Whitehead automorphisms $\varphi_1, ..., \varphi_l$. In the column on the left we can see the graph core$(A)$, which decreases its size at each step, until we get a rose core$(\varphi_l ... \varphi_1(A))$. In the central column we have core$(H \cap A)$ and in the right column we have core$(H)$. The map $\widehat{i^k}$ is surjective for $k = 0, ..., l' - 1$. The upper box describes a path in $\Lambda_r(E)$ from core$(H)$ to $G_1$. The lower box describes a path in $\Lambda_{r-1}(E)$ from $G_2$ to $G_3$.

the action of $\varphi_{k+1}$ on $\varphi_k \circ ... \circ \varphi_1(A)$ is fine, and such that core$(\varphi_l \circ ... \circ \varphi_1(A))$ is a graph with one vertex. Without loss of generality, we can assume that the edges of core$(\varphi_l \circ ... \circ \varphi_1(A))$ are labeled with the elements of $S$ for some proper subset $S \subset \{a_1, ..., a_n\}$.
For $k = 0, ..., l$ let $i^k : \varphi_k \circ ... \circ \varphi_1(H \cap A) \to \varphi_k \circ ... \circ \varphi_1(H)$ be the inclusion map,

and let $\widehat{i^k} : \text{core}(\varphi_k \circ ... \circ \varphi_1(H \cap A)) \to \text{core}(\varphi_k \circ ... \circ \varphi_1(H))$ be the induced map between core graphs, see Figure 3.4. By Lemma 3.2.11 the action of $\varphi_{k+1}$ on $\varphi_k \circ ... \circ \varphi_1(H \cap A)$ is fine. By Lemma 3.2.10 we have $\|\text{core}(\varphi_{k+1} \circ ... \circ \varphi_1(H \cap A))\|_e \leq \|\varphi_k \circ ... \circ \varphi_1(H)\|_e$ for $k = 0, ..., l-1$. By Lemma 3.2.12 we have $\|\text{im}(\widehat{i^{k+1}})\|_e \leq \|\text{im}(\widehat{i^k})\|_e$ for $k = 0, ..., l-1$.

By Lemma 3.4.10, we have that $\widehat{i^k}$ is either surjective or injective. Notice that $\widehat{i^l}$ is not surjective: since $\text{ffs}(H) = F_n$, we have that the edges of $\text{core}(\varphi_l \circ ... \circ \varphi_1(H))$ make use of all the labels $\{a_1, ..., a_n\}$, while $\varphi_l \circ ... \circ \varphi_1(H \cap A)$ only uses labels from $S$. Let $0 \leq l' \leq l$ be the smallest index such that $\widehat{i^{l'}}$ isn't surjective. Define $G_1 = \text{im}(\widehat{i^{l'-1}}) = \text{core}(\varphi_{l'-1} \circ ... \circ \varphi_1(H))$, or $G_1 = \text{core}(H)$ if $\widehat{i^0}$ is not surjective. Since $\widehat{i^{l'}}$ is not surjective, it has to be injective: thus we can define $G_2 = \text{im}(\widehat{i^{l'}}) = \text{core}(\varphi_{l'} \circ ... \circ \varphi_1(H \cap A))$. Define $G_3 = \text{core}(\varphi_l \circ ... \circ \varphi_1(H \cap A))$.

For $k = 0, ..., l'-2$ we have $\|\text{im}(\widehat{i^{k+1}})\|_e \leq \|\text{im}(\widehat{i^k})\|_e$, giving a path in $\Lambda_r(E)$ from $\text{im}(\widehat{i^0}) = \text{core}(H)$ to $\text{im}(\widehat{i^{l'-1}}) = G_1$. This yields property (i) of the statement.

Since $\widehat{i^{l'}}$ is injective, we have that $\text{im}(\widehat{i^{l'}}) = G_2$ is a subgraph of $\text{core}(\varphi_{l'} \circ ... \circ \varphi_1(H)) = \text{core}(\varphi_{l'}(\pi_1(G_1, *))$ and satisfies $\|G_2\|_e = \|\text{im}(\widehat{i^{l'}})\|_e \leq \|\text{im}(\widehat{i^{l'-1}})\|_e \leq E$. This yields property (ii) of the statement.

For $k = l', ..., l-1$ we have $\|\text{core}(\varphi_{k+1} \circ ... \circ \varphi_1(H \cap A))\|_e \leq \|\varphi_k \circ ... \circ \varphi_1(H)\|_e$, giving a path in $\Lambda_{r-1}(E)$ from $G_2$ to $G_3$. This yields property (iii) of the statement.

We have an inclusion $\varphi_l \circ ... \circ \varphi_1(H \cap A) \leq \varphi_l \circ ... \circ \varphi_1(A)$. Since $\text{core}(\varphi_l \circ ... \circ \varphi_1(A))$ only uses labels from the proper subset $S \subset \{a_1, ..., a_n\}$, then so does $G_3 = \text{core}(\varphi_l \circ ... \circ \varphi_1(H \cap A))$. This yields property (iv) of the statement.

The subgroup $\varphi_l \circ ... \circ \varphi_1(H \cap A)$ is echelon since $H \cap A$ was by assumption. This yields property (v) of the statement.

($\Leftarrow$) Suppose we are given $G_1, G_2, G_3$ satisfying (i)-(v), and we want to prove that $H$ is echelon.

Let $B = \langle S \rangle$ be generated by the letters in condition (iv), so that $B$ is a proper free factor of $F_n$; take a subgroup $K_3 \leq F_n$ such that $\text{core}(K_3) = G_3$ and $K_3 \leq B$: this can be done thanks to condition (iv). Condition (v) tells us that $K_3$ is echelon. Take also $K_1, K_2 \leq F_n$ such that $\text{core}(K_1) = G_1$ and $\text{core}(K_2) = G_2$.

Condition (iii) tells us that $\mu(K_3) = K_2$ for some automorphism $\mu$ of $F_n$. To be precise, we take a path in $\Lambda_{r-1}(E)$ from $G_2$ to $G_3$, we look at the Whitehead automorphisms that label the edges of that path, and we take the composition of their inverses; we also compose with a suitably chosen conjugation automorphism. Condition (ii) tells us that $\nu(K_2)$ is a free factor in $K_1$ for some automorphism $\nu$ of $F_n$. The automorphism $\nu$ is obtained as the inverse of the automorphism $\varphi$ given by condition (ii), composed with a conjugation automorphism. Condition (i) implies that $\eta(K_1) = H$ for some automorphism $\eta$ of $F_n$.

Let $A = \eta \circ \nu \circ \mu(B)$ and notice that $A$ is a proper free factor of $\text{ffs}(H) = F_n$; in

particular $H \cap A$ is a proper free factor in $H$. Let $H' = \eta \circ \nu \circ \mu(K_3)$ and notice that $H' \leq A$ (because $K_3 \leq B$) and $H'$ is a proper free factor in $H$ (because $\nu(K_2)$ is a proper free factor in $K_1$) and $H'$ has rank $r-1$ (because $K_3$ has rank $r-1$, since $G_3$ is a vertex of $\Lambda_{r-1}(E)$). Now $H' \leq H \cap A$ are two proper free factors of $H$ and $H'$ has rank $r-1$: this forces $H' = H \cap A$.

Thus we have found a free factor $A \leq \mathrm{ffs}(H)$ such that $\mathrm{rank}(H \cap A) = r-1$ and $H \cap A$ is echelon. Lemma 3.4.11 gives us the desired conclusion. □

**Theorem 3.4.14.** *There is an algorithm that takes as input a finitely generated subgroup $H \leq F_n$ and tells us whether it is echelon or not. In case the answer is affirmative, it also computes a basis with respect to which $H$ is echelon.*

*Proof.* We proceed by induction on $r = \mathrm{rank}(H)$. For $r = 1$ we have that every subgroup is echelon with respect to any basis. Suppose now we have the algorithm for subgroups of rank $r-1$. Let $H \leq F_n$ be a finitely generated subgroup of rank $\mathrm{rank}(H) = r$. Without loss of generality we can assume $\mathrm{ffs}(H) = F_n$ (otherwise we compute a basis for $\mathrm{ffs}(H)$ using the algorithm of Theorem 3.4.4, and then we use $\mathrm{ffs}(H)$ as ambient group instead of $F_n$). Let $E = \|\mathrm{core}(H)\|_{\mathrm{e}}$.

We build the finite graphs $\Lambda_r(E)$ and $\Lambda_{r-1}(E)$; we take all graphs $G_1$ such that there is a path in $\Lambda_r(E)$ from $\mathrm{core}(H)$ to $G_1$; for each such $G_1$ and for each Whitehead automorphism $\varphi$, we take all the subgraphs $G_2$ of $\varphi(G_1)$ with at most $E$ edges; for each such $G_2$, we take all graphs $G_3$ such that there is a path in $\Lambda_{r-1}(E)$ from $G_2$ to $G_3$; finally, for each such $G_3$, we check whether it uses all the labels or not, and whether its fundamental group is echelon or not (and we are able to do this algorithmically, by inductive hypothesis, since $\mathrm{rank}(G_3) = r-1$). If no graph $G_3$ satisfies both the conditions, then by Proposition 3.4.13 we have that $H$ is not echelon, and the algorithm gives negative answer. If there is a graph $G_3$ that satisfies both the conditions, then by Proposition 3.4.13 we have that $H$ is echelon and the algorithm gives affirmative answer.

It remains, in case of an affirmative answer, to compute a basis with respect to which $H$ is echelon. We take the graph $G_3$ with echelon fundamental group, and we consider an echelon subgroup $K_3 \leq F_n$ with $\mathrm{core}(K_3) = G_3$. The inductive hypothesis on the subgroup $K_3$ allows us to compute a minimal flag $B_1 \leq ... \leq B_{r-1} \leq F_n$ for $K_3$.

We take the proper subset $S \subset \{a_1, ..., a_n\}$ of letters which appear as labels of at least one edge of $G_3$, and we call $B = \langle S \rangle$. We take finitely generated subgroups $K_1, K_2 \leq F_n$ with $\mathrm{core}(K_1) = G_1$ and $\mathrm{core}(K_2) = G_2$. As in the proof of Proposition 3.4.13, we have $K_3 \leq B$ and we are able (by looking at the paths in $\Lambda_r(E)$ and $\Lambda_{r-1}(E)$) to compute an automorphism $\rho$ of $F_n$ such that $\rho(K_3) \leq H$ is a proper free factor in $H$ of rank $r-1$. We finally proceed as in the proof of Lemma 3.4.11, and we consider the chain of free factors $\rho(B_1) \leq ... \leq \rho(B_{r-1}) \leq \mathrm{ffs}(H) = F_n$: this is a flag for $H$, as desired. □

### 3.4.4 About the intersection of echelon subgroups

In [Ros13] A. Rosenmann asked whether the intersection of two echelon subgroups is always echelon. We give negative answer to this question: the following proposition provides a counterexample.

**Proposition 3.4.15.** *Consider the free group $F_2 = \langle a, b\rangle$. Then we have the following:*

*(i) The subgroup $H = \langle a^2, b^2a^2b^2\rangle$ is echelon with respect to the ordered basis $a, b$.*

*(ii) The subgroup $H' = \langle b^2, a^2b^2a^2\rangle$ is echelon with respect to the ordered basis $b, a$.*

*(iii) The subgroup $H \cap H' = \langle a^2b^2a^2b^2, b^2a^2b^2a^2\rangle$ is not echelon.*

*Proof.* Parts (i) and (ii) are immediate. Using the algorithm of Proposition 1.2.21 we can see that $\mathrm{core}_*(H \cap H')$ is as in Figure 3.5, and thus $H \cap H' = \langle a^2b^2a^2b^2, b^2a^2b^2a^2\rangle$. To prove that $H \cap H'$ is not echelon, we might just use the algorithm of Theorem 3.4.14, but we prefer to provide here a shorted argument. Let $K = H \cap H'$.

Suppose by contradiction that $K$ is echelon with respect to the ordered basis $p, q$ of $F_2$. Then we must have $\mathrm{rank}(K \cap \langle p\rangle) = 1$ meaning that $K$ contains a power $p^\alpha$ of the primitive element $p$. The inclusion $i : \langle p^\alpha\rangle \to K$ gives a map of graphs $\widehat{i} : \mathrm{core}(\langle p^\alpha\rangle) \to \mathrm{core}(K)$ and the image $\mathrm{im}(\widehat{i})$ contains at least one of the left cycle *aabbaabb* or the right cycle *bbaabbaa* of Figure 3.5. In both cases, the Whitehead graph of $\mathrm{core}(\langle p^\alpha\rangle)$ contains a cycle involving all the four vertices $a, \overline{a}, b, \overline{b}$, and thus it can't contain a cut vertex (see Definitions 2.2.1 and 2.2.2). But the Whitehead graph of $\mathrm{core}(\langle p^\alpha\rangle)$ is the same as the Whitehead graph of $\mathrm{core}(\langle p\rangle)$, and by Theorem 3.2.5 this has to contain a cut vertex, contradiction. □

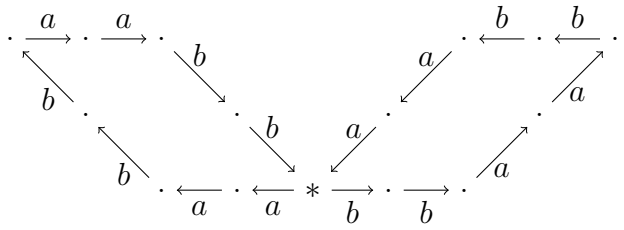


Figure 3.5: The graph $\mathrm{core}_*(K)$ where $K$ is generated by $a^2b^2a^2b^2$ and $b^2a^2b^2a^2$.

The next proposition shows, by means of a counterexample, that the property of being echelon isn't transitive: there are finitely generated subgroups $K \leq H \leq F_n$ such that $K$ is echelon in $H$ and $H$ is echelon in $F_n$, but $K$ isn't echelon in $F_n$.

**Proposition 3.4.16.** *Consider the free group $F_3 = \langle a, b, c\rangle$ and let $w = c^2a^2b^2a^2b^2c^2$. Then we have the following:*

*(i) The subgroup $H = \langle a, b, w\rangle$ is echelon in $F_3$.*

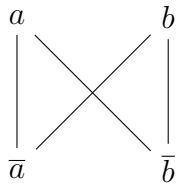


Figure 3.6: The Whitehead graph of the graph given by a single path labeled $a^2b^2a^2b^2$. This doesn't contain a cut vertex, and any Whitehead graph containing this can't contain a cut vertex.

*(ii) The subgroup $K = \langle w\overline{a}, awb\rangle$ is a free factor in $H$, and thus $K$ is echelon in $H$.*
*(iii) The subgroup $K$ isn't echelon in $F_3$.*

*Proof.* Part (i) is immediate since $H$ is echelon with respect to the ordered basis $a, b, c$. Part (ii) is immediate too since $w\overline{a}, awb$ is a basis for $K$ and $w\overline{a}, awb, w$ is a basis for $H$. To prove that $K$ is not echelon in $F_3$, we may just use the algorithm of Theorem 3.4.14, but we provide here a shorter proof.

Suppose by contradiction $K$ is echelon in $F_3$: then there is a free factor $J \leq F_3$ such that $\mathrm{rank}(K \cap J) = 1$. Let $\alpha : \mathrm{core}(K \cap J) \to \mathrm{core}(K)$ be the map induced by the inclusion: the graph $\mathrm{core}(K \cap J)$ is a subdivision of the unit circle, so the map $\alpha$ is essentially a loop inside $\mathrm{core}(K)$, see Figure 3.7. This loop is reduced (because $\mathrm{core}(K \cap J)$ is folded), so it's quite easy to see that the subdivided circle $\mathrm{core}(K \cap J)$ has to contain a path labeled with $w = c^2a^2b^2a^2b^2c^2$. We now compose this path with the map $\beta : \mathrm{core}(K \cap J) \to \mathrm{core}(J)$ induced by the inclusion, and we deduce that $\mathrm{core}(J)$ has to contains a path labeled with $w = c^2a^2b^2a^2b^2c^2$ too. In particular, the Whitehead graph of $\mathrm{core}(J)$ contains the graph of Figure 3.8 as a subgraph, and thus it can't contain a cut vertex. But from Theorem 3.2.5 it follows that $\mathrm{core}(J)$ has to be a one-vertex graph; since it contains a path labeled with $w = c^2a^2b^2a^2b^2c^2$, this implies that $\mathrm{core}(J)$ has to be the rose $R_3$ and thus $J = F_3$, contradiction. □

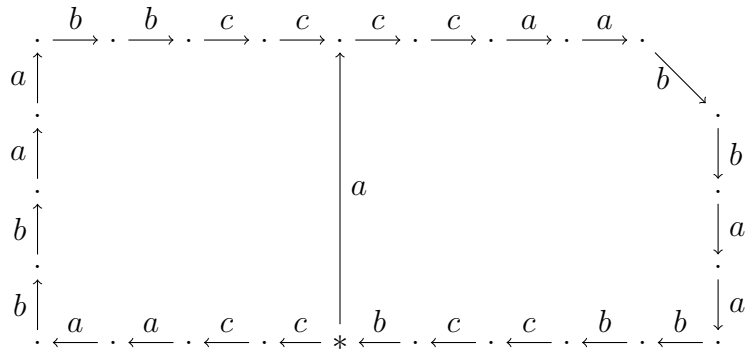


Figure 3.7: The graph $\mathrm{core}_*(K)$ where $K = \langle w\overline{a}, awb\rangle$ and $w = c^2a^2b^2a^2b^2c^2$.

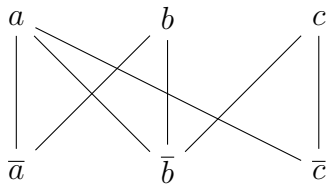


Figure 3.8: The Whitehead graph of the graph given by a single path labeled $c^2a^2b^2a^2b^2c^2$. This doesn't contain a cut vertex, and any Whitehead graph containing this can't contain a cut vertex.

# 4 | Ideals of equations for elements in a free group

Given an extension of fields $K \subseteq F$ and an element $\alpha \in F$, a first interesting question to ask is to determine whether the element $\alpha$ is algebraic over $K$, i.e. whether it satisfies some non-trivial equation with coefficients in $K$. In other words we want to determine whether there exists a non-trivial polynomial $p(x) \in K[x]$ such that $p(\alpha) = 0$. If the answer is affirmative, one tries to study the ideal $I_\alpha \subseteq K[x]$ of equations for $\alpha$ over $K$: this turns out to be a principal ideal, and thus its structure is very simple.

Completely analogous questions can be asked in the context of group theory, but the answers turn out to be much more complicated.

## 4.1 Equations for elements in a free group

Let $F_n$ be a free group generated by $n$ elements $a_1, ..., a_n$. Let $H \leq F_n$ be a finitely generated subgroup and consider an infinite cyclic group $\langle x \rangle \cong \mathbb{Z}$. An **equation** in $x$ with coefficients in $H$ is an equality of the form $w(x) = 1$ with $w(x) \in H * \langle x \rangle$ in the free product of $H$ and $\langle x \rangle$. With an abuse of notation, we will also call the element $w(x)$ "equation", instead of the equality $w(x) = 1$.

For an element $g \in F_n$, consider the map $\varphi_g : H * \langle x \rangle \to F_n$ that is the inclusion on $H$, and that sends $x$ to $g$; this is the "evaluation in $g$" map, since the equation $w(x) \in H * \langle x \rangle$ is sent to the element $w(g) \in F_n$. We say that $g$ is a **solution** for the equation $w$ if $\varphi_g(w) = 1$. We define the **ideal** $\mathfrak{I}_g$ to be the normal subgroup $\mathfrak{I}_g = \ker \varphi_g$ of $H * \langle x \rangle$.

### 4.1.1 Equations in a free group

In what follows, it will be of fundamental importance to consider equations as paths in a suitable labeled graph. Fix a finitely generated subgroup $H \leq F_n$ and an element $g \in F_n$. Consider the labeled graph $G = \mathrm{core}_*(H) \vee \mathrm{core}_*(\langle g \rangle)$ given by the disjoint union of $\mathrm{core}_*(H)$ and $\mathrm{core}_*(\langle g \rangle)$ where we identify the two basepoints to a single point. Let $f : G \to R_n$ be the labeling map, inducing a map $f_* : \pi_1(G, *) \to F_n$ of

fundamental groups. As in the proof of Theorems 1.3.3 and 1.3.4, we also consider the isomorphism $\theta : H * \langle x \rangle \to \pi_1(G, *)$ which sends each element of $H$ to the homotopy class of the corresponding reduced path in $\text{core}_*(H)$, and the element $x$ to the homotopy class of the path in $\text{core}_*(\langle g \rangle)$ corresponding to the element $g$; it is immediate to see that $f_* \circ \theta = \varphi_g$ as maps from $H * \langle x \rangle$ to $\pi_1(R_n, *) = F_n$.

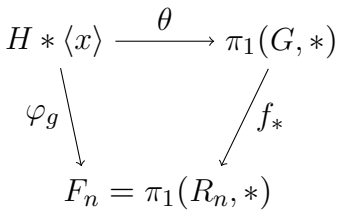


Figure 4.1: The diagram commutes.

**Definition 4.1.1.** *Let $w \in H * \langle x \rangle$ be a non-trivial equation. Define the **corresponding path** $\sigma : I_l \to G$ as the unique reduced path in the homotopy class $\theta(w)$ (see Proposition 1.1.3).*

**Definition 4.1.2.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$. Define the **corresponding equation** $w \in H * \langle x \rangle$ as $w = \theta^{-1}([\sigma])$.*

The two above definitions give a bijection between non-trivial equations $w \in H * \langle x \rangle$ and reduced paths $\sigma : I_l \to G$. Cyclically reduced paths correspond to **cyclically reduced equations**, i.e. equations $w \in H * \langle x \rangle$ such that, when we write $w$ as a reduced word in the letters $a_1, ..., a_n, x$, the word is also cyclically reduced. Equations $w \in \mathfrak{I}_g$ correspond to paths $\sigma$ such that $f \circ \sigma$ is homotopically trivial.

### 4.1.2 Elements that depend on a subgroup

We now introduce the notion of *dependence* on a subgroup, i.e. of being "algebraic" over the subgroup. Fixed a finitely generated subgroup $H \leq F_n$, we provide an effective characterization for all the elements $g$ that are "algebraic" over $H$, which is due to A. Rosenmann and E. Ventura, see [RV21].

**Definition 4.1.3.** *Let $H \leq F_n$ be a finitely generated subgroup and let $g \in F_n$ be any element. We say that $g$ **depends** on $H$ if the ideal $\mathfrak{I}_g$ is non-trivial.*

**Proposition 4.1.4.** *Let $H \leq F_n$ be a finitely generated subgroup and let $g \in F_n$. Let $G$ be the labeled graph as defined in Section 4.1.1. Then the following are equivalent:*

*(i) The element $g$ depends on $H$.*

*(ii)* $\text{rank}(\langle H, g \rangle) \leq \text{rank}(H)$.

*(iii) Along a maximal chain of folding operations for $G$, at least one non-rank-preserving folding operation takes place.*

*Proof.* We start with the graph $G$ defined in the previous Section 4.1.1, and we apply a maximal sequence of folding operations. We observe that the fundamental group of the initial graph $G$ has rank $\text{rank}(H)+1$, while the fundamental group of the final graph $\text{core}_*(\langle H, g\rangle)$ has rank $\text{rank}(\langle H, g\rangle)$.

We have that the ideal $\mathfrak{I}_g$ is non-trivial if and only if the sequence contains at least one non-rank-preserving folding operation, if and only if the rank of the final graph is strictly smaller than the rank of the initial graph, i.e. $\text{rank}(\langle H, g\rangle) \leq \text{rank}(H)$. □

*Remark.* Notice that condition (iii) of Proposition 4.1.4 doesn't depend on the chosen maximal folding sequence, see Theorem 1.3.2.

The following theorem provides a characterization, given a finitely generated subgroup $H \leq F_n$, of all elements $g$ that depend on $H$.

**Theorem 4.1.5** (A. Rosenmann, E. Ventura, [RV21])**.** *Let $H \leq F_n$ be a finitely generated subgroup. Then there is an algorithm that computes a finite set of elements $g_1, ..., g_k \in F_n$ such that, for every $g \in F_n$, the following are equivalent:*

*(i) The element $g$ depends on $H$.*

*(ii) The element $g$ belongs to one of the double cosets $Hg_1H, ..., Hg_kH$.*

*Proof.* Consider the graph $C = \text{core}_*(H)$. Given two distinct vertices $v, w$ of $C$, we call $C_{v=w}$ the labeled graph obtained identifying $v$ and $w$ to a single point. We say that $C_{v=w}$ is *interesting* if a maximal sequence of folding operations for $C_{v=w}$ contains at least one non-rank-preserving folding operation.

Let $g \in F_n$ be any element. Consider the graph $G = C \vee \text{core}_*(\langle g\rangle)$ as defined in the previous Section 4.1.1: we have that $g$ depends on $H$ if and only if, when performing a maximal sequence of folding operations on $G$, the sequence contains at least one non-rank-preserving folding operation.

CASE 1: Suppose that $g$ can be written as $g = ab$ for some reduced words $a, b$ with the following property: $\text{core}_*(H)$ contains paths $\sigma_a$ from $*$ to $v$ and $\sigma_b$ from $w$ to $*$ such that the words that we read along $\sigma_a, \sigma_b$ are $a, b$ respectively. Of course $\text{core}_*(\langle g\rangle)$ contains paths $\tau_a$ and $\tau_b$ starting at $*$ and such that along $\tau_a, \tau_b$ we read the words $a, b$ respectively. We can thus take the graph $G$ and perform folding operations that identify $\tau_a$ with $\sigma_a$ and $\tau_b$ with $\sigma_b$: we observe that these folding operations are all rank-preserving, and that the resulting graph is exactly $C_{v=w}$. Thus we have that $g$ depends on $H$ if and only if $C_{v=w}$ is interesting.

CASE 2: Suppose conversely that $g$ can't be written in that form. Then we write $g = ag'b$ for some reduced words $a, g', b$ with $g'$ non-empty and with the following property: $\text{core}_*(H)$ contains paths $\sigma_a$ from $*$ to $v$ and $\sigma_b$ from $w$ to $*$ such that the words that we read along $\sigma_a, \sigma_b$ are $a, b$ respectively, and $a, b$ are the words of maximum length with this property. Again, $\text{core}_*(\langle g\rangle)$ contains paths $\tau_a$ and $\tau_b$ starting at $*$ and such that along $\tau_a, \tau_b$ we read the words $a, b$ respectively. We can thus take the graph $G$ and perform folding operations that identify $\tau_a$ with $\sigma_a$ and

$\tau_b$ with $\sigma_b$: we observe that these folding operations are rank-preserving, and that the resulting graph is the same as $\text{core}_*(H)$ but with an edge-path going from $v$ to $w$ and labeled with $g'$. At this point, no more folding operation is possible (because $a$ and $b$ where of maximal length), and thus in this case $g$ doesn't depend on $H$.

For each interesting graph $C_{v=w}$, we choose a combinatorial path in $C$ going from $*$ to $v$, and let $a$ be the word that you read along that path; similarly we choose a combinatorial path in $C$ going from $w$ to $*$, and let $b$ be the word that you read along that path; we define the element $g_{v=w} = ab$ of $F_n$. Observe that a different choice of paths from $*$ to $v$ and from $w$ to $*$ would have led to another element in the same double coset $Hg_{v=w}H$.

From the analysis of cases 1 and 2 above, it easily follows that $g$ depends on $H$ if and only if $g$ falls into one of the double cosets $Hg_{v=w}H$ for some $C_{v=w}$ interesting, or $g$ belongs to the trivial double coset $H$. The conclusion follows. □

## 4.2 The minimum possible degree in an ideal

Let $F_n$ be a free group with basis $a_1, ..., a_n$ and let $H \leq F_n$ be a finitely generated subgroup. For an equation $w \in H * \langle x \rangle$ we have that $w$ has a unique expression as a reduced word in the alphabet $\{x, \overline{x}\} \cup H \setminus \{1\}$.

**Definition 4.2.1.** *Define the **degree** of $w$ as the number of occurrences of $x$ and $\overline{x}$ in the cyclic reduction of $w$.*

If we take into account that $H$ is a subgroup of $F_n$, we can write $w$ as a reduced word in the letters $a_1, ..., a_n, x, \overline{a}_1, ..., \overline{a}_n, \overline{x}$. The degree of $w$ can be computed by looking at the number of occurrences of $x$ and $\overline{x}$ in the cyclic reduction of $w$ as a word in this alphabet as well.

Our aim in this section is, given $H \leq F_n$ finitely generated and $g \in F_n$, to obtain information about equations in $\mathfrak{I}_g$ in relation to their degree. Let $G$ be the graph defined in Section 4.1.1. We will prove the following theorem:

**Theorem 4.2.2** ([Asc22b])**.** *Let $L$ be the number of edges of the graph $G$ and let $d_{\min}$ be the minimum possible degree for a non-trivial equation in $\mathfrak{I}_g$. Then there is a non-trivial equation $w \in \mathfrak{I}_g$ of degree $d_{\min}$ and such that the corresponding path $\sigma : I_l \to G$ has length $l \leq 16L^2 d_{\min}$.*

As a corollary, this provides an algorithm to compute the minimum possible degree $d_{\min}$ for a non-trivial equation in $\mathfrak{I}_g$.

### 4.2.1 Reduction of paths

Before going deeper into the study of the ideal of the equations $\mathfrak{I}_g$, we need to introduce some notation about reduction of combinatorial paths in a graph. Consider

the graph $I_l$ given by the subdivision of the unit interval $[0,1]$ into $l$ edges, as defined in Section 1.1.1, and for an edge $s$ of $I_l$ denote with $o^-(s), o^+(s)$ the vertices of $s$. We adopt the convention that $o^-(s) < o^+(s)$ are points of the unit interval.

Let $G$ be a graph and $\sigma : I_l \to G$ be a combinatorial path. If $\sigma$ is not reduced, then we can find two consecutive edges $s, t$ of $I_l$ such that $\sigma$ sends $s, t$ to the same edge $e$ of $G$, but crossed with opposite orientations. Let's say we have $o^+(s) = o^-(t)$: we consider the interval $s \cup t = [o^-(s), o^+(t)]$ and we collapse it to a point. We obtain a graph isomorphic to $I_{l-2}$, and we can define a map $\sigma' : I_{l-2} \to G$ which is equal to $\sigma$, except on the collapsed interval, where we set it to be equal to $\sigma(o^-(s)) = \sigma(o^+(t))$. The map $\sigma' : I_{l-2} \to G$ is a combinatorial path, and it is homotopic to $\sigma$ (relative to the endpoints). If the path $\sigma'$ is not yet reduced, then we can reiterate the same process. This motivates the following definition:

**Definition 4.2.3.** *Let $G$ be a graph and let $\sigma : I_l \to G$ be a combinatorial path. A **reduction process** for $\sigma$ is a sequence $(s_1, t_1), ..., (s_m, t_m)$ with the following properties:*

*(i) $s_1, t_1, ..., s_m, t_m$ are pairwise distinct edges of $I_l$.*

*(ii) For every $k = 1, ..., m$ we have $o^-(s_k) < o^-(t_k)$.*

*(iii) For every $k = 1, ..., m$, if we collapse each of $s_1, t_1, ..., s_{k-1}, t_{k-1}$ to a point, in the quotient graph the edges $s_k, t_k$ are adjacent.*

*(iv) For every $k = 1, ..., m$ the map $\sigma$ sends $s_k, t_k$ to the same edge of $G$ crossed with opposite orientations.*

Think of $(s_k, t_k)$ as the $k$-th cancellation to be performed on the path $\sigma$. Condition (iii) says that, after performing the first $k-1$ cancellations, the edges $s_k, t_k$ are adjacent, ready to be canceled against each other. Condition (iv) ensures that $\sigma$ sends $s_k, t_k$ to the same edge of $G$ but with opposite orientations, so that the cancellation actually makes sense. Condition (ii) is just a useful convention, saying that the edges $s_k, t_k$ appear in this order on the unit interval $I_l$.

**Lemma 4.2.4.** *Let $G$ be a graph and let $\sigma : I_l \to G$ be a combinatorial path, together with a reduction process $(s_1, t_1), ..., (s_m, t_m)$. Then for every $1 \leq \alpha < \beta \leq m$ the edges $s_\alpha, t_\alpha, s_\beta, t_\beta$ appear on the interval in one of these orders: $s_\alpha, t_\alpha, s_\beta, t_\beta$ or $s_\beta, t_\beta, s_\alpha, t_\alpha$ or $s_\beta, s_\alpha, t_\alpha, t_\beta$.*

*Proof.* When we collapse $s_1, t_1, ..., s_{\alpha-1}, t_{\alpha-1}$ to a point, we have that $s_\alpha$ and $t_\alpha$ become adjacent. This means that none of $s_\beta, t_\beta$ can occur between $s_\alpha$ and $t_\alpha$. The conclusion follows. □

Let $\sigma : I_l \to G$ be a path and let $(s_1, t_1), ..., (s_m, t_m)$ be a reduction process for $\sigma$. If $2m < l$, then we can collapse each of the edges $s_1, t_1, ..., s_m, t_m$ to a point in order to get a graph isomorphic to $I_{l-2m}$. We can define a continuous map $\sigma' : I_{l-2m} \to G$ which is equal to $\sigma$ on the edges which are not collapsed in the process. The map $\sigma'$

is a combinatorial path which is homotopic to $\sigma$ (relative to the endpoints), and it is called **residual path** of the cancellation process.

**Proposition 4.2.5.** *Let $\sigma : I_l \to G$ be a combinatorial path and let $(s_1, t_1), ..., (s_m, t_m)$ be a reduction process for $\sigma$. Then exactly one of the following holds:*
*(i) We have $2m = l$ and $\sigma(0) = \sigma(1)$ and $\sigma$ is nullhomotopic (relative to the endpoints).*
*(ii) We have $2m < l$ and the residual path $\sigma' : I_{l-2m} \to G$ is reduced.*
*(iii) There is a couple $(s_{m+1}, t_{m+1})$ such that $(s_1, t_1), ..., (s_m, t_m), (s_{m+1}, t_{m+1})$ is a reduction process for $\sigma$.*

*Proof.* Suppose that $2m < l$ and that the residual path $\sigma' : I_{l-2m} \to G$ is not reduced. Then there are two adjacent edges $s', t'$ in $I_{l-2m}$ such that $\sigma'$ sends $s', t'$ to the same edge of $G$, crossed with opposite orientation; let's also assume $o^-(s') < o^-(t')$. The domain $I_{l-2m}$ of $\sigma'$ is a quotient of the domain $I_l$ of $\sigma$; thus we find unique edges $s_{m+1}, t_{m+1}$ of $I_l$ such that the quotient sends $s_{m+1}, t_{m+1}$ to $s', t'$ respectively; notice that $o^-(s_{m+1}) < o^-(t_{m+1})$ and that $s_{m+1}, t_{m+1}$ are distinct, and they are also distinct from $s_1, t_1, ..., s_m, t_m$. From the definition of $\sigma'$, it is immediate to see that $\sigma$ sends $s_{m+1}, t_{m+1}$ to the same edge of $G$, crossed with opposite orientation. It follows that $(s_1, t_1), ..., (s_m, t_m), (s_{m+1}, t_{m+1})$ is a reduction process for $\sigma$, as desired. □

The above proposition essentially says that a reduction process can be inductively extended, until we get a path which is either trivial or reduced. A reduction process $(s_1, t_1), ..., (s_m, t_m)$ is called **maximal** if it can't be extended by adding a couple of edges $(s_{m+1}, t_{m+1})$, i.e. if it falls into case (i) or (ii) of Proposition 4.2.5. Of course every path admits at least one maximal reduction process. Despite the maximal reduction process not being unique in general, the residual path is unique, and it coincides with the reduction of $\sigma$:

**Proposition 4.2.6.** *Let $\sigma : I_l \to G$ be a path which is not homotopically trivial (relative to the endpoints) and let $\overline{\sigma} : I_r \to G$ be its reduction. Then for every maximal reduction process $(s_1, t_1), ..., (s_m, t_m)$ for $\sigma$, we have $l - 2m = r$ and the residual path coincides with $\overline{\sigma}$.*

*Proof.* The residual path of a maximal reduction process is homotopic to $\sigma$ (relative to the endpoints) and reduced. The conclusion follows from Proposition 1.1.3. □

The following graphical representation of a reduction process will be useful. Let $\sigma : I_l \to G$ be a combinatorial path and let $(s_1, t_1), ..., (s_m, t_m)$ be a reduction process for $\sigma$. Consider $I_l$ as a subdivision of the unit interval $[0, 1] \times \{0\} \subseteq \mathbb{R}^2$. For each couple $(s_i, t_i)$, take a smooth path $r_i$ in the upper half-plane connecting the midpoint of $s_i$ to the midpoint of $t_i$. The paths $r_1, ..., r_m$ can be taken to be pairwise disjoint, as in Figure 4.2.

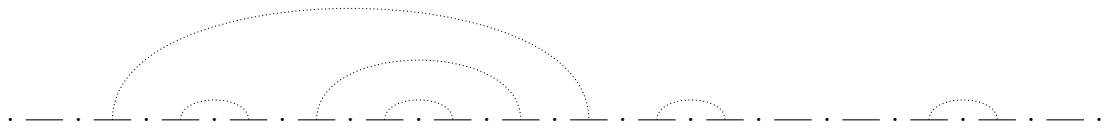

Figure 4.2: An example of a possible diagram for a reduction process.

### 4.2.2 The degree of an equation and innermost cancellations

We pointed out that, for our exposition, it is fundamental to look at equations for $g$ over $H$ as paths in a suitable graph $G$, as defined in Section 4.1.1. The degree of a cyclically reduced equation can be computed by looking at how many times the corresponding path crosses the edges of $\text{core}(\langle g \rangle)$.

**Lemma 4.2.7.** *Let $\sigma : I_l \to G$ be a cyclically reduced path. Let $e$ be any edge of $G$ that belongs to the subgraph $\text{core}(\langle g \rangle)$. Then the degree of the equation $w$ corresponding to $\sigma$ coincides with the number of times $\sigma$ crosses the edge $e$ (in either direction).*

*Proof.* Write the equation $w$ as a cyclically reduced word $c_1 x^{\alpha_1} c_2 x^{\alpha_2} ... c_r x^{\alpha_r} c_{r+1}$ with $\alpha_1, ..., \alpha_r \in \mathbb{Z} \setminus \{0\}$ and $c_1, ..., c_{r+1} \in H$. Then in the graph $G$ we have that $\theta(w) = \theta(c_1) \cdot \theta(x^{\alpha_1}) \cdot ... \cdot \theta(x^{\alpha_r}) \cdot \theta(c_{r+1})$, where the $\cdot$ symbol denotes the concatenation of paths (without any homotopy). It is immediate to see that $\theta(x^{\alpha_i})$ crosses each edge of $\text{core}(\langle g \rangle)$ exactly $|\alpha_i|$ times, for $i = 1, ..., r$, and that $\theta(c_i)$ is contained in $\text{core}_*(H)$ and thus it doesn't cross any edge of $\text{core}(\langle g \rangle)$. The conclusion follows. □

Non-trivial equations $w \in \mathfrak{I}_g$ with $g$ as a solution correspond to reduced paths $\sigma : I_l \to G$ such that $f \circ \sigma$ is homotopycally trivial (relative to the endpoints). In this case we can take a maximal reduction process $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ for $f \circ \sigma$. The following two lemmas, which will be of fundamental importance in what follows, tell us that the degree of an equation is closely related to the number of innermost cancellations.

**Definition 4.2.8.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopycally trivial (relative to the endpoints); let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. A couple $(s_i, t_i)$ is called **innermost cancellation** if $s_i$ and $t_i$ are adjacent on the interval $I_l$.*

**Lemma 4.2.9.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial (relative to its endpoints); let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Let $(s_i, t_i)$ be an innermost cancellation. Then, among $\sigma(s_i)$ and $\sigma(t_i)$, one is an edge of $\text{core}_*(H)$ and the other is an edge of $\text{core}_*(\langle g \rangle)$ (and the vertex between them is the basepoint).*

*Proof.* Since $(s_i, t_i)$ is a couple of a reduction process for $f \circ \sigma$, we have that $(f \circ \sigma)(s_i)$ and $(f \circ \sigma)(t_i)$ are the same edge of fold$(G)$ but with opposite orientations. In particular $(f \circ \sigma)(s_i)$ and $(f \circ \sigma)(t_i)$ have the same label and opposite orientations, and, since $p$ is label-preserving, the two edges $\sigma(s_i)$ and $\sigma(t_i)$ have the same label and opposite orientations too. Observe that $\sigma(s_i)$ and $\sigma(t_i)$ are adjacent but distinct, since $\sigma$ is a reduced path. This means that $\sigma(s_i)$ and $\sigma(t_i)$ can't both belong to $\text{core}_*(H)$ (because it is folded), and can't both belong to $\text{core}_*(\langle g \rangle)$ (because it is folded too). Thus one of them has to belong to $\text{core}_*(H)$ and the other to $\text{core}_*(\langle g \rangle)$, and the conclusion follows. □

**Lemma 4.2.10.** *Let $\sigma : I_l \to G$ be a cyclically reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial (relative to its endpoints); let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal cancellation process for $f \circ \sigma$. Suppose the equation $w \in \mathfrak{I}_g$ corresponding to $\sigma$ has degree $d$. Then the reduction process contains at most $2d$ innermost cancellations.*

*Proof.* As in the proof of Lemma 4.2.7, write the equation $w$ as a cyclically reduced word

$$c_1 x^{\alpha_1} c_2 x^{\alpha_2} ... c_r x^{\alpha_r} c_{r+1}$$

with $\alpha_1, ..., \alpha_r \in \mathbb{Z} \setminus \{0\}$ and $c_1, ..., c_{r+1} \in H$. In the graph $G$ we have $\sigma = \theta(w) = \theta(c_1) \cdot \theta(x^{\alpha_1}) \cdot ... \cdot \theta(x^{\alpha_r}) \cdot \theta(c_{r+1})$, where the $\cdot$ symbol denotes the concatenation of paths (without any homotopy). We see that $w$ has degree $d = |\alpha_1| + ... + |\alpha_r| \geq r$ and, using Lemma 4.2.9, that the path $\sigma$ contains at most $2r$ innermost cancellations. The conclusion follows. □

### 4.2.3 Parallel cancellation

In this section we introduce the parallel cancellation moves, which allow us to produce a shorter equation from a longer one. We give a characterization of which parallel cancellation moves preserve the degree of the equation. Recall that $I_l$ is the unit interval $[0, 1]$ subdivided into $l$ segments, and recall that for an edge $s$ of $I_l$, we denote with $o^-(s), o^+(s)$ the endpoints of $s$, ordered on the interval in such a way that $o^-(s) < o^+(s)$.

**Definition 4.2.11.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. We say that two couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ with $\alpha < \beta$ are **parallel** if they satisfy the following conditions:*

*(i) The edges $s_\beta, s_\alpha, t_\alpha, t_\beta$ appear in this order on $I_l$.*

*(ii) The map $\sigma$ sends $s_\alpha, s_\beta$ to the same edge of $G$ crossed with the same orientation.*

*(iii) The map $\sigma$ sends $t_\alpha, t_\beta$ to the same edge of $G$ crossed with the same orientation.*

The reason behind the definition of parallel couples is that they allow us to perform a cancellation move, which we now describe, that will be of fundamental importance in the proof of the main theorem. Let $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ be two parallel couples; we take the subgraph of $I_l$ given by the interval $[o^-(s_\beta), o^-(s_\alpha)]$ and collapse it to a point; we also take the subgraph of $I_l$ given by the interval $[o^+(t_\alpha), o^+(t_\beta)]$ and collapse it to a point (see Figure 4.3). We obtain a graph isomorphic to $I_{l'}$, and notice that $2 \leq l' \leq l-2$ (because there are at least two edges that get collapsed, namely $s_\beta$ and $t_\beta$, and two that don't get collapsed, namely $s_\alpha$ and $t_\alpha$). We can define a map $\sigma' : I_{l'} \to G$ which is equal to $\sigma$, except on the collapsed interval $[o^-(s_\beta), o^-(s_\alpha)]$, where we set it equal to $\sigma(o^-(s_\beta)) = \sigma(o^-(s_\alpha))$, and except on the collapsed interval $[o^+(t_\alpha), o^+(t_\beta)]$, where we set it equal to $\sigma(o^+(t_\alpha)) = \sigma(o^+(t_\beta))$. This gives a well-defined combinatorial path $\sigma' : I_{l'} \to G$.

**Lemma 4.2.12** (Parallel cancellation)**.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Suppose for some $1 \leq \alpha < \beta \leq l/2$ the couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ are parallel and define the map $\sigma' : I_{l'} \to G$ as above. Then $\sigma'$ is a reduced path with $\sigma'(0) = \sigma'(1) = *$ and $f \circ \sigma'$ is homotopically trivial. A maximal reduction process for $f \circ \sigma'$ can be obtained from $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ by removing the couples containing edges which get collapsed in the definition of $I_{l'}$.*

*Proof.* Consider the map $\sigma' : I_{l'} \to G$ and we want to show that it is reduced. At the vertices in the interval $[0, o^-(s_\beta))$, the local injectivity of $\sigma'$ immediately follows from the local injectivity of $\sigma$; the same holds for the vertices in the intervals $[o^+(s_\alpha), o^-(t_\alpha)]$ and $(o^+(t_\beta), 1]$. For the vertex of $I_{l'}$ corresponding to the collapsed interval $[o^-(s_\beta), o^-(s_\alpha)]$, the local injectivity of $\sigma'$ follows from the local injectivity of $\sigma$ at $o^-(s_\beta)$ (and here we use the hypothesis that $\sigma$ sends $s_\beta$ and $s_\alpha$ to the same edge of $G$, crossed with the same orientation). Similarly, for the vertex of $I_{l'}$ corresponding to the collapsed interval $[o^+(t_\alpha), o^+(t_\beta)]$, the local injectivity of $\sigma'$ follows from the local injectivity of $\sigma$ at $o^+(t_\beta)$. This shows that $\sigma'$ is reduced.

It is easy to see, using Lemma 4.2.4, that for every $1 \leq i \leq m$ we have that either both or none of $s_i, t_i$ is collapsed to a point. Consider the sequence $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ and we remove the couple of edges that get collapsed (preserving the order of the other couples): the remaining couples contain the edges of $I_{l'}$, each appearing exactly once. We thus get a sequence $(q_1, r_1), ..., (q_{l'/2}, r_{l'/2})$ of couples of edges of $I_{l'}$. We want to prove that this is a reduction process for $\sigma'$ (the thesis then immediately follows).

We take a couple $(q_i, r_i)$ for some $1 \leq i \leq l'/2$ and we want to prove that, if in $I_{l'}$ we collapse each of $q_1, r_1, ..., q_{i-1}, r_{i-1}$ to a point, the edges $q_i, r_i$ become adjacent. We have $(q_i, r_i) = (s_j, t_j)$ for some $1 \leq j \leq l/2$. But In the sequence $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ we have that, collapsing each of $(s_1, t_1), ..., (s_{j-1}, t_{j-1})$, the edges $s_j, t_j$ become adjacent; of the couples $(s_1, t_1), ..., (s_{j-1}, t_{j-1})$, some get collapsed when passing from $I_l$ to $I_{l'}$, and the others are exactly the couples

$(q_1, r_1), ..., (q_{i-1}, r_{i-1})$; thus, if we collapse $(q_1, r_1), ..., (q_{i-1}, r_{i-1})$ too, the two edges $q_i$ and $r_i$ become adjacent, as desired.

Finally, take a couple $(q_i, r_i)$ for some $1 \leq i \leq l'/2$, and we want to prove that $f \circ \sigma'$ sends $q_i$ and $r_i$ to the same edge of fold$(G)$ crossed with opposite orientation. But $(q_i, r_i) = (s_j, t_j)$ for some $1 \leq j \leq l/2$, and $f \circ \sigma$ sends $s_j$ and $t_j$ to the same edge of fold$(G)$ crossed with opposite orientation. Since $\sigma'$ is defined to coincide with $\sigma$ on the edges $q_i = s_j$ and $r_i = t_j$, we have that $f \circ \sigma'$ sends $q_i$ and $r_i$ to the same edge of fold$(G)$ crossed with opposite orientation.

Thus $(q_1, r_1), ..., (q_{l'/2}, r_{l'/2})$ is a maximal reduction process for $f \circ \sigma'$, as desired. □

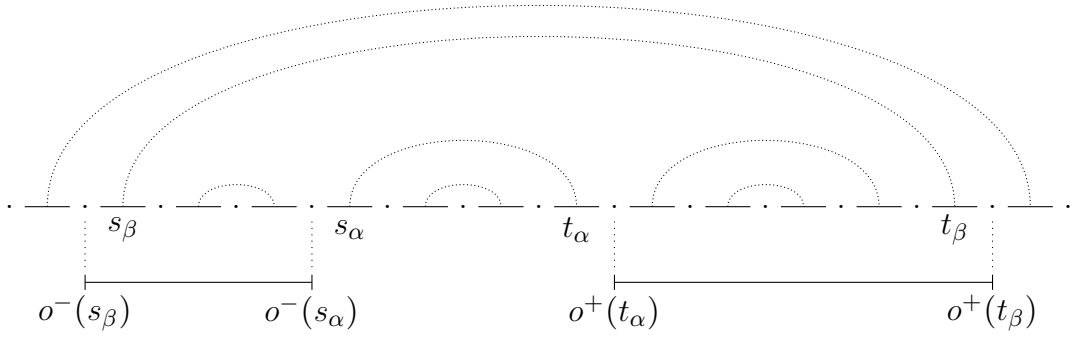


Figure 4.3: An example of a diagram for a maximal reduction process. The cancellation move collapses two intervals, which are painted below the interval, i.e. $[o^-(s_\beta), o^-(s_\alpha)]$ and $[o^+(t_\alpha), o^+(t_\beta)]$.

The following two lemmas give us information about how the degree of an equation changes when we perform a parallel cancellation move on the corresponding path.

**Lemma 4.2.13.** *Let $\sigma : I_l \to G$ and $\sigma' : I_{l'} \to G$ be cyclically reduced paths with $\sigma(0) = \sigma(1) = \sigma'(0) = \sigma'(1) = *$ and suppose $\sigma'$ is obtained from $\sigma$ by means of a cancellation move as described in Lemma 4.2.12. Then the degrees $d, d'$ of the corresponding equations $w, w'$ satisfy $d' \leq d$.*

*Proof.* Fix an edge $e$ of $G$ belonging to core$(\langle g \rangle)$ and apply Lemma 4.2.7: the domain of $\sigma'$ is the domain of $\sigma$ with some edges collapsed, and thus the number of times $\sigma'$ crosses the edge $e$ is lesser or equal than the number of times $\sigma$ does. □

**Definition 4.2.14.** *Let $\sigma : I_l \to G$ and $\sigma' : I_{l'} \to G$ be reduced paths with $\sigma(0) = \sigma(1) = \sigma'(0) = \sigma'(1) = *$ and suppose $\sigma'$ is obtained from $\sigma$ by means of a cancellation move as described in Lemma 4.2.12. We say that the parallel cancellation move is **degree-preserving** if the two equations $w, w'$ corresponding to the paths $\sigma, \sigma'$ have the same degree $d = d'$.*

**Lemma 4.2.15.** *Let $\sigma : I_l \to G$ be a cyclically reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Suppose there are two parallel couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ with $1 \leq \alpha < \beta \leq l/2$, and let $\sigma' : I_{l'} \to G$ be the reduced path obtained with the cancellation move described in Lemma 4.2.12. If the cancellation move is degree-preserving, then the images by $\sigma$ of the two intervals $[o^-(s_\beta), o^-(s_\alpha)]$ and $[o^+(t_\alpha), o^+(t_\beta)]$ are contained in* $\mathrm{core}(H)$*; moreover, the two paths $f \circ \sigma\big|_{[o^-(s_\beta), o^-(s_\alpha)]}$ and $f \circ \sigma\big|_{[o^+(t_\alpha), o^+(t_\beta)]}$ are the same, but walked in reverse direction.*

*Proof.* By hypothesis, $\sigma(s_\beta)$ and $\sigma(s_\alpha)$ are the same edge $e$ of $G$ crossed with the same orientation. Suppose first that $e$ doesn't belong to $\mathrm{core}(H)$. Then any reduced path that starts and ends with $e$ has to cross all the edges of $\mathrm{core}(\langle g \rangle)$. Thus, by Lemma 4.2.7, when we collapse the interval $[o^-(s_\beta), o^-(s_\alpha)]$ the degree strictly decreases.

Suppose now that $e$ belongs to $\mathrm{core}(H)$. Suppose there is an edge $s$ in the interval $[o^-(s_\beta), o^-(s_\alpha)]$ such that $\sigma(s)$ doesn't belong to $\mathrm{core}(H)$. If $\sigma(s)$ belongs to $\mathrm{core}(\langle g \rangle)$, then, by Lemma 4.2.7, when we collapse the interval $[o^-(s_\beta), o^-(s_\alpha)]$ the degree strictly decreases. If $\sigma(s)$ doesn't belong to $\mathrm{core}(H)$ nor to $\mathrm{core}(\langle g \rangle)$, then at least one of the paths $\sigma\big|_{[o^-(s_\beta), o^-(s)]}$ and $\sigma\big|_{[o^+(s), o^-(s_\alpha)]}$ crosses all the edges of $\mathrm{core}_*(\langle g \rangle)$; in particular, by Lemma 4.2.7, when collapsing the interval $[o^-(s_\beta), o^-(s_\alpha)]$ the degree strictly decreases. Thus the only possibility is that the path $\sigma\big|_{[o^-(s_\beta), o^-(s_\alpha)]}$ is contained in $\mathrm{core}(H)$. Similarly, we obtain that the path $\sigma\big|_{[o^+(t_\alpha), o^+(t_\beta)]}$ is contained in $\mathrm{core}(H)$ too. This proves the first part of the lemma.

For the second part, suppose the interval $[o^-(s_\beta), o^-(s_\alpha)]$ contains the two edges $s_i, t_i$ for some couple $(s_i, t_i)$ of our reduction process. Then the interval $[o^-(s_\beta), o^-(s_\alpha)]$ has to contain an innermost couple $(s_j, t_j)$ of our reduction process, i.e. a couple with $s_j, t_j$ adjacent on $I_l$. But by Lemma 4.2.9, at least one of the edges $\sigma(s_j), \sigma(t_j)$ has to belong to $\mathrm{core}_*(\langle g \rangle)$, which is a contradiction with our assumptions. Thus the interval $[o^-(s_\beta), o^-(s_\alpha)]$ does not contain both $s_i, t_i$ for any couple $(s_i, t_i)$ of our reduction process. The same holds for the interval $[o^+(t_\alpha), o^+(t_\beta)]$.

Now, using Lemma 4.2.4, it is easy to see that the reduction process has to pair up the edges of the interval $[o^-(s_\beta), o^-(s_\alpha)]$ with the edges of the interval $[o^+(t_\alpha), o^+(t_\beta)]$, and the pairing has to be done in decreasing order. It follows that $f \circ \sigma\big|_{[o^-(s_\beta), o^-(s_\alpha)]}$ and $f \circ \sigma\big|_{[o^+(t_\alpha), o^+(t_\beta)]}$ are the same path, walked in reverse directions. □

### 4.2.4 The minimum possible degree for a non-trivial equation

Let $L$ be the number of edges of the graph $G$.

**Proposition 4.2.16.** *Let $w \in \mathfrak{I}_g$ be a cyclically reduced equation of degree $d$ and let $\sigma : I_l \to G$ be the corresponding path; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be any maximal reduction process for $f \circ \sigma$. Suppose $l > 16L^2 d$. Then the reduction process contains two parallel couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$.*

*Proof.* To each couple $(s_i, t_i)$ we associate the quadruple of edges $(\sigma(s_i), \epsilon, \sigma(t_i), \delta)$ where $\sigma(s_i), \sigma(t_i)$ are edges of $G$ and $\epsilon, \delta \in \{+1, -1\}$ tell us the orientation with which $\sigma(s_i)$ and $\sigma(t_i)$ cross their images. There are $l/2$ couples that are sent to $4L^2$ possible quadruples; but by hypothesis we have $l/2 > 4L^2 \cdot 2d$, so we can find at least $2d + 1$ couples $(s_{i_1}, t_{i_1}), ..., (s_{i_{2d+1}}, t_{i_{2d+1}})$ with $i_1 < ... < i_{2d+1}$ which are sent to the same quadruple; this means that $\sigma$ sends $s_{i_1}, ..., s_{i_{2d+1}}$ all to the same edge of $G$ crossed with the same orientation, and $t_{i_1}, ..., t_{i_{2d+1}}$ all to the same edge of $G$ crossed with the same orientation.

Each of the couples $(s_{i_k}, t_{i_k})$ has to contain an innermost cancellation, i.e. there is a cancellation $(q_k, r_k)$ with $o^+(s_{i_k}) \leq o^+(q_k) = o^-(r_k) \leq o^-(t_{i_k})$. By Lemma 4.2.10 there are at most $2d$ innermost cancellations: since we have $2d + 1$ couples $(s_{i_1}, t_{i_1}), ..., (s_{i_{2d+1}}, t_{i_{2d+1}})$, two of them, let's say $(s_{i_j}, t_{i_j})$ and $(s_{i_k}, t_{i_k})$ with $j < k$, have to contain the same innermost cancellation $(q_j, r_j) = (q_k, r_k)$. But this forces $s_{i_j}, s_{i_k}, t_{i_k}, t_{i_j}$ to appear in this order on the interval $I_l$. Thus the two couples $(s_{i_j}, t_{i_j}), (s_{i_k}, t_{i_k})$ are parallel, as desired. □

We are now ready to prove Theorem 4.2.2.

*Proof of Theorem 4.2.2.* Let $w \in \mathfrak{I}_g$ be a non-trivial equation of degree $d_{\min}$, and let $\sigma : I_l \to G$ be the corresponding path. Suppose also that, between the equations of degree $d_{\min}$, the equation $w$ has the property that the length $l$ of the corresponding path is the minimum possible. This in particular implies that $w$ is cyclically reduced.

Assume by contradiction that $l > 16L^2 d_{\min}$. Then take any maximal reduction process $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ for $f \circ \sigma$, and by Proposition 4.2.16 we can find two parallel couples $(s_i, t_i), (s_j, t_j)$. We perform the corresponding parallel cancellation move (according to Lemma 4.2.12) and we obtain a path $\sigma' : I_{l'} \to G$, with corresponding equation $w' \in \mathfrak{I}_g$ with $w' \neq 1$. Notice that $l' < l$; moreover, by Lemma 4.2.13, the degree $d'$ of $w'$ satisfies $d' \leq d_{\min}$, but since $d_{\min}$ is the minimum possible this implies $d' = d_{\min}$. But then $l' < l$ contradicts the minimality of $l$. This proves the theorem. □

**Corollary 4.2.17.** *There is an algorithm that, given $H$ and $g$ such that $g$ depends on $H$, produces a non-trivial equation $w \in \mathfrak{I}_g$ of minimum possible degree.*

*Algorithm.* We first produce an upper bound $D$ on the minimum degree of an equation in $\mathfrak{I}_g$; this is done for example by computing a non-trivial equation in $\mathfrak{I}_g$ using Theorem 1.3.4, and taking its degree $D$. Given this upper bound $D$, we take all the non-trivial reduced paths $\sigma : I_l \to G$ from the basepoint to itself and of

length $l \leq 16L^2D$. For each such path $\sigma$, we check whether $f \circ \sigma$ is homotopically trivial (in linear time on a pushdown automaton, with a free reduction process), and we compute the degree of the corresponding equation $w$. We take the minimum of all the degrees of those equations: this is also the minimum possible degree for a non-trivial equation in $\mathfrak{I}_g$. □

## 4.3 The set of minimum-degree equations

In this section we describe a parallel insertion move and we show that it is an inverse to the degree-preserving cancellation moves. We also provide a few lemmas that help us manipulate sequences of insertion moves. Our aim is to provide an explicit characterization of the set of all the equations of minimum possible degree (and more generally, of the set of all the equations of a certain fixed degree).

### 4.3.1 Parallel insertion

Recall that, for every vertex $v \in \text{core}(H)$, the group $\pi_1(\text{core}(H), v)$ can be seen as a subgroup of $F_n$, by means of the injective map $\pi_1(f) : \pi_1(\text{core}(H), v) \to \pi_1(R_n, *)$, where $f : \text{core}(H) \to R_n$ is the labeling map.

**Definition 4.3.1.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Fix a couple $(s_\alpha, t_\alpha)$ such that $\sigma(s_\alpha)$ and $\sigma(t_\alpha)$ belong to* core$(H)$*; an element $u \in F_n$ is called **insertion word for $\sigma$ at** $(s_\alpha, t_\alpha)$ if it satisfies the following conditions:*

*(i) $u$ belongs to the subgroup $\pi_1(\text{core}(H), \sigma(o^-(s_\alpha))) \cap \pi_1(\text{core}(H), \sigma(o^+(t_\alpha)))$ of $F_n$.*

*(ii) $u$ begins with the label of $\sigma(s_\alpha)$, if $\sigma$ crosses $\sigma(s_\alpha)$ with the same orientation of the labeling, or with the inverse of that label, if $\sigma$ crosses $\sigma(s_\alpha)$ with opposite orientation to the labeling.*

*(iii) $u$ is cyclically reduced.*

Let $u$ be a insertion word at $(s_\alpha, t_\alpha)$ for the path $\sigma$. Then there is a unique reduced path $\tau_1 : I_r \to G$ representing $u \in \pi_1(\text{core}(H), \sigma(o^-(s_\alpha)))$; similarly, there is a unique reduced path $\tau_2 : I_r \to G$ representing $\overline{u} \in \pi_1(\text{core}(H), \sigma(o^+(t_\alpha)))$. These two paths have the same length $r$, which is also the length of the word $u$. Now cut $I_l$ at the two points $o^-(s_\alpha)$ and $o^+(t_\alpha)$, and insert an interval of length $r$ at each of these two cuts, in order to obtain an interval $I_{l+2r}$; define the map $\sigma' : I_{l+2r} \to G$ which is equal to $\sigma$ on the edges that belonged to $I_l$, and is equal to $\tau_1$ on the interval added at the cut at $o^-(s_\alpha)$, and is equal to $\tau_2$ on the interval added at the cut at $o^+(t_\alpha)$; see also Figure 4.4.

**Lemma 4.3.2** (Parallel insertion)**.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Let $(s_\alpha, t_\alpha)$ be a couple such that $\sigma(s_\alpha)$ and $\sigma(t_\alpha)$ belong to* core$(H)$ *and let $u$ be an insertion word for $\sigma$ at $(s_\alpha, t_\alpha)$. Let $\sigma' : I_{l+2r} \to G$ be the path defined as above. Then $\sigma'$ is a reduced path with $\sigma'(0) = \sigma'(1) = *$ and $f \circ \sigma'$ is homotopically trivial. Moreover, there is a maximal reduction process for $f \circ \sigma'$ containing the couples $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$.*

*Proof.* Of course we have $\sigma'(0) = \sigma'(1) = *$. The fact that $\sigma'$ is reduced follows from the fact that $\sigma$ is reduced, and from the fact that $u$ is cyclically reduced (here it is important that $u$ is cyclically reduced and begins with the label of $\sigma(s_\alpha)$ or with its inverse: otherwise the local injectivity of $\sigma'$ may fail at $o^-(s_\alpha)$). Let $e_r, ..., e_1$ be the edges of the interval which is the domain of $\tau_1$ and let $e'_1, ..., e'_r$ be the edges of the interval which is the domain of $\tau_2$; we mean that $e_r, ..., e_1$ and $e'_1, ..., e'_r$ appear in this order on $I_{l+2r}$. Then a maximal reduction process for $f \circ \sigma'$ is given by $(s_1, t_1), ..., (s_\alpha, t_\alpha), (e_1, e'_1), ..., (e_r, e'_r), (s_{\alpha+1}, t_{\alpha+1}), ..., (s_{l/2}, t_{l/2})$, and in particular $f \circ \sigma'$ is homotopically trivial, as desired. □

*Remark.* Notice that these moves of parallel insertion depend on the existence of an element $u \in \pi_1(\text{core}(H), \sigma(o^-(s_\alpha))) \cap \pi_1(\text{core}(H), \sigma(o^+(t_\alpha)))$ with some specific properties. The two subgroups $\pi_1(\text{core}(H), \sigma(o^-(s_\alpha)))$ and $\pi_1(\text{core}(H), \sigma(o^+(t_\alpha)))$ are both conjugates of $H$, so there are cases where the possibilities for $u$ are very limited (for example if $H$ is malnormal in $F_n$, meaning that every two distinct conjugates of $H$ have trivial intersection). In any case it is possible that $\sigma(o^-(s_\alpha)) = \sigma(o^+(t_\alpha))$, giving the possibility for at least some insertion move to be performed.

The following two lemmas show that the parallel insertion moves of Lemma 4.3.2 are essentially the inverse of the parallel cancellation moves of Lemma 4.2.12 which are degree-preserving as in Definition 4.2.14.

**Lemma 4.3.3.** *Let $\sigma : I_l \to G$ be a cyclically reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Suppose there are two parallel couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ and let $\sigma' : I_{l'} \to G$ be the path obtained with the cancellation move of Lemma 4.2.12. Suppose that the cancellation move is degree-preserving, and let $u$ be the word that we read when going along $\sigma([o^-(s_\beta), o^-(s_\alpha)])$. Then $\sigma$ can be obtained from $\sigma'$ with an insertion move as described in Lemma 4.3.2, using the insertion word $u$ for $\sigma'$ at $(s_\alpha, t_\alpha)$.*

*Proof.* Immediate from Lemma 4.2.15. □

**Lemma 4.3.4.** *Let $\sigma' : I_l \to G$ be a cyclically reduced path with $\sigma'(0) = \sigma'(1) = *$ and such that $f \circ \sigma'$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma'$. Let $(s_\alpha, t_\alpha)$ be a couple such that $\sigma'(s_\alpha)$ and $\sigma'(t_\alpha)$ belong to* core$(H)$ *and let $u$ be an insertion word for $\sigma'$ at $(s_\alpha, t_\alpha)$; let $\sigma$ be the path*

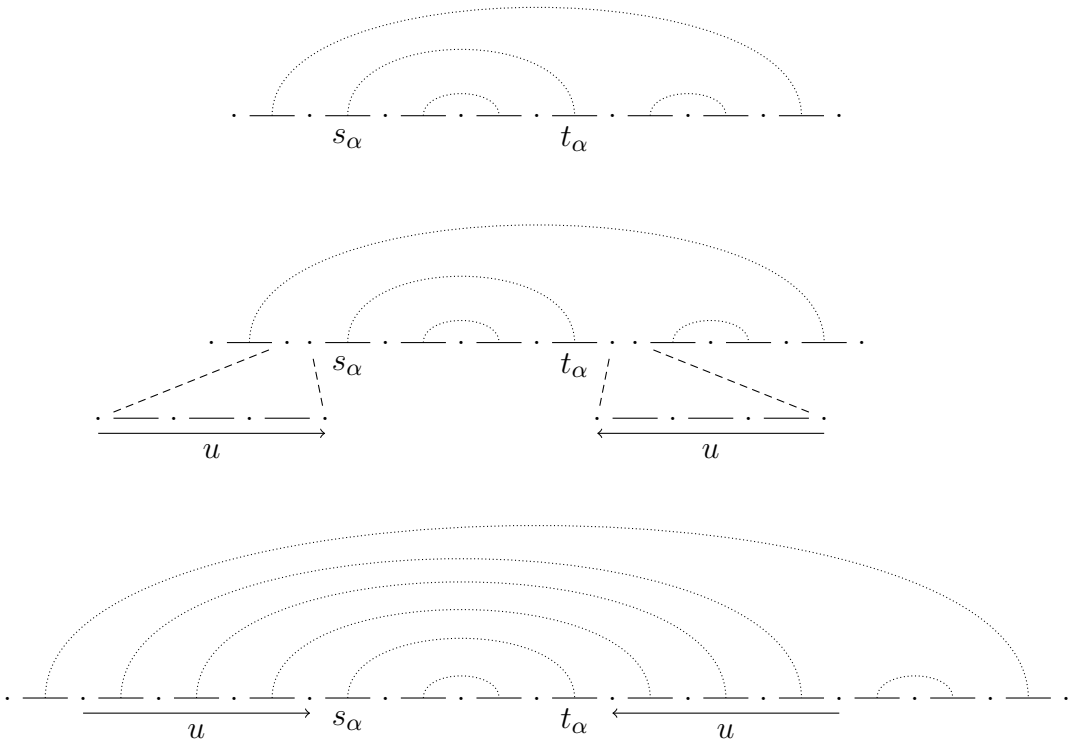


Figure 4.4: An example of an insertion move. In the image above, we can see a diagram for a maximal reduction process for $\sigma$. In the image in the middle, we see the two cuts at $o^-(s_\alpha)$ and $o^+(t_\alpha)$ and the two pieces $u$ and $\overline{u}$ ready to be inserted. In the image below, we see the result after the insertion move, and a diagram for a maximal reduction process for $\sigma'$.

*obtained from $\sigma'$ by means of the insertion move of Lemma 4.3.2. Then $\sigma'$ can be obtained from $\sigma$ by means of a cancellation move which collapses the intervals that we just added; moreover this cancellation move is degree-preserving.*

*Proof.* Immediate from the definitions. □

We are now going to prove the technical Lemmas 4.3.5, 4.3.6 and 4.3.7; these will allow us to manipulate a sequence of insertion moves. The following lemma says that, if we take a path $\sigma$ and we have two parallel insertion moves that we want to perform on $\sigma$, then we can perform them in any order that we want, and we get the same result.

**Lemma 4.3.5.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Let $(s_\alpha, t_\alpha), (s_{\alpha'}, t_{\alpha'})$ be distinct couples such that $\sigma(s_\alpha), \sigma(t_\alpha), \sigma(s_{\alpha'}), \sigma(t_{\alpha'})$ belong to $\mathrm{core}(H)$, and let $u, u'$ be insertion words for $\sigma$ at $(s_\alpha, t_\alpha), (s_{\alpha'}, t_{\alpha'})$ respectively. Perform on $\sigma$ the insertion move relative to*

*$u, (s_\alpha, t_\alpha)$ and then the insertion move relative to $u', (s_{\alpha'}, t_{\alpha'})$ in order to obtain a path $\mu_1$. Perform on $\sigma$ the insertion move relative to $u', (s_{\alpha'}, t_{\alpha'})$ and then the insertion move relative to $u, (s_\alpha, t_\alpha)$ in order to get a path $\mu_2$. Then $\mu_1$ and $\mu_2$ are the same path.*

*Proof.* The two domains of $\mu_1, \mu_2$ are defined starting with the same interval $I_l$, and adding edges as explained in Lemma 4.3.2. The edges added are the same, and the maps $\mu_1, \mu_2$ are defined in the same way on those edges. The only thing that changes is the order in which the edges are added, but the resulting paths $\mu_1$ and $\mu_2$ are the same. □

The following lemma says that, if we take a path and we perform two parallel insertion moves at the same couple of edges, then we can consolidate then into one single insertion move instead (at the same couple of edges).

**Lemma 4.3.6.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Let $(s_\alpha, t_\alpha)$ be a couple in the reduction process such that $\sigma(s_\alpha), \sigma(t_\alpha)$ belong to* core$(H)$*, and let $u, u'$ be insertion words for $\sigma$ at $(s_\alpha, t_\alpha)$. Perform on $\sigma$ the insertion move relative to $u, (s_\alpha, t_\alpha)$ and then the insertion move relative to $u', (s_\alpha, t_\alpha)$ in order to obtain a path $\mu_1$. Perform on $\sigma$ the insertion move relative to $uu', (s_\alpha, t_\alpha)$ in order to obtain a path $\mu_2$. Then $\mu_1$ and $\mu_2$ are the same path.*

*Proof.* Completely analogous to the proof of Lemma 4.3.5. □

The following Lemma 4.3.7 says that, if we take a path and we perform a parallel insertion move at a couple of edges, and then another insertion move at a couple of edges that we just added, then we can again consolidate the two insertion moves into a single one. Notice that this is slightly different from the previous Lemma 4.3.6.

**Lemma 4.3.7.** *Let $\sigma : I_l \to G$ be a reduced path with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be a maximal reduction process for $f \circ \sigma$. Let $(s_\alpha, t_\alpha)$ be a couple such that $\sigma(s_\alpha)$ and $\sigma(t_\alpha)$ belong to* core$(H)$*, and let $u$ be an insertion word for $\sigma$ at $(s_\alpha, t_\alpha)$; let $\sigma'$ be the reduced path obtained with the insertion move relative to $u, (s_\alpha, t_\alpha)$, and take a maximal reduction process for $f \circ \sigma'$ containing all the couples $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$. Let $(s', t')$ be a couple in the reduction process for $f \circ \sigma'$ that does not belong to the reduction process for $f \circ \sigma$, and let $u'$ be an insertion word for $\sigma'$ at $(s', t')$; let $\sigma''$ be the path obtained from $\sigma'$ after performing the insertion move relative to $u', (s', t')$. Then there is an insertion word $\overline{u}$ for $\sigma$ at $(s_\alpha, t_\alpha)$ such that, if we perform on $\sigma$ the insertion move relative to $\overline{u}, (s_\alpha, t_\alpha)$, we obtain $\sigma''$.*

*Proof.* Completely analogous to the proof of Lemma 4.3.5. □

### 4.3.2 Characterization of all the minimum-degree equations

Recall that $L$ is the number of edges of $G$. Let $d_{\min}$ be the minimum possible degree for a non-trivial equation $w \in \mathfrak{I}_g$.

**Theorem 4.3.8** ([Asc22b]). *Let $w \in \mathfrak{I}_g$ be a cyclically reduced equation of degree $d_{\min}$ and let $\sigma : I_l \to G$ be the corresponding reduced path. Then there is a cyclically reduced equation $w' \in \mathfrak{I}_g$ of degree $d_{\min}$ with corresponding path $\sigma' : I_{l'} \to G$, and a maximal reduction process for $\sigma'$, such that:*

*(i) The path $\sigma'$ has length $l' \leq 16L^2 d_{\min}$.*

*(ii) The path $\sigma$ can be obtained from $\sigma'$ by means of at most $l'/2$ insertion moves (as in Lemma 4.3.2), each of them performed on a distinct couple of edges of $I_{l'}$.*

*Proof.* If the length of $\sigma$ is $l > 16L^2 d_{\min}$, then by Proposition 4.2.16 we can perform a cancellation move on $\sigma$ in order to get a shorter path. The degree can't strictly increase, by Lemma 4.2.13, and can't strictly decrease, since $d_{\min}$ was minimum. Thus we obtain a strictly shorter path, whose corresponding equation has the same degree $d_{\min}$. We reiterate the process, and after a finite number of parallel cancellation moves we have to obtain a path $\sigma' : I_{l'} \to G$ with corresponding equation of degree $d_{\min}$ and of length $l' \leq 16L^2 d_{\min}$.

Since $\sigma'$ is obtained from $\sigma$ by means of a finite number of parallel cancellation moves, by Lemma 4.3.3 this means that $\sigma$ can be obtained from $\sigma'$ by means of a sequence of insertion moves of Lemma 4.3.2. Take a sequence of insertion moves $\iota_1, ..., \iota_p$ that changes $\sigma'$ into $\sigma$, and has minimum length $p$ between all such sequences.

Suppose there are two insertions $\iota_q, \iota_r$ with $q < r$ such that $\iota_r$ acts on a couple of edges that is added by $\iota_q$: then we take an innermost couple of insertions with that property, so that each transformation $\iota_j$ with $q < j < r$ acts on a couple of edges different from $\iota_r$. In particular, by Lemma 4.3.5, we can change the order in our sequence in order to bring $\iota_r$ adjacent to $\iota_q$, and we can then apply Lemma 4.3.7 in order to substitute $\iota_q, \iota_r$ with a single insertion move. This contradicts the minimality of the length $p$ of the sequence.

Thus in our sequence $\iota_1, ..., \iota_q$ we have that each insertion move acts on a couple of edges of the original interval of definition $I_{l'}$ of $\sigma'$. If two insertion moves $\iota_q, \iota_r$ with $q < r$ act on the same couple of edges of $I_{l'}$, then we reason as above, and by means of Lemmas 4.3.5 and 4.3.6 we can substitute them with a single insertion move, contradicting the minimality of $p$.

It follows that each couple of insertion moves of the sequence $\iota_1, ..., \iota_p$ acts on a different couple of edges of the original interval of definition $I_{l'}$ of $\sigma'$, and in particular $p \leq l'/2$. The conclusion follows. □

### 4.3.3 Equations of an arbitrary fixed degree

Until now we focused on the study of the equations of minimum possible degree, but the results can be generalized to equations of any fixed degree. Let $d \geq 1$ be an integer. Let $L$ be the number of edges of the graph $G$. The following proposition is similar to Proposition 4.2.16, but with the difference that this time we are looking for a parallel cancellation move which is degree-preserving.

**Proposition 4.3.9.** *Let $w \in \mathfrak{I}_g$ be a cyclically reduced equation of degree $d$ and let $\sigma : I_l \to G$ be the corresponding path; let $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ be any maximal reduction process for $f \circ \sigma$. Suppose $l > 32L^4d^2 + 16L^3d$. Then the reduction process contains two parallel couples $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ such that the corresponding parallel cancellation move is degree-preserving.*

*Proof.* Take the domain $I_l$ of $\sigma$ and remove all the edges $r$ with the following property: $r$ belongs to a couple $(s_i, t_i)$ such that at least one of $\sigma(s_i), \sigma(t_i)$ is an edge of $\text{core}_*(\langle g \rangle)$. By Lemma 4.2.7, for every edge $e$ of $\text{core}_*(\langle g \rangle)$ there are exactly $d$ edges of $I_l$ that are sent to $e$; and $\text{core}_*(\langle g \rangle)$ contains at most $L$ edges. This means that we removed from $I_l$ at most $2Ld$ edges, and thus there remain at most $2Ld + 1$ connected components, which we call $C_1, ..., C_a$ for $a \leq 2Ld + 1$. Since $l \geq 16L^3d(2Ld + 1) + 1$, there is at least one connected component $\overline{C} \in \{C_1, ..., C_a\}$ of length at least $16L^3d + 1$.

We observe that there is no couple $(s_i, t_i)$ with both $s_i, t_i \in \overline{C}$: otherwise, the interval $[o^-(s_i), o^+(t_i)] \subseteq \overline{C}$ would contain an innermost cancellation, and thus by Lemma 4.2.9 we would find an edge of $\overline{C}$ which is sent to $\text{core}_*(\langle g \rangle)$, contradiction.

Suppose the connected component $\overline{C}$ contains at least $8L^3d + 1$ edges $s_i$ belonging to couples $(s_i, t_i)$ of the cancellation process (otherwise $\overline{C}$ has to contain at least $8L^3d + 1$ edges $t_i$ belonging to couples $(s_i, t_i)$ of the reduction process, and the reasoning is analogous). To each such edge $s_i$, we associate the quintuple $(\sigma(s_i), \epsilon, \sigma(t_i), \delta, C_k)$ where $\sigma(s_i), \sigma(t_i)$ are edges of $G$ and $\epsilon, \delta \in \{+1, -1\}$ tell us the orientation with which $\sigma(s_i)$ and $\sigma(t_i)$ cross their images, and $C_k \in \{C_1, ..., C_a\} \setminus \{\overline{C}\}$ is the connected component which $t_i$ belongs to. Since we have at least $8L^3d + 1$ edges $s_i$ in $\overline{C}$ and at most $L \cdot 2 \cdot L \cdot 2 \cdot (2Ld)$ possible quintuples, there are at least two edges $s_{i_1}, s_{i_2}$ with the same associated quintuple $(\sigma(s_{i_1}), \epsilon, \sigma(t_{i_1}), \delta, C_k) = (\sigma(s_{i_2}), \epsilon, \sigma(t_{i_2}), \delta, C_k)$.

It immediately follows that $(s_{i_1}, t_{i_1})$ and $(s_{i_2}, t_{i_2})$ are parallel couples. Without loss of generality we can assume that $o^-(s_{i_1}) < o^-(s_{i_2})$; we have that the interval $[o^-(s_{i_1}), o^-(s_{i_2})]$ is contained in $\overline{C}$ and the interval $[o^+(t_{i_2}), o^+(t_{i_1})]$ is contained in $C_k$. Thus, when we perform the cancellation move relative to the parallel couples $(s_{i_1}, t_{i_1})$ and $(s_{i_2}, t_{i_2})$, we only remove edges whose image is in $\text{core}_*(H)$. We conclude from Lemma 4.2.7 that the cancellation move is degree-preserving, as desired. □

We are now ready to state and prove the analogues to Theorem 4.2.2 and to Theorem 4.3.8.

**Theorem 4.3.10** ([Asc22b])**.** *Suppose $\mathfrak{I}_g$ contains a non-trivial equation of degree $d$. Then $\mathfrak{I}_g$ contains a non-trivial equation $w$ of degree $d$ such that the corresponding path $\sigma : I_l \to G$ has length $l \leq 32L^4d^2 + 16L^3d$.*

*Proof.* Take any non-trivial equation in $\mathfrak{I}_g$ of degree $d$ and such that the corresponding path $\sigma : I_l \to G$ has minimum length $l$; in particular this implies that the equation is cyclically reduced. If $l > 32L^4d^2 + 16L^3d$ then by Proposition 4.3.9 we can perform a degree-preserving cancellation move on $\sigma$, and thus we can find a non-trivial equation in $\mathfrak{I}_g$ of degree $d$ whose corresponding path is strictly shorter, contradiction. Thus we must have $l \leq 32L^4d^2 + 16L^3d$, and the conclusion follows. □

**Corollary 4.3.11.** *There is an algorithm that, given $H \leq F_n$ finitely generated and $g \in F_n$ and an integer $d \geq 1$, tells us whether $\mathfrak{I}_g$ contains non-trivial equations of degree $d$, and, if so, produces an equation $w \in \mathfrak{I}_g$ of degree $d$.*

**Theorem 4.3.12** ([Asc22b])**.** *Let $w \in \mathfrak{I}_g$ be a cyclically reduced equation of degree $d$ and let $\sigma : I_l \to G$ be the corresponding reduced path. Then there is a cyclically reduced equation $w' \in \mathfrak{I}_g$ of degree $d$ with corresponding path $\sigma' : I_{l'} \to G$, and a maximal reduction process for $\sigma'$, such that:*

*(i) The path $\sigma'$ has length $l' \leq 32L^4d^2 + 16L^3d$.*

*(ii) The path $\sigma$ can be obtained from $\sigma'$ by means of at most $l'/2$ insertion moves (as in Lemma 4.3.2), each of them performed on a distinct couple of edges of $I_{l'}$.*

*Proof.* Completely analogous to the proof of Theorem 4.3.8. □

### 4.3.4 The set of possible degrees

Let $H \leq F_n$ be a finitely generated subgroup, let $g \in F_n$ be an element that depends on $H$, and let $\mathfrak{I}_g$ be the ideal of the equations for $g$ over $H$.

**Definition 4.3.13.** *Define $D_g = \{d \in \mathbb{N} :$ there is a non-trivial equation $w \in \mathfrak{I}_g$ of degree $d\}$.*

**Lemma 4.3.14.** *If $d, d' \in D_g$ and $k \geq 0$ then $d + d' + 2k \in D_g$.*

*Proof.* Let $w \in \mathfrak{I}_g$ be an equation of degree $d$: up to cyclic permutation, we can assume that $w$ is of the form $c_1x^{e_1}...c_\alpha x^{e_\alpha}$ with $c_1, ..., c_\alpha \in H \setminus \{1\}$ and $e_1, ..., e_\alpha \in \mathbb{Z} \setminus \{0\}$. Similarly, let $w' \in \mathfrak{I}_g$ be an equation of degree $d'$, and similarly we assume that $w' = c'_1x^{e'_1}...c'_\beta x^{e'_\beta}$ with $c'_1, ..., c'_\beta \in H \setminus \{1\}$ and $e'_1, ..., e'_\beta \in \mathbb{Z} \setminus \{0\}$. Without loss of generality, also assume that $e'_\beta > 0$ and we take $h \in H \setminus \{1, c_1\}$. Then $w'' = \overline{h}wh\overline{x}^k w' x^k$ belongs to $\mathfrak{I}_g$ and has degree $d + d' + 2k$, for any $k \geq 0$. The conclusion follows. □

Denote with $2\mathbb{N}$ the set of non-negative even numbers.

**Theorem 4.3.15.** *Exactly one of the following possibilities takes place:*

*(i) $D_g$ contains an odd number and $\mathbb{N} \setminus D_g$ is finite.*

*(ii) $D_g$ contains only even numbers and $2\mathbb{N} \setminus D_g$ is finite.*

*Proof.* If $\mathfrak{I}_g$ contains only equations of even degree, then we take any equation of even degree $d$, and by Lemma 4.3.14 we are able to obtain equations of degree $d + d + 2k$ for every $k \geq 0$. Thus in this case we have that $2\mathbb{N} \setminus D_g$ is finite.

Suppose now $\mathfrak{I}_g$ contains an equation of odd degree $d$. Then by Lemma 4.3.14 we are able to obtain equations of degree $d + d + 2k$ for every $k \geq 0$, and thus equations of every even degree big enough. In particular we are able to obtain an equation of degree $2d$, and thus by Lemma 4.3.14 we are able to obtain equations of degree $2d + d + 2k$ for every $k \geq 0$, and thus equations of every odd degree big enough. Thus in this case we have that $\mathbb{N} \setminus D_g$ is finite. □

In order to understand whether we fall into case (i) or (ii) of Theorem 4.3.15, it is enough to look at a set of normal generators for $\mathfrak{I}_g$.

**Lemma 4.3.16.** *Let $H \leq F_n$ be a finitely generated subgroup and let $g \in F_n$ be an element that depends on $H$. Suppose the set of equations $W \subseteq \mathfrak{I}_g$ generates $\mathfrak{I}_g$ as normal subgroup of $H * \langle x \rangle$, and suppose every equation $w \in W$ has even degree. Then every equation in $\mathfrak{I}_g$ has even degree.*

*Proof.* Consider the homomorphism $\phi : H * \langle x \rangle \to \mathbb{Z}/2\mathbb{Z}$ defined by $\phi(h) = 0$ for every $h \in H$ and $\phi(x) = 1$. Observe that $\phi$ sends equations of even degree to 0 and equations of odd degree to 1. Since every $w \in W$ has even degree, we have that $W$ is contained in $\ker \phi$. But then the normal subgroup $\mathfrak{I}_g$ generated by $W$ is contained in $\ker \phi$ too, and thus $\mathfrak{I}_g$ only contains equations of even degree. □

**Theorem 4.3.17.** *Given $H \leq F_n$ finitely generated and $g \in F_n$ that depends on $H$, there is an algorithm that:*

*(a) Determines whether we fall into case (i) or (ii) of Theorem 4.3.15.*

*(b) Computes the finite set $\mathbb{N} \setminus D_g$ or $2\mathbb{N} \setminus D_g$ respectively.*

*Proof.* Let $\mathfrak{I}_g = \langle\langle w_1, ..., w_k \rangle\rangle$ be a finite set of normal generators for $\mathfrak{I}_g$, which can be obtained with the algorithm of Theorem 1.3.3. According to Lemma 4.3.16, if one of $w_1, ..., w_k$ has odd degree then we fall into case (i) of Theorem 4.3.15, otherwise we fall in case (ii) of Theorem 4.3.15.

If we fall into case (i), then we take $w_i$ of degree $d_i$ odd, and with the same proof of Theorem 4.3.15 we have that $\mathbb{N} \setminus D_g \subseteq \{1, ..., 3d_i\}$. For each degree $d \in \{1, ..., 3d_i\}$ we use Corollary 4.3.11 to determine whether $d$ belongs to $D_g$. If we fall into case (ii), we perform an analogous procedure. □

## 4.4 Examples

We now provide a few examples for the reader, to illustrate the techniques introduced in this chapter. In each example $D_g$ denotes the set introduced in Definition 4.3.13 and $d_{\min}$ denotes the minimum of $D_g$.

### 4.4.1 Cyclic subgroups

Let $F_n = \langle a_1, ..., a_n \rangle$ and suppose that $H$ has rank 1, let's say $H = \langle h \rangle$ for some $h \in F_n$ with $h \neq 1$. In order for an element $g$ to depend on $H$, we must have that $g, h$ belong to the same cyclic subgroup of $F_n$. We can use $\langle H, g \rangle$ as ambient free group instead of $F_n$: without loss of generality, in the following we assume that $F_n = \langle a \rangle$ and that $H = \langle h \rangle$ where $h = a^m$ with $m \geq 1$, and that $g = a^k$ with $k \geq 0$ coprime with $m$.

The graph $G = \mathrm{core}_*(H) \vee \mathrm{core}_*(\langle g \rangle)$ here has rank 2 while $\mathrm{core}_*(\langle H, g \rangle)$ has rank 1. This means that the algorithm of Theorem 1.3.4 produces a single generator for the ideal $\mathfrak{I}_g \trianglelefteq H * \langle x \rangle$. One possible such generator $w_{m,k}$ for each $m \geq 1$ and $k \geq 0$ coprime can be obtained by means of the following recursive formula:

$$\begin{cases} w_{1,0}(h, x) = \overline{x} \\ w_{m,k}(h, x) = w_{m-k,k}(h\overline{x}, x) \text{ for } m > k \\ w_{m,k}(h, x) = w_{m,k-m}(h, x\overline{h}) \text{ for } m \leq k \end{cases}$$

Moreover, with this definition it is possible to prove by induction that $w_{m,k}(h, x)$ contains $k$ occurrences of $h$, no occurrence of $\overline{h}$, no occurrence of $x$, and $m$ occurrences of $\overline{x}$. In particular $w_{m,k} \in \mathfrak{I}_g$ is an equation of degree $m$.

*Remark.* In the case $h = a^5$ and $x = a^2$ we have the generator $w_{5,2}(h, x) = h\overline{x}^2 h\overline{x}^3$ for the ideal $\mathfrak{I}_g$. We observe that the most immediate candidate $h^2\overline{x}^5$ doesn't work, because it is contained in $\mathfrak{I}_g$ but it doesn't generate the whole ideal.

*Remark.* The following is a well-known property of one-relator groups due to Magnus: if two elements of a free group generate the same normal subgroup, then they coincide, up to conjugation and inverse. In particular, the generator $w_{m,k}$ defined above is essentially the unique generator for the ideal $\mathfrak{I}_g$.

Let $w \in \langle h, x \rangle$ be a non-trivial cyclically reduced element, which up to conjugation can be written in the form $w = h^{e_1} x^{f_1} ... h^{e_r} x^{f_r}$ with $r \geq 1$ and $e_1, ..., e_r, f_1, ..., f_r \in \mathbb{Z} \setminus \{0\}$. The condition $w \in \mathfrak{I}_g$ is equivalent to $(e_1 + ... + e_r)m + (f_1 + ... + f_r)k = 0$, and since $m, k$ are coprime this means that for some $p \in \mathbb{Z}$ we have $f_1 + ... + f_r = pm$ and $e_1 + ... + e_r = -pk$. The degree of the equation is $d = |f_1| + ... + |f_r|$.

Suppose $m = 1$. Then we have $w_{1,k} = h^k \overline{x}$. In this case $d_{\min} = 1$ and $D_g = \mathbb{N} \setminus \{0\}$.

Suppose $m \geq 2$ is even. Then $d_{\min} = 2$ and $D_g = 2\mathbb{N} \setminus \{0\}$. For the $\supseteq$ inclusion, we have the equation $[h, x^s]$ of degree $2s$ for each $s \geq 1$. For the $\subseteq$ inclusion, notice

that the unique generator $w_{m,k}$ has even degree, and thus by Lemma 4.3.16 each equation has even degree.

Suppose $m \geq 2$ is odd. Then $d_{\min} = 2$ and $D_g = \{d : d \geq 2 \text{ even}\} \cup \{d : d \geq m \text{ odd}\}$. For the $\supseteq$ inclusion, we have the equation $[h, x]$ of degree 2 and the equation $w_{m,k}$ of degree $m$, and we can use Lemma 4.3.14. For the $\subseteq$ inclusion, we notice that, if an equation is written in the form $w = h^{e_1}x^{f_1}...h^{e_r}x^{f_r}$ as above, then either $f_1 + ... + f_r = 0$, in which case the degree $d = |f_1| + ... + |f_r|$ is even, or $m \leq |f_1 + ... + f_r| \leq |f_1| + ... + |f_r| = d$.

### 4.4.2 An ideal with only even-degree equations

Let $F_2 = \langle a, b\rangle$ and consider the subgroup $H = \langle h_1, h_2\rangle$ with $h_1 = ba$ and $h_2 = ab^2\overline{a}$ and the element $g = a$. We can build the corresponding graph $G$, see Figure 4.5, and we have that $\pi_1(G, *)$ is a free group with three generators $[\mu_{h_1}], [\mu_{h_2}], [\mu_g]$, which are the homotopy classes of the reduced paths $\mu_{h_1}, \mu_{h_2}, \mu_g$ corresponding to the elements $h_1, h_2 \in H$ and $g$ respectively. We can perform a sequence of rank-preserving folding operations on $G$, see Figure 4.6, and we end up with a rose $R'$ with one $a$-labeled edge $e_1$ and two $b$-labeled edges $e_2, e_3$. Let $p : (G, *) \to (R', *)$ be the map given by the composition of the folding operations, and notice that by Proposition 1.3.2 this is a pointed homotopy equivalence: a pointed homotopy inverse can be built following the chain of folding operations, and is given by $q : (R', *) \to (G, *)$ which sends the edge $e_1$ to the path $\mu_g$, the edge $e_2$ to the path $\mu_{h_1}\overline{\mu}_g$ and the edge $e_3$ to the path $\overline{\mu}_g\mu_{h_2}\mu_g\mu_g\overline{\mu}_{h_1}$. In order to obtain generators for the kernel $\mathfrak{I}_g \leq H * \langle x\rangle$ we have to look at the image $q_*(e_3\overline{e}_2)$: we obtain that the kernel is generated (as a normal subgroup) by just one equation $\mathfrak{I}_g = \langle\langle \overline{x}h_2xx\overline{h}_1x\overline{h}_1\rangle\rangle$.

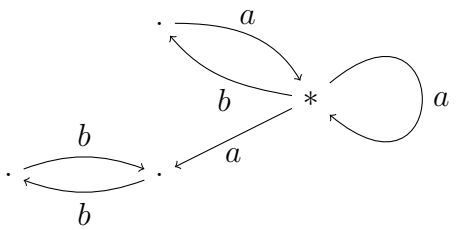


Figure 4.5: In the picture we can see the graph $G$ of Example 4.4.2. Here $H = \langle ba, ab^2\overline{a}\rangle$ and $g = a$. On the left of the basepoint we have the graph $\text{core}_*(H)$, while on the right we have $\text{core}_*(\langle g\rangle)$.

We observe that this unique generator has even degree, and thus Lemma 4.3.16 tells us that every equation in $\mathfrak{I}_g$ has even degree. We shall explain why there is no equation of degree 2, there is exactly one equation of degree 4 up to conjugation and inverse and there are equations of degree 6. By Lemma 4.3.14 it follows that $d_{\min} = 4$ and $D_g = 2\mathbb{N} \setminus \{0, 2\}$.

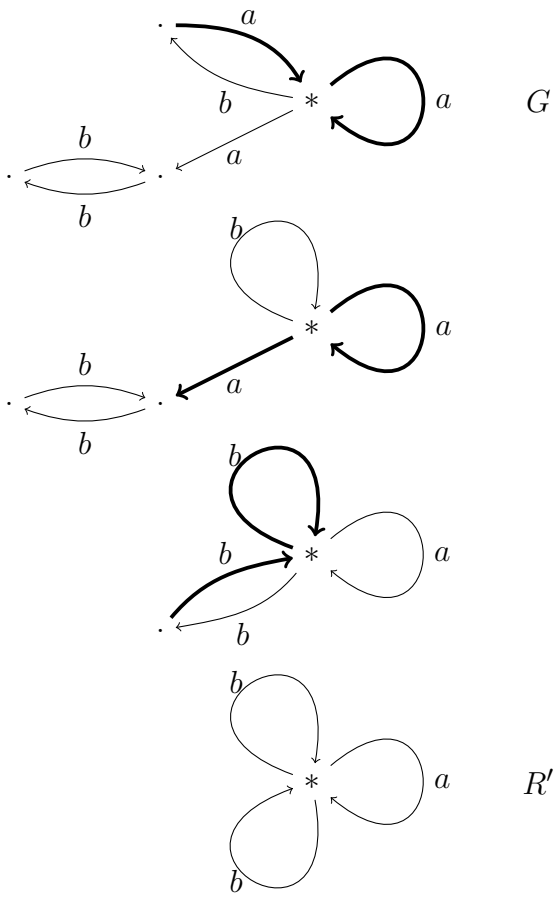


Figure 4.6: A maximal rank-preserving folding sequence for the graph $G$ of Example 4.4.2. At each step, we highlight in bold the two edges that are going to be folded in the next step.

*Remark.* In order characterize all the equations of degrees $2, 4, 6$, we could use Theorem 4.3.12; this is too long to do by hands, but quite easy to do with the aid of a computer. It is also possible to prove them with some combinatorics of the cancellation between words; we do not provide a full proof here, but rather a sketch. Consider the map between free groups $\psi : \langle h_1, h_2, x\rangle \to \langle h_1, x\rangle$ with $\psi(h_1) = h_1, \psi(x) = x, \psi(h_2) = xh_1\overline{x}h_1\overline{x}\overline{x}$ and notice that $\mathfrak{I}_g = \ker\psi$. Up to conjugation, an equation $w \in \mathfrak{I}_g$ can be written as reduced word $w(h_1, h_2, x) = u_1(h_1, h_2)x^{e_1}...u_r(h_1, h_2)x^{e_r}$. We now substitute each occurrence of $h_2$ with $xh_1\overline{x}h_1\overline{x}\overline{x}$, and each occurrence of $\overline{h_2}$ with $xx\overline{h}_1x\overline{h}_1\overline{x}$, and after this substitution we reduce the obtained word, until we get the trivial word. During the reduction process, each block $xh_1\overline{x}h_1\overline{x}\overline{x}$ and $xx\overline{h}_1x\overline{h}_1\overline{x}$, obtained from an occurrence of $h_2$ or $\overline{h}_2$, will completely cancel at some point: we take the occurrence of $h_2$ such that the corresponding block is the first to

completely cancel during the reduction process. We now look at the word $w$ near that occurrence of $h_2$ or $\overline{h}_2$, and we obtain that $w$ contains at least one of

$$h_2 x x \overline{h}_1 x,\ \overline{x} h_2 x x,\ x \overline{h}_1 \overline{x} h_2 x,\ x \overline{h}_1 x \overline{h}_1 \overline{x} h_2,\ h_2 x x \overline{h}_1 \overline{x} \overline{h}_2,\ \overline{h}_2 x x \overline{h}_1 \overline{x} h_2$$

(or of their inverses) as a subword. These can be substituted with (respectively)

$$x h_1,\ h_1 \overline{x} h_1,\ h_1 \overline{x},\ \overline{x},\ x x \overline{h}_1 \overline{x},\ x x \overline{h}_1 \overline{x}$$

in order to get a shorter (possibly not reduced) equation.

This immediately implies that $\mathfrak{I}_g$ contains no equation of degree 2, and it also allows to deduce that the only equations of degree 4 are the conjugates of the generator. With some more work, it is also possible to give a characterization of all the degree 6 equations.

### 4.4.3 An ideal with both even-degree and odd-degree equations

Consider the subgroup $H = \langle h_1, h_2 \rangle$ with $h_1 = b$ and $h_2 = ababa$ and the element $g = a$. We see the corresponding graph $G$ in figure 4.7. We can now proceed as in Example 4.4.2: we choose a maximal sequence of rank-preserving folding operations for $G$, we build a homotopy inverse to the sequence of folding operations, we obtain a generator for the normal subgroup $\mathfrak{I}_g \trianglelefteq H * \langle x \rangle$. Whatever sequence of folding operations you choose, you will always get the same generator, up to inverse and cyclic permutations, namely $\mathfrak{I}_g = \langle\langle \overline{h}_2 x h_1 x h_1 x \rangle\rangle$.

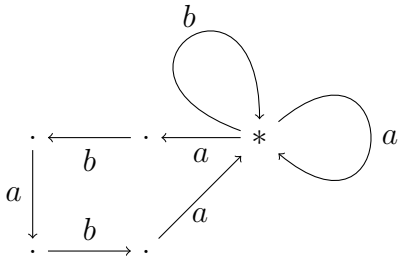


Figure 4.7: In the picture we can see the graph $G$ of Example 4.4.3. Here $H = \langle b, ababa \rangle$ and $g = a$. On the left of the basepoint we have the graph $\mathrm{core}_*(H)$. On the right of the basepoint we have the graph $\mathrm{core}_*(\langle g \rangle)$.

We have that $\mathfrak{I}_g$ contains no equation of degree 1. It contains equations of degree 2, which are exactly the ones of the form $[(h_2 h_1)^i, x h_1]$ for $i \neq 0$ up to conjugation and inverses. It also contains equations of degree 3 (possibly essentially different from the generator). By Lemma 4.3.14 it follows that $d_{\min} = 2$ and $D_g = \{d : d \geq 2\}$.

We observe that the equations of minimum possible degree are not enough in this case to generate the whole ideal: in fact, according to Lemma 4.3.16, equations of degree 2 generate a normal subgroup containing only even-degree equations.

*Remark.* As in Example 4.4.2, it is possible to characterize equations of degree 2 and 3 with Theorem 4.3.12, using a computer, or it is possible to do it by hands with some combinatorics of the cancellations inside the words; and again, for this second method, we provide a sketch below.

Consider the map between free groups $\psi : \langle h_1, h_2, x\rangle \to \langle h_1, x\rangle$ with $\psi(h_1) = h_1, \psi(x) = x, \psi(h_2) = xh_1xh_1x$ and notice that $\mathfrak{I}_g = \ker\psi$. Up to conjugation, an equation $w \in \mathfrak{I}_g$ can be written as reduced word $w(h_1, h_2, x) = u_1(h_1, h_2)x^{e_1}...u_r(h_1, h_2)x^{e_r}$. We now substitute each occurrence of $h_2$ with $xh_1xh_1x$ and each occurrence of $\overline{h_2}$ with $xh_1xh_1x$; as in Example 4.4.2 we take the occurrence of $h_2$ or $\overline{h}_2$ such that the corresponding $xh_1xh_1x$ or $\overline{x}\overline{h}_1\overline{x}\overline{h}_1\overline{x}$ is the first to completely cancel, and we look at the word $w$ near that occurrence of $h_2$ or $\overline{h}_2$. We obtain that $w$ contains at least one of

$$\overline{h}_2xh_1x,\ x\overline{h}_2x,\ xh_1x\overline{h}_2,\ \overline{h}_2xh_1h_2,\ h_2h_1x\overline{h}_2,$$

(or of their inverses) as a subword. These can be substituted with (respectively)

$$\overline{x}\overline{h}_1,\ \overline{h}_1\overline{x}\overline{h}_1,\ \overline{h}_1\overline{x},\ h_1x,\ xh_1$$

in order to get a shorter (possibly not reduced) equation.

Dealing with some cases it can be proved that equations of degree 2 are exactly the ones of the form $[(h_2h_1)^i, xh_1]$ for $i \neq 0$ up to conjugation and inverses, and one can produce equations of degree 3 which are essentially different from the generator.

### 4.4.4 An ideal with two generators

Consider the subgroup $H = \langle h_1, h_2, h_3\rangle$ with $h_1 = a^2\overline{b}\overline{a}$ and $h_2 = a^3$ and $h_3 = ba\overline{b}$ and the element $g = a^2\overline{b}$. We can see the corresponding graph $G$ in Figure 4.8 and a maximal sequence of rank-preserving folding operations in Figure 4.9. The group $\pi_1(G, *)$ is a free group with four generators $[\mu_{h_1}], [\mu_{h_2}], [\mu_{h_3}], [\mu_g]$, which are the homotopy classes of the reduced paths $\mu_{h_1}, \mu_{h_2}, \mu_{h_3}, \mu_g$ corresponding to the elements $h_1, h_2, h_3 \in H$ and $g$ respectively. At the end of the sequence of folding operations we obtain a rose $R'$ with one $b$-labeled edge $e_1$ and three $a$-labeled edges $e_2, e_3, e_4$. The map $p : (G, *) \to (R', *)$ given by the composition of the folding operations is a homotopy equivalence, according to Proposition 1.3.2, and a pointed homotopy inverse is $q : (R', *) \to (G, *)$ which sends the edge $e_1$ to the path $\mu_{h_3}\overline{\mu}_g\overline{\mu}_{h_1}\mu_g$, the edge $e_2$ to the path $\overline{\mu}_g\mu_{h_1}\mu_g\overline{\mu}_{h_3}\overline{\mu}_g\mu_{h_2}$, the edge $e_3$ to the path $\overline{\mu}_{h_1}\mu_g$ and the edge $e_4$ to the path $\overline{\mu}_g\mu_{h_1}\mu_g\mu_{h_3}\overline{\mu}_g\overline{\mu}_{h_1}\mu_g$. We look at the images $q_*(e_2\overline{e}_3)$ and $q_*(e_4\overline{e}_3)$ and we obtain that the kernel $\mathfrak{I}_g \leq H * \langle x\rangle$ is generated (as normal subgroup) by the equations $\mathfrak{I}_g = \langle\langle \overline{x}h_1x\overline{h}_3\overline{x}h_2\overline{x}h_1, \overline{x}h_1xh_3\overline{x}\rangle\rangle$.

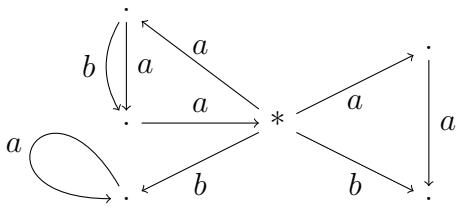


Figure 4.8: In the picture we can see the graph $G$ of Example 4.4.4. Here $H = \langle a^2\overline{b}\overline{a}, a^3, ba\overline{b}\rangle$ and $g = a^2\overline{b}$. On the left of the basepoint we have the graph $\mathrm{core}_*(H)$. On the right of the basepoint we have the graph $\mathrm{core}_*(\langle g\rangle)$.

It is easy to show that $\mathfrak{I}_g$ contains equations of degree 2, but not equations of degree 1. It follows that $d_{\min} = 2$ and $D_g = \mathbb{N} \setminus \{0, 1\}$. We observe that, despite $d_{\min} = 2$, equations of degree 2 are not enough to generate the whole ideal $\mathfrak{I}_g$; in fact, the equations of degree 2 generate a normal subgroup containing only even-degree equations (see Lemma 4.3.16), while $\mathfrak{I}_g$ also contains equations of odd degree.

## 4.5 Equations in more variables

We point out that most of the results of this chapter can be generalized to equations in more than one variable. Let $F_n$ be a free group generated by $n$ elements $a_1, ..., a_n$. Let $H \leq F_n$ be a finitely generated subgroup and let $\langle x_1\rangle, \langle x_2\rangle, ..., \langle x_m\rangle$ be infinite cyclic groups.

**Definition 4.5.1.** *An **equation** with coefficients in $H$ is an element $w \in H * \langle x_1\rangle * ... * \langle x_m\rangle$.*

**Definition 4.5.2.** *Define the **multi-degree** of an equation $w \in H * \langle x_1\rangle * ... * \langle x_m\rangle$ as the $m$-tuple $(d_1, ..., d_m)$ of integer numbers, where $d_i$ is the number of occurrences of $x_i$ and of $\overline{x}_i$ in the cyclic reduction of $w$.*

For $(g_1, ..., g_m) \in (F_n)^m$ we define the map $\varphi_{g_1,...,g_m} : H * \langle x_1\rangle * ... * \langle x_m\rangle \to F_n$ such that $\varphi_{g_1,...,g_m}\big|_H$ is the inclusion and $\varphi_{g_1,...,g_m}(x_i) = g_i$ for $i = 1, ..., m$.

**Definition 4.5.3.** *We say that an $m$-tuple $(g_1, ..., g_m) \in (F_n)^m$ is a **solution** to the equation $w \in H * \langle x_1\rangle * ... * \langle x_m\rangle$ if $w \in \ker \varphi_{g_1,...,g_m}$.*

**Definition 4.5.4.** *For $(g_1, ..., g_m) \in (F_n)^m$ we define the **ideal** $\mathfrak{I}_{g_1,...,g_m}$ to be the normal subgroup $\mathfrak{I}_{g_1,...,g_m} = \ker \varphi_{g_1,...,g_m} \trianglelefteq H * \langle x_1\rangle * ... * \langle x_m\rangle$.*

**Definition 4.5.5.** *We say that $(g_1, ..., g_m) \in (F_n)^m$ **depends on** $H$ if $\mathfrak{I}_{g_1,...,g_m}$ is non-trivial.*

Fix now an $m$-tuple $(g_1, ..., g_m) \in (F_n)^m$. As we did in the one-variable case, we now want to see equations as paths in a suitable graph. Let $G = \mathrm{core}_*(H) \vee$

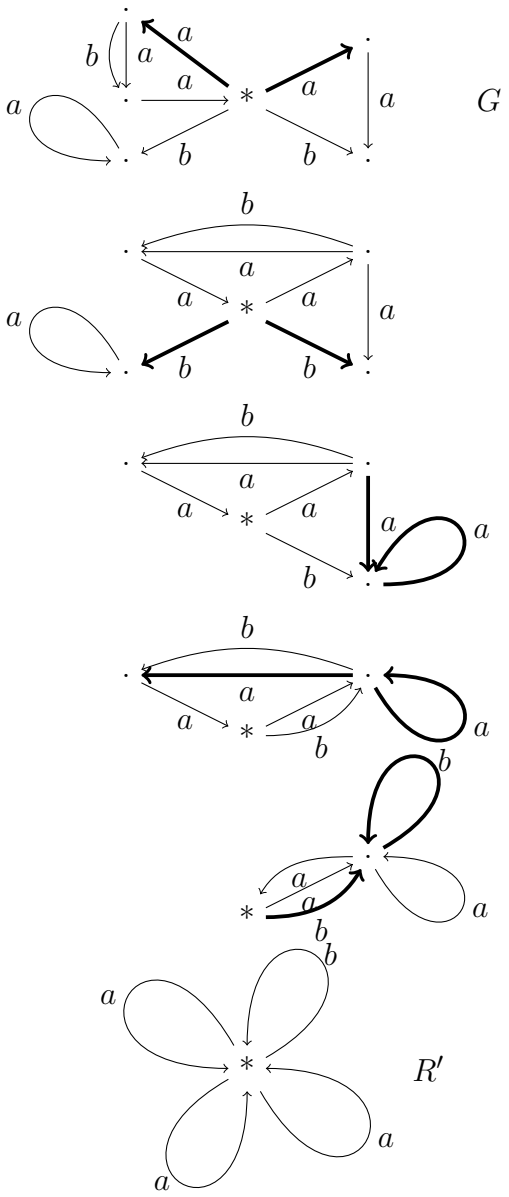


Figure 4.9: A maximal rank-preserving folding sequence for the graph $G$ of Example 4.4.4. At each step, we highlight in bold the two edges that are going to be folded in the next step.

$\mathrm{core}_*(\langle g_1\rangle) \vee ... \vee \mathrm{core}_*(\langle g_m\rangle)$ be the labeled graph given by the disjoint union of $\mathrm{core}_*(H), \mathrm{core}_*(\langle g_1\rangle), ..., \mathrm{core}_*(\langle g_m\rangle)$, where we identify all the basepoints to a unique point. Let also $f : G \to R_n$ be the labeling map.

With the same argument as in Theorem 1.3.3 we can prove the following theorem:

**Theorem 4.5.6.** *We have the following:*

*(i) The ideal* $\mathfrak{I}_{g_1,...,g_m} \trianglelefteq H * \langle x_1 \rangle * ... * \langle x_m \rangle$ *is finitely generated as a normal subgroup.*
*(ii) The set of generators for* $\mathfrak{I}_{g_1,...,g_m}$ *can be taken to be a subset of a basis for* $H * \langle x_1 \rangle * ... * \langle x_m \rangle$.
*(iii) There is an algorithm that, given* $H$ *and* $g_1, ..., g_m$*, computes a finite set of normal generators for* $\mathfrak{I}_{g_1,...,g_m}$ *which is also a subset of a basis for* $H * \langle x_1 \rangle * ... * \langle x_m \rangle$.

We have an isomorphism $\theta : H * \langle x_1 \rangle * ... * \langle x_m \rangle \to \pi_1(G, *)$ so that we can define the same correspondence as in Definitions 4.1.1 and 4.1.2. Non-trivial equations $w \in \mathfrak{I}_{g_1,...,g_m}$ correspond to reduced paths $\sigma : I_l \to G$ with $\sigma(0) = \sigma(1) = *$ and such that $f \circ \sigma$ is homotopically trivial (relative to its endpoints). The following three lemmas relate the degree of an equation to its corresponding path, and the proofs are exactly the same as for Lemmas 4.2.7, 4.2.9 and 4.2.10.

**Lemma 4.5.7.** *Let* $\sigma : I_l \to G$ *be a cyclically reduced path. For* $i = 1, ..., m$ *let* $e_i$ *be any edge of* $G$ *that belongs to the subgraph* $\mathrm{core}(\langle g_i \rangle)$*. Let also* $(d_1, ..., d_m)$ *be the multi-degree of the equation* $w$ *corresponding to* $\sigma$*. Then the path* $\sigma$ *crosses the edge* $e_i$ *exactly* $d_i$ *times (in either direction).*

**Lemma 4.5.8.** *Let* $\sigma : I_l \to G$ *be a reduced path with* $\sigma(0) = \sigma(1) = *$ *and such that* $f \circ \sigma$ *is homotopically trivial (relative to its endpoints); let* $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ *be a maximal reduction process for* $f \circ \sigma$*. Let* $(s_i, t_i)$ *be an innermost cancellation. Then, among* $\sigma(s_i)$ *and* $\sigma(t_i)$*, one is an edge of* $\mathrm{core}_*(H)$ *and the other is an edge of* $\mathrm{core}_*(\langle g_1 \rangle) \vee ... \vee \mathrm{core}_*(\langle g_m \rangle)$.

**Lemma 4.5.9.** *Let* $\sigma : I_l \to G$ *be a cyclically reduced path with* $\sigma(0) = \sigma(1) = *$ *and such that* $f \circ \sigma$ *is homotopically trivial (relative to its endpoints); let* $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ *be a maximal cancellation process for* $f \circ \sigma$*. Let* $(d_1, ..., d_m)$ *be the multi-degree of the equation* $w \in \mathfrak{I}_{g_1,...,g_m}$ *corresponding to* $\sigma$*. Then the reduction process contains at most* $2(d_1 + ... + d_m)$ *innermost cancellations.*

We can define a parallel cancellation move, in the exact same way as in Definition 4.2.11 and Lemma 4.2.12. Lemma 4.2.15 about degree-preserving parallel cancellation moves remains true too, where degree-preserving means that the equations $w, w'$, before and after the parallel cancellation move respectively, have the same multi-degree $(d_1, ..., d_m) = (d'_1, ..., d'_m)$.
It is also possible to define a parallel insertion move, exactly as in Definition 4.3.1 and Lemma 4.3.2. Lemmas 4.3.3, 4.3.4, 4.3.5, 4.3.6, 4.3.7 remain true.
In the exact same way as we proved Proposition 4.3.9 and Theorems 4.3.10 and 4.3.12, we can prove the following.

**Proposition 4.5.10.** *Let* $w \in \mathfrak{I}_{g_1,...,g_m}$ *be a cyclically reduced equation of multi-degree* $(d_1, ..., d_m)$ *and let* $\sigma : I_l \to G$ *be the corresponding path; let* $(s_1, t_1), ..., (s_{l/2}, t_{l/2})$ *be any maximal reduction process for* $f \circ \sigma$*. Suppose* $l > 32L^4(d_1 + ... + d_m)^2 + 16L^3(d_1 + ... + d_m)$*. Then the reduction process contains two parallel couples* $(s_\alpha, t_\alpha), (s_\beta, t_\beta)$ *such that the corresponding parallel cancellation move is degree-preserving.*

**Theorem 4.5.11.** *Suppose $\mathfrak{I}_{g_1,...,g_m}$ contains a non-trivial equation of degree $(d_1, ..., d_m)$. Then $\mathfrak{I}_{g_1,...,g_m}$ contains non-trivial equation $w$ of degree $(d_1, ..., d_m)$ such that the corresponding path $\sigma : I_l \to G$ has length $l \leq 32L^4(d_1 + ... + d_m)^2 + 16L^3(d_1 + ... + d_m)$.*

**Corollary 4.5.12.** *There is an algorithm that, given $H, g_1, ..., g_m$ and an $m$-tuple $(d_1, ..., d_m)$ of non-negative integers, tells us whether $\mathfrak{I}_{g_1,...,g_m}$ contains non-trivial equations of multi-degree $(d_1, ..., d_m)$, and, if so, produces an equation $w \in \mathfrak{I}_{g_1,...,g_m}$ of multi-degree $(d_1, ..., d_m)$.*

**Theorem 4.5.13.** *Let $w \in \mathfrak{I}_{g_1,...,g_m}$ be a cyclically reduced equation of multi-degree $(d_1, ..., d_m)$ and let $\sigma : I_l \to G$ be the corresponding path. Then there is a cyclically reduced equation $w' \in \mathfrak{I}_{g_1,...,g_m}$ of degree $(d_1, ..., d_m)$ with corresponding path $\sigma' : I_{l'} \to G$, and a maximal reduction process for $\sigma'$, such that:*

*(i) The path $\sigma'$ has length $l' \leq 32L^4(d_1 + ... + d_m)^2 + 16L^3(d_1 + ... + d_m)$.*

*(ii) The path $\sigma$ can be obtained from $\sigma'$ by means of at most $l'/2$ insertion moves, each of them performed on a distinct couple of edges of $I_{l'}$.*

# 5 | Equations over free groups and context-free languages

In the same notation as in the previous chapter, we consider a free group $F_n$ generated by $n$ elements $a_1, ..., a_n$, and given a finitely generated subgroup $H \leq F_n$ and an element $g \in F_n$, we want to study the ideal $\mathfrak{I}_g$ of the equations for $g$ over $H$. However here we take a different approach, based on the notion of context-free language.

Context-free languages have been widely studied in computer science, and they arise naturally in group theory. In particular, a celebrated theorem of D. E. Muller and P. E. Schupp states that the word problem of a finitely generated group is context-free if and only if the group is virtually free; for more details see [MS83]. The results of this chapter are based on the fact that the ideal $\mathfrak{I}_g$ is context-free as a subset of $H * \langle x \rangle$: this allows us to apply to our problem a wide family of tools that have been developed in computer science. We are thus able to improve several of the results of the previous Chapter 4, in particular those concerning the running-time complexity of the algorithms. Context-free languages also allow us to obtain information about the growth of the number of equations in $\mathfrak{I}_g$ of a certain degree $d$ as a function of their length.

In Section 5.1 we briefly recall the notion of context-free grammar and language, and we review some classical results about context-free languages. We also recall the notion of context-free subset of a free group, which will be fundamental in the subsequent discussion.

In Section 5.2 we prove that the ideal $\mathfrak{I}_g$ is a context-free subset of $H * \langle x \rangle$, and we provide an algorithm that determines whether the ideal is trivial or not. This algorithm is very different in nature from the one provided in [RV21]. We also introduce the set $\mathfrak{J}_{g,d}$ of all equations in $\mathfrak{I}_g$ of degree $d$, and we prove that $\mathfrak{J}_{g,d}$ is context-free too; this provides an algorithm that determines whether $\mathfrak{I}_g$ contains equations of a given degree $d$ or not.

In Section 5.3 we study the growth of the number of equations in the sets $\mathfrak{I}_g$ and $\mathfrak{J}_{g,d}$. We prove that this growth is either polynomial or exponential, and we show that the type of growth can be determined algorithmically (including the degree of the growth, in case it's polynomial).

In Section 5.4 we focus on the running time of the algorithms, proving that most of the algorithms can be run in polynomial time.

## 5.1 Context-free languages and free groups

In this section we introduce the notions of regular language and of context-free language, which are widely used in computer science; see for example [HMU06] for an exhaustive introduction to the topic. We also discuss the notion of context-free subset of a free group.

Let $\mathcal{A}$ be a finite set, which we call the **alphabet**; the elements of $\mathcal{A}$ will be called **letters**. The free monoid over $\mathcal{A}$ will be denoted $\mathcal{A}^*$; the elements of $\mathcal{A}^*$ will be called **words**. The length of a word $w \in \mathcal{A}^*$, denoted $l(w)$, is the number of letters in the word. A **language** is any subset $\mathcal{L} \subseteq \mathcal{A}^*$.

### 5.1.1 Regular languages

**Definition 5.1.1.** *A **finite automaton** is a quadruple $(Q, \delta, Q_0, Q_f)$ where:*

*(i) $Q$ is a finite set, whose elements are called **states** of the automaton.*

*(ii) $\delta : Q \times \mathcal{A} \to \wp(Q)$ is a function, called **transition function**.*

*(iii) $Q_0$ is a subset of $Q$, whose elements are called **initial states**.*

*(iv) $Q_f$ is a subset of $Q$, whose elements are called **final states**.*

The transition function tells us that, if we are at a state $q \in Q$ and we read a letter $c \in \mathcal{A}^*$, then we are allowed to move to any of the states in the set $\delta(q, c) \subseteq Q$. More generally, given a subset of states $T \subseteq Q$ and a word $w \in \mathcal{A}^*$, define $\delta(T, w)$ to be the subset of states where we are allowed to move by reading $w$; to be formal, if $w = w_1 ... w_{l(w)}$, then we define inductively $T_0 = T$ and $T_{i+1} = \bigcup_{q \in T_i} \delta(q, w_{i+1})$, and we define $\delta(T, w) = T_{l(w)}$.

We say that the automaton **accepts** a word $w$ if the intersection of $\delta(Q_0, w)$ and $Q_f$ is nonempty; this means that there is at least one way to start at one of the initial states and to move around the automaton ending up in a final state.

**Definition 5.1.2.** *A finite automaton $(Q, \delta, Q_0, Q_f)$ is called **deterministic** if it satisfies:*

*(i) $\delta(q, a)$ is a singleton for each $q \in Q$ and $a \in \mathcal{A}$.*

*(ii) $Q_0$ is a singleton.*

*For a deterministic finite automaton we call $q_0$ the unique initial state, i.e. $Q_0 = \{q_0\}$. With an abuse of notation, we also consider the transition function as a map $\delta : Q \times \mathcal{A} \to Q$.*

Condition (i) means that there is a unique initial state; condition (ii) means that from each state and for each letter that we read, we can jump on exactly one state. We don't require any condition on the set $Q_f$.

**Proposition 5.1.3.** *Let $\mathcal{L} \subseteq \mathcal{A}^*$ be a language. Then the following are equivalent:*

*(i) There is a finite automaton such that $w \in \mathcal{L}$ if and only if the automaton accepts $w$.*
*(ii) There is a deterministic finite automaton such that $w \in \mathcal{L}$ if and only if the automaton accepts $w$.*
*In this case $\mathcal{L}$ is called* ***regular language****.*

*Proof.* See Theorem 2.12 of [HMU06]. □

**Proposition 5.1.4.** *Let $\mathcal{L}, \mathcal{L}'$ be regular languages. Then we have the following:*
*(i) $\mathcal{L} \cup \mathcal{L}'$ is a regular language.*
*(ii) $\mathcal{L} \cap \mathcal{L}'$ is a regular language.*
*(iii) The complement $\mathcal{L}^c$ of $\mathcal{L}$ is a regular language.*

*Proof.* For part (iii) see Theorem 4.5 of [HMU06]. For part (ii) see Theorem 4.8 of [HMU06]. For part (i) use that $\mathcal{L} \cup \mathcal{L}' = ((\mathcal{L})^c \cap (\mathcal{L}')^c)^c$. □

### 5.1.2 Context-free grammars

**Definition 5.1.5.** *A* ***context-free grammar*** *is a triple $(\mathcal{N}, \mathcal{P}, S)$ where*
*(i) $\mathcal{N}$ is a finite set, disjoint from $\mathcal{A}$, whose elements are called* ***non-terminal symbols****.*
*(ii) $\mathcal{P}$ is a finite set of* ***production rules*** *$(N, u)$ with $N \in \mathcal{N}$ and $u \in (\mathcal{A} \cup \mathcal{N})^*$.*
*(iii) $S \in \mathcal{N}$ is a fixed element, called the* ***initial symbol****.*

We will denote a production rule $(N, u) \in \mathcal{P}$ also as $N \to u$. If there are several production rules $(N, u), (N, u'), ... \in \mathcal{P}$ for the same non-terminal symbol $N$, we will sometimes write

$$N \quad \to \quad u \quad | \quad u' \quad | \quad ...$$

Production rules allow us to substitute a non-terminal symbol with a sequence of both letters and non-terminal symbols, as follows. Suppose we are given a word $v \in (\mathcal{A} \cup \mathcal{N})^*$ such that the $k$-th symbol of $v$ is $N \in \mathcal{N}$, and suppose we are given a production rule $(N, u) \in \mathcal{P}$. Then we take that specific occurrence of $N$ in $v$, and substitute it with $u$, in order to obtain a new word $v'$. In this case, we denote $v \to_{k,(N,u)} v'$.

**Definition 5.1.6.** *Given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$, a* ***derivation*** *is a sequence*

$$v_0 \to_{k_0,(S,u_0)} v_1 \to_{k_1,(N_1,u_1)} v_2 \to ... \to_{k_{r-1},(N_{r-1},u_{r_1})} v_r$$

*with $v_0, v_1, ..., v_{r-1}, v_r \in (\mathcal{A} \cup \mathcal{N})^*$. In this case we say that $v_0$* ***derives*** *$v_r$ and we denote $v_0 \overset{*}{\to} v_r$.*

Given a context-free grammar, we can produce a language; we start with the word given by the single initial symbol $S$, and we inductively apply substitutions, according to the given production rules, until we get a word which doesn't contain any non-terminal symbol. The set of words that can be obtained in this way is the language associated to our context-free grammar.

**Definition 5.1.7.** *Let $\mathcal{L} \subseteq \mathcal{A}^*$ be a language. Then $\mathcal{L}$ is called **context-free language** if there is a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ such that $w \in \mathcal{L}$ if and only if $S \xrightarrow{*} w$.*

Context-free languages are closed under finite union, but not under finite intersection or complement; instead, they are closed under intersection with a regular language.

**Proposition 5.1.8.** *If $\mathcal{L}, \mathcal{L}'$ are a context-free languages, then $\mathcal{L} \cup \mathcal{L}'$ is a context-free language. Moreover there is an algorithm that, given context-free grammars for $\mathcal{L}, \mathcal{L}'$, produces a context-free grammar for $\mathcal{L} \cup \mathcal{L}'$.*

*Proof.* See Theorem 7.24 of [HMU06]. □

**Proposition 5.1.9.** *If $\mathcal{L}$ is a context-free language and $\mathcal{R}$ is a regular language, then $\mathcal{L} \cap \mathcal{R}$ is a context-free language. Moreover there is an algorithm that, given a context-free grammar for $\mathcal{L}$ and a finite automaton for $\mathcal{L}$, produces a context-free grammar for $\mathcal{L} \cap \mathcal{R}$.*

*Proof.* See Theorem 7.27 of [HMU06]. □

Context-free languages are also closed under inverse image. Let $\mathcal{A}, \mathcal{B}$ be two alphabets and let $\psi : \mathcal{A} \to \mathcal{B}^*$ be a set map. Then there is a unique extension $\psi : \mathcal{A}^* \to \mathcal{B}^*$ that preserves concatenations (which we still denote by $\psi$ with an abuse of notation).

**Proposition 5.1.10.** *If $\mathcal{L} \subseteq \mathcal{B}^*$ is a context-free language, then $\psi^{-1}(\mathcal{L}) \subseteq \mathcal{A}^*$ is a context-free language. Moreover there is an algorithm that, given a context-free grammar for $\mathcal{L}$ and the map $\psi$, produces s context-free grammar for $\psi^{-1}(\mathcal{L})$.*

*Proof.* See Theorem 7.30 of [HMU06]. □

The most important property that we want to test about context-free grammar is whether the corresponding context-free language is empty or not.

**Proposition 5.1.11.** *There is an algorithm that, given a context-free grammar, determines whether the corresponding language is empty or not, and in case produces a word in the language.*

*Proof.* See Section 7.4.3 of [HMU06]. □

*Remark.* The constructions described in [HMU06] for the proofs of Propositions 5.1.9 and 5.1.10 are indeed algorithmic, but require the notion of push-down automaton, and involve a process of conversion from context-free grammar to push-down automaton and vice-versa. Later, in Section 5.4, we will describe constructions, for those grammars specific for our needs, that don't require the notion of push-down automaton; our aim is to optimize the algorithms about equations in order to make them run faster. Section 5.4 also includes a discussion about the algorithm of Proposition 5.1.11.

### 5.1.3 Context-free languages and free groups

Let $F_n$ be a free group generated by $a_1, ..., a_n$; we denote with $\overline{w} \in F_n$ the inverse of an element $w \in F_n$. Consider the alphabet $\mathcal{A} = \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$ and the set $\mathcal{A}^*$ of finite words over this alphabet. There is a natural projection map $\eta : \mathcal{A}^* \to F_n$ that sends each word in $\mathcal{A}^*$ to the element of $F_n$ represented by that word.

**Definition 5.1.12.** *Let $W \subseteq F_n$ be any subset. Define the language $\mathcal{L}(W) := \eta^{-1}(W) \subseteq \mathcal{A}^*$. This means that $\mathcal{L}(W)$ is the set of all (reduced and unreduced) words that represent an element of $W$.*

**Definition 5.1.13.** *Let $W \subseteq F_n$ be any subset. We say that $W$ is **context-free** if $\mathcal{L}(W)$ is a context-free language.*

*Remark.* As it is stated, the above definition seems to depend on the chosen basis for $F_n$. As we will always work with a fixed basis for $F_n$, this would not be an issue for us. However, we still prefer to point out that the above definition is independent on the chosen basis of $F_n$. To be precise, let $g_1, ..., g_k$ be any finite set of generators for $F_n$, and let $\mathcal{L}_{g_1,...,g_k}(W)$ be the set of words in $g_1, ..., g_k, \overline{g}_1, ..., \overline{g}_k$ that represent an element of $W$: then it easily follows from Proposition 5.1.10 that $\mathcal{L}_{g_1,...,g_k}(W)$ is context-free if and only if $\mathcal{L}(W)$ is context-free.

**Lemma 5.1.14.** *The trivial subset $\{1\} \subseteq F_n$ is context-free.*

*Proof.* Consider the grammar given by a single non-terminal symbol $T$ and production rules

$$T \quad \to \quad \epsilon \quad | \quad TT \quad | \quad a_i T \overline{a}_i \quad | \quad \overline{a}_i T a_i \qquad \text{for } i = 1, ..., n. \qquad \square$$

It is natural to ask what happens if, instead of considering the language $\mathcal{L}(W)$ of all words representing an element of $W$, we consider the language of all *reduced* words representing an element of $W$. This corresponds essentially to taking the intersection with a regular language.

**Proposition 5.1.15.** *Let $W$ be any subset of $F_n$. The the following are equivalent:*
*(i) The language $\mathcal{L}(W)$ of all words representing elements of $W$ is context-free.*

*(ii) The language $\mathcal{L}_{\mathrm{r}}(W)$ of all reduced words representing elements of $W$ is context-free.*

*Proof.* (i)⇒(ii). We take the context-free language $\mathcal{L}(W)$ and we intersect it with the regular language of all words in $\mathcal{A}^*$ which are reduced. We obtain the language $\mathcal{L}_{\mathrm{r}}(W)$ which, by Proposition 5.1.9, is context-free.

(ii)⇒(i). Suppose $(\mathcal{N}, \mathcal{P}, S)$ is a context-free grammar generating $\mathcal{L}_{\mathrm{r}}(W)$ and let $(\mathcal{N}_1, \mathcal{P}_1, T)$ be a grammar for the language $\mathcal{L}(1)$, for example the one described in Lemma 5.1.14. For every production rule

$$N \to u_1 ... u_l$$

in $\mathcal{P}$ with $u_1, ..., u_l \in (\mathcal{A} \cup \mathcal{N})$, we define a production rule

$$N \to T u_1 T u_2 T ... u_{l-1} T u_l T$$

and we call $\mathcal{P}'$ the set of all these new production rules. Then the context-free grammar $(\mathcal{N} \cup \mathcal{N}_1, \mathcal{P}' \cup \mathcal{P}_1, S)$ generates the language $\mathcal{L}(W)$, as desired. □

The following proposition is a technical result which will be useful in what follows.

**Proposition 5.1.16.** *Let $W$ be a subset of $F_n$, and let $W'$ be the subset of $F_n$ of all elements which are conjugate to an element of $W$. If $W$ is context-free, then $W'$ is context-free.*

*Proof.* Let $(\mathcal{N}, \mathcal{P}, S)$ be a context-free grammar for the language $\mathcal{L}_{\mathrm{r}}(W)$ of all reduced words representing an element of $W$. Add a new non-terminal symbol $S'$ in order to obtain $\mathcal{N}' = \mathcal{N} \cup \{S'\}$. Take the grammar $\mathcal{P}$ and add the production rules $S' \to a_i S' \overline{a}_i$ and $S' \to \overline{a}_i S' a_i$ for $i = 1, ..., n$, and the production rule $S' \to S$, in order to obtain a new grammar $\mathcal{P}'$. Then the grammar $(\mathcal{N}', \mathcal{P}', S')$ generates a language $\mathcal{L}'$ that contains only words in $W'$, and that contains all the reduced words representing elements in $W'$. We intersect $\mathcal{L}'$ with the regular language of all words which are reduced, and we obtain the language $\mathcal{L}_{\mathrm{r}}(W')$, which by Proposition 5.1.9 is context-free. The conclusion follows. □

## 5.2 Context-free languages and equations over a free group

Let $F_n$ be a free group on $n$ generators $a_1, ..., a_n$. Let $H \leq F_n$ be a finitely generated subgroup of rank $r$, and let $h_1, ..., h_r$ be a basis for $H$. Let $g \in F_n$ be any element. Let $\mathfrak{I}_g \trianglelefteq H * \langle x \rangle$ be the ideal of the equations for $g$ over $H$. We here prove that $\mathfrak{I}_g$ is a context-free subset of $H * \langle x \rangle$: this provides an algorithm that tells whether $\mathfrak{I}_g$ is trivial or not, and in case also produces a non-trivial equation for $g$ over $H$. An

algorithm for this purpose already follows from [RV21] (see Theorem 4.1.5), but our algorithm is completely different in nature.

We then turn our attention to the subsets $\mathfrak{J}_{g,d} \subseteq \mathfrak{I}_g$ of the equations of a certain fixed degree $d \geq 1$. We prove that these subsets of $H * \langle x \rangle$ are context-free too: this gives an algorithm that tells us whether there is a non-trivial equation of degree $d$.

### 5.2.1 Ideals of equations are context-free

**Theorem 5.2.1.** *The set $\mathfrak{I}_g$ is context-free as a subset of $H * \langle x \rangle$.*

*Proof.* Consider the alphabets $\mathcal{A} = \{a_1, \overline{a}_1, ..., a_n, \overline{a}_n\}$ and $\mathcal{B} = \{h_1, \overline{h}_1, ..., h_r, \overline{h}_r, x, \overline{x}\}$ where $h_1, ..., h_r$ is a basis for $H$. Consider a map $\psi : \mathcal{B}^* \to \mathcal{A}^*$ such that $\psi$ sends each element of $\mathcal{B}$ to a word representing that element in the free group $F_n$ (and $x$ and $\overline{x}$ to words representing the elements $g$ and $\overline{g}$ respectively), and such that $\psi$ preserves concatenations. By Lemma 5.1.14 we have that $\mathcal{L}(1) \subseteq \mathcal{A}^*$ is context-free. By Proposition 5.1.10 we obtain that $\psi^{-1}(\mathcal{L}(1)) \subseteq \mathcal{B}^*$ is context-free. But $\psi^{-1}(\mathcal{L}(1)) = \mathcal{L}(\mathfrak{I}_g)$ and the conclusion follows. □

*Remark.* Let $N$ be a normal subgroup of a finitely generated free group $F$. We point out that $N$ being context-free is the same as the quotient $F/N$ being virtually free. This is a result of D. E. Muller and P. E. Schupp; for more details see [MS83].

**Theorem 5.2.2.** *There is an algorithm that, given $H \leq F_n$ finitely generated and $g \in F_n$, tells us whether $\mathfrak{I}_g$ contains a non-trivial equation or not and, in case it does, produces a non-trivial equation in $\mathfrak{I}_g$.*

*Proof.* We proceed as in the proof of Proposition 5.2.1. Lemma 5.1.14 provides an explicit grammar for the language $\mathcal{L}(1) \subseteq \mathcal{A}^*$. According to Proposition 5.1.10, we can algorithmically build a grammar for the language $\mathcal{L}(\mathfrak{I}_g)$. It is easy to construct a finite automaton for the regular language $\mathcal{R}$ of all non-trivial reduced words in $\mathcal{B}^*$: by Proposition 5.1.9 we can algorithmically obtain a context-free grammar for the language $\mathcal{L}(\mathfrak{I}_g) \cap \mathcal{R}$ of all non-trivial reduced words representing an equation in $\mathfrak{I}_g$. We now apply to this grammar the algorithm of Proposition 5.1.11, and the conclusion follows. □

### 5.2.2 Equations of a certain fixed degree

**Theorem 5.2.3.** *The set $\mathfrak{J}_{g,d}$ is context-free as a subset of $H * \langle x \rangle$.*

*Proof.* By Theorem 5.2.1 we have that the language $\mathcal{L}(\mathfrak{I}_g)$ is context-free. It is easy to construct a finite automaton for the language $\mathcal{R}$ of all cyclically reduced words that contain exactly $d$ occurrences of $x$ and $\overline{x}$. By Proposition 5.1.9 we obtain a context-free language $\mathcal{L}(\mathfrak{I}_g) \cap \mathcal{R}$, whose elements are exactly the cyclically reduced

words representing an equation for $g$ of degree $d$. The conclusion now follows from Proposition 5.1.16. □

**Theorem 5.2.4** ([Asc22a]). *There is an algorithm that, given $H \leq F_n$ finitely generated and $g \in F_n$ and $d \in \mathbb{N}$, tells us whether the set $\mathfrak{J}_{g,d}$ is empty or not and, if such exists, produces an element in $\mathfrak{J}_{g,d}$.*

*Proof.* We proceed as in the proof of Theorem 5.2.2. In the proof of Theorem 5.2.2 we proved that it is possible to algorithmically build a grammar for the language $\mathcal{L}(\mathfrak{I}_g)$. It is quite easy to produce an algorithm that constructs a finite automaton for the regular language $\mathcal{R}$ of all cyclically reduced words that contain exactly $d$ occurrences of $x$ and $\overline{x}$. By Proposition 5.1.9 we obtain an algorithm to construct a grammar for the language $\mathcal{L}(\mathfrak{I}_g) \cap \mathcal{R}$. We apply the algorithm of Proposition 5.1.11 to this grammar and the conclusion follows. □

## 5.3 Asymptotic growth rate of the number of equations

The aim of this section is to study the growth rate of the number of equations up to a given length. We deal with this question both for the set $\mathfrak{I}_g$ of all equations, and for the set $\mathfrak{J}_{g,d}$ of equations of a certain given degree $d$.

**Definition 5.3.1.** *Let $\rho : [0, +\infty) \to [0, +\infty)$ be a non-decreasing function. We say that $\rho$ has*

*(i)* ***exponential growth*** *if $e^{\alpha M} \leq \rho(M) \leq e^{\beta M}$ for some $\alpha, \beta > 0$ and for all $M$ big enough.*

*(ii)* ***polynomial growth of degree*** *$k$, where $k \geq 0$ is an integer, if $\alpha M^k \leq \rho(M) \leq \beta M^k$ for some $\alpha, \beta > 0$ and for all $M$ big enough.*

Polynomial growth of degree 1 is usually called **linear growth**.

### 5.3.1 Growth rate of context-free languages

Let $\mathcal{A}$ be an alphabet and let $\mathcal{L} \subseteq \mathcal{A}^*$ be a language. The growth function is defined by

$$\rho_{\mathcal{L}}(M) = |\{w \in \mathcal{L} : l(w) \leq M\}|$$

We are interested in the growth of the function $\rho_{\mathcal{L}}$ when $\mathcal{L}$ is a context-free language. This has already been studied, for example in [BG02] and in [Inc01], and the possibilities are described in full in [GKRS08]; we restate here their result.

**Theorem 5.3.2** ([GKRS08]). *Let $\mathcal{L}$ be a context-free language. Then exactly one of the following takes place:*

*(i) The function $\rho_{\mathcal{L}}$ has polynomial growth of degree $k$ for some $k \in \mathbb{N}$.*

*(ii) The function $\rho_{\mathcal{L}}$ has exponential growth.*

*Moreover, there is an algorithm that, given a context-free grammar for $\mathcal{L}$, tells us (in polynomial time) which of the above cases takes place, and, in case (i), also tells us the degree $k$ of the growth.*

Let now $F_n$ be a free group on $n$ generators $a_1, ..., a_n$ and consider as usual the alphabet $\mathcal{A} = \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$. If $\mathcal{L}(1)$ is the language of all words representing the identity element, then the function $\rho_{\mathcal{L}(1)}$ has exponential growth. In particular, for every non-empty subset $S \subseteq F_n$ we have that $\rho_{\mathcal{L}(S)}$ has exponential growth. Essentially, here the exponential growth is given by the fact that each element of $F_n$ can be represented by a lot of words; in order to eliminate ambiguity, in what follows we will consider only cyclically reduced words.

**Definition 5.3.3.** *Let $S \subseteq F_n$ be a subset which is invariant under conjugation. Define $\mathcal{L}_{\mathrm{cr}}(S) \subseteq \mathcal{A}^*$ to be the language of all cyclically reduced words that represent an element of $S$.*

### 5.3.2 Growth rate of sets of equations

Let $H \leq F_n$ be a finitely generated subgroup with basis $h_1, ..., h_r$ and let $g \in F_n$ be an element that depends on $H$. We work over the alphabet $\mathcal{B} = \{h_1, \overline{h}_1, ..., h_n, \overline{h}_n, x, \overline{x}\}$. We want to study the growth of the function $\rho_g(M) := \rho_{\mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_g)}(M)$ which counts the number of cyclically reduced equations up to a certain length, and of the functions $\rho_{g,d}(M) := \rho_{\mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_{g,d})}(M)$ which count the number of cyclically reduced equations of degree $d$ up to a certain length.

**Theorem 5.3.4.** *The function $\rho_g(M)$ has exponential growth.*

*Proof.* The language $\mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_g)$ is context-free, so by Theorem 5.3.2 it has either polynomial growth (of some degree $k \in \mathbb{N}$) or exponential growth. To prove the theorem, it is enough to find an exponential lower bound.

Assume that $H$ has rank at least 2. Let $w \in \mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_g)$ and let $t_1, t_2, t_3, t_4 \in \mathcal{B}$ be the first and the last letter of $w$ and their inverses. Choose a reduced word $h \in \mathcal{B}^*$ which doesn't begin or end with any of $t_1, t_2, t_3, t_4$: then we have a word $hw\overline{h}w \in \mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_g)$. Different choices of $h$ give different elements $hw\overline{h}w$. Since we have exponentially many choices for $h$ of length at most $M$, we obtain an exponential lower bound on $\rho_g$.

Assume now that $H$ has rank 1. This means that $H = \langle h \rangle$ where $h = b^m$ and $g = b^k$ for some $b \in F_n \setminus \{1\}$ and $m \in \mathbb{Z} \setminus \{0\}$ and $k \in \mathbb{Z}$. In this case $\mathcal{B} = \{h, \overline{h}, x, \overline{x}\}$. Fix $M$ and for each $(\alpha_1, ..., \alpha_{2M-1}) \in \{1, 2\}^{2M-1}$ define the word $w_{\alpha_1,...,\alpha_{2M-1}} \in \mathcal{B}^*$ given by

$$w_{\alpha_1,...,\alpha_{2M-1}} = h^{\delta_1\alpha_1} x h^{\delta_2\alpha_2} \overline{x} h^{\delta_3\alpha_3} x h^{\delta_4\alpha_4} \overline{x} \ ... \ h^{\delta_{2M-1}\alpha_{2M-1}} x h^{-(\delta_1\alpha_1+...+\delta_{2M-1}\alpha_{2M-1})} \overline{x}$$

where $\delta_1 = 1$ and $\delta_j \in \{+1, -1\}$ is defined inductively by setting $\delta_j = 1$ if and only if we have $\delta_1\alpha_1 + ... + \delta_{j-1}\alpha_{j-1} \leq 0$. We point out that $h^e$ for $e$ negative integer has to be interpreted as the word $\overline{h}^{-e}$. It is easy to check that $w_{\alpha_1,...,\alpha_{2M-1}}$ belongs to $\mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_g)$, and that different choices of $(\alpha_1, ..., \alpha_{2M-1})$ produce different words. It is easy to show by induction that $|\delta_1\alpha_1 + ... + \delta_j\alpha_j| \leq 2$ for $j = 1, ..., 2M-1$, and thus it follows that $w_{\alpha_1,...,\alpha_{2M-1}}$ has length at most $6M$. Since we have $2^{2M-1}$ choices for $(\alpha_1, ..., \alpha_{2M-1})$, this provides an exponential lower bound for $\rho_g(M)$. □

**Theorem 5.3.5.** *Let $d \in \mathbb{N}$ be a non-negative integer. Then the function $\rho_{g,d}(M)$ has either exponential growth, or polynomial growth of degree $k$ for some $k \in \mathbb{N}$. Moreover, there is an algorithm that tells us which case takes place, and in the second case computes the degree $k$ of the growth.*

*Proof.* As in the proof of Theorem 5.2.4, we can produce a grammar for the context-free language $\mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_{g,d})$. We apply Theorem 5.3.2 to this grammar. The conclusion follows. □

### 5.3.3 Degrees with polynomial growth rate

Let $H \leq F_n$ be a finitely generated subgroup with basis $h_1, ..., h_r$ and let $g \in F_n$ be an element that depends on $H$. We work over the alphabet $\mathcal{B} = \{h_1, \overline{h}_1, ..., h_n, \overline{h}_n, x, \overline{x}\}$.

**Definition 5.3.6.** *Define the following sets:*
*(i)* $D_g = \{d \in \mathbb{N} :$ *there is a non-trivial equation $w \in \mathfrak{J}_g$ of degree $d$*$\}$.
*(ii)* $D_g^{\mathrm{pol},k} = \{d \in D_g : \rho_{g,d}$ *has polynomial growth of degree $k$*$\}$.
*(iii)* $D_g^{\exp} = \{d \in D_g : \rho_{g,d}$ *has exponential growth*$\}$.

According to Theorem 5.3.5, we have a partition $D_g = D_g^{\exp} \sqcup \bigsqcup_{k\in\mathbb{N}} D_g^{\mathrm{pol},k}$.

**Lemma 5.3.7.** *Suppose $H$ has rank at least 2. Then for each $d, d' \in D_g$ and $k \geq 0$ we have $d + d' + 2k \in D_g^{\exp}$.*

*Remark.* Compare this with Lemma 4.3.14.

*Proof.* Let $w \in \mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_{g,d})$ and, up to cyclic permutation, we can assume that $w$ is of the form $c_1x^{e_1}...c_\alpha x^{e_\alpha}$ with $c_1, ..., c_\alpha \in \{h_1, \overline{h}_1, ..., h_k, \overline{h}_k\}^*$ and $e_1, ..., e_\alpha \in \mathbb{Z} \setminus \{0\}$. Similarly let $w' \in \mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_{g,d'})$ where $w' = c'_1x^{e'_1}...c'_\beta x^{e'_\beta}$ with $c'_1, ..., c'_\beta \in \{h_1, \overline{h}_1, ..., h_k, \overline{h}_k\}^*$ and $e'_1, ..., e'_\beta \in \mathbb{Z} \setminus \{0\}$. Without loss of generality we assume $e'_\beta > 0$.
We fix $k \geq 0$ and for each $h \in \{h_1, \overline{h}_1, ..., h_r, \overline{h}_r\}^*$ we consider the word $w_h = \overline{h}wh\overline{x}^k w'x^k \in \mathcal{B}^*$, which represents an equation for $g$. If the $h$ is reduced, and the first letter of $h$ is different from the first of $c_1$, and the last letter of $h$ is not the inverse of the first of $c'_1$, then the word $w_h$ is cyclically reduced, and in particular it

belongs to $\mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_{g,d})$. Since $H$ has rank $r \geq 2$, there are exponentially many choices for such an $h$ of a given length and with those properties. This proves the desired result. □

**Lemma 5.3.8.** *Suppose $H$ has rank at least 2. Then $D_g \setminus D_g^{\exp}$ is finite.*

*Remark.* By Theorem 4.3.15, we have that $D_g$ coincides, up to a finite set, with either the set of natural numbers or the set of non-negative even numbers.

*Proof.* If $D_g$ only contains even numbers, then take $d \in D_g$. By Lemma 5.3.7, every even number $\geq 2d$ belongs to $D_g^{\exp}$, and we are done. Suppose now $D_g$ contains an odd number $d \in D_g$: by Lemma 5.3.7 we have that $D_g^{\exp}$ contains all even numbers $\geq 2d$ and all odd numbers $\geq 3d$, and we are done. □

**Theorem 5.3.9** ([Asc22a])**.** *Suppose $H$ has rank at least 2. Then there is an algorithm that computes the finite sets $D_g^{\mathrm{pol},k}$ for $k \in \mathbb{N}$; all but finitely many of these are empty.*

*Proof.* Using Theorem 1.3.4, we can produce algorithmically a finite set of normal generators $\mathfrak{J}_g = \langle\langle w_1, ..., w_k \rangle\rangle$ for $\mathfrak{J}_g \trianglelefteq H * \langle x \rangle$.

Suppose $w_1, ..., w_k$ all have even degree: then $D_g$ only contains even numbers, according to Lemma 4.3.16. We take $d_1$ to be the degree of $w_1$, and we apply Theorem 5.3.5 to check the type of growth of $\rho_{g,d}$ for all $d < 2d_1$; as in Lemma 5.3.8 we have that every even degree $\geq 2d_1$ belongs to $D_g^{\exp}$, and so we are done.

Suppose one of $w_1, ..., w_k$ has odd degree: let's say $w_1$ has degree $d_1$ odd. We apply Theorem 5.3.5 to check the type of growth of $\rho_{g,d}$ for all $d < 3d_1$; as in Lemma 5.3.8 we have that each degree $\geq 3d_1$ belongs to $D_g^{\exp}$, and so we are done. □

## 5.4 Running time of the algorithms

In this section we are interested in bounding the running time of the algorithms described in this chapter. We assume that we are working on a RAM machine; we assume that the machine can perform in time $O(1)$ the most basic operations, including sum and multiplication of two integer numbers. Some of the algorithms have context-free grammars as input or output, and thus we need a way to quantify the size of a context-free grammar; we express the complexity of a context-free grammar in terms of the following parameters:

**Definition 5.4.1.** *Let $(\mathcal{N}, \mathcal{P}, S)$ be a context-free grammar.*
*(i) We denote with $\|\mathcal{P}\| = \sum_{(N,u)\in\mathcal{P}}(1 + l(u))$ the **total size** of the grammar.*
*(ii) We denote with $\mathrm{ram}(\mathcal{P})$ the maximum number of non-terminal symbols in $u$ for $(N, u) \in \mathcal{P}$.*

*Remark.* Here the notation $\mathrm{ram}(\mathcal{P})$ is for "ramification" of $\mathcal{P}$.

### 5.4.1 Checking whether a context-free language is empty or not

We now provide a classical algorithm to check whether a context-free grammar produces the empty language or not, taken from [HMU06]; more generally, we provide algorithms to produce words in the language generated by a given context-free grammar.

The following Proposition 5.4.2 below is a restatement of Proposition 5.1.11, but with an additional bound on the running time of the algorithm. The algorithm described in the proof of Proposition 5.4.2 below is already described in Section 7.4.3 of [HMU06]; however, we need to make the proof more explicit in order to keep track of the complexity of the variants we need.

**Proposition 5.4.2.** *There is an algorithm that, given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$, tells us whether the corresponding context-free language is empty of not. The algorithm runs in time $O(\|\mathcal{P}\|)$.*

*Description of the algorithm.* We assume that the non-terminal symbols are numbered from 0 to $|\mathcal{N}| - 1$, with the initial symbol $S$ being indexed as 0, and that the production rules are numbered from 0 to $|\mathcal{P}| - 1$. We initialize the following data structures:

(i) An array `occurrences` of length $|\mathcal{N}|$. For $i = 0, ..., |\mathcal{N}| - 1$, the entry `occurrences`$[i]$ is the list of the occurrences of the $i$-th non-terminal symbol in the production rules: the list `occurrences`$[i]$ contains a couple of integers $(c, d)$ if and only if the $c$-th production rule is $N \to u$ and the $d$-th symbol of $u$ is the $i$-th non-terminal symbol. We read the list of all production rules once, and whenever we find a non-terminal symbol, we add the corresponding couple to the suitable entry of `occurrences`.

(ii) An array `terminates` of length $|\mathcal{N}|$, each entry containing either `true` or `false`. We initialize `terminates`$[i]$ = `false` for $i = 0, ..., |\mathcal{N}| - 1$.

(iii) An array `remaining` of length $|\mathcal{P}|$. For $j = 0, ..., |\mathcal{P}| - 1$, we consider the $j$-th production rule $N \to u$, and we initialize `remaining`$[j]$ to be the number of non-terminal symbols in $u$, counted with repetition.

(iv) A queue `queue` that contains the indices of some production rules. We use a first-in first-out queue (but it doesn't really matter). We initialize `queue` by reading the array `remaining` once, and whenever we find `remaining`$[j] = 0$ we add $j$ to `queue`.

We now describe the `Update` step. We take (and remove) an index $j$ from `queue`, and we consider the $j$-th production rule $(N, u)$, where $N$ is the $i$-th non-terminal symbol. If `terminates`$[i]$ is `true` we immediately terminate the `Update` step; otherwise we set `terminates`$[i]$ from `false` to `true` and we proceed. We read the list `occurrences`$[i]$: when we read the couple $(c, d)$ we decrease by 1 the number `remaining`$[c]$, and if `remaining`$[c]$ becomes 0, then we add $c$ to `queue`.

The algorithm now runs as follows: it initializes the data structures (i),(ii),(iii),(iv) as explained above, and then it starts running the `Update` step until `queue` becomes empty. At that point the algorithm stops and gives `terminates`$[0]$ as output. □

*Proof that the algorithm works and bounds on the running time.*

(i) The array `occurrences` is initialized at the beginning of the algorithm and then never modified.

(ii) At any point during the algorithm, `terminates`$[i]$ is `true` if and only if we have found out that the $i$-th non-terminal symbol can produce a word in $\mathcal{A}^*$. Symbols that have been inserted in `queue` but have not yet gone through the `Update` step still remain on `false`.

(iii) At any point during the algorithm, if $N \to u$ is the $j$-th production rule, then `remaining`$[j]$ is the number of non-terminal symbols in $u$ such that we don't know whether they can produce a word in $\mathcal{A}^*$ or not. In fact, at the beginning of the algorithm we have that the $i$-th symbol satisfies `terminates`$[i]$ = `false` for $i = 0, ..., |\mathcal{N}| - 1$, and thus we initialize `remaining` by counting all the non-terminal symbols. At each `Update` step, we update `terminates`$[i]$ = `true` exactly at the same time when we update `remaining`; we decrease the entries of `remaining` according to the occurrences of the $i$-th non-terminal symbol, which we read from `occurrences`.

(iv) The queue `queue` contains the indices of the production rules that have been discovered to produce a word in $\mathcal{A}^*$; such a production $N \to u$ tells us that the symbol $N$ can produce a word in $\mathcal{A}^*$, and thus we have to update the other data structures. During the initialization, we add to `queue` exactly the non-terminal symbols $N$ which have some production rule $N \to u$ with $u \in \mathcal{A}^*$, since this proves that each of them can produce a word in $\mathcal{A}^*$. During the update process, whenever the $j$-th production rule $N \to u$ satisfies `remaining`$[j] = 0$, this means that all the non-terminal symbols in $u$ have been proved to produce some word in $\mathcal{A}^*$, and thus we have to add $j$ to `queue`. Notice that each production rule is added to `queue` at most once, when the corresponding entry of `remaining` becomes 0.

The `Update` step consists of taking a production rule $N \to u$ from `queue` and updating the other data structures adding the information that $N$ can produce a word in $\mathcal{A}^*$. If $N$ had already been found out to produce a word in $\mathcal{A}^*$, then we just skip to updating the next production rule in `queue`; otherwise we modify the value of `terminates` accordingly, and then we read from `occurrences` the occurrences of $N$ in the production rules, and we update the array `remaining`, taking into account that $N$ no longer counts as non-terminal, as it can be substituted with a word in $\mathcal{A}^*$. It is possible that some entries of `remaining` decrease to 0 in the update process, meaning that other production rules have to be added to `queue`.

This shows that, at the end of the algorithm, if `terminates`$[i]$ = `true` then the $i$-th non-terminal symbol can produce a word in $\mathcal{A}^*$. We want to prove that the converse implication holds. Suppose by contradiction that the algorithm ended, and the $i$-th

non-terminal symbol $N$ can produce a word in $\mathcal{A}^*$ but still has `terminates`$[i] =$ `false`. We take a derivation $N \to v_1 \to ... \to v_r$ with $v_r \in \mathcal{A}^*$ and without loss of generality we can assume that both $N$ and the derivation have been chosen with $r$ minimum possible. The minimality of $r$ implies that every non-terminal symbol in $v_1$ has the corresponding entry of `terminates` equal to `true`. The discussion above implies that, at the end of the algorithm, the production rule $(N, v_1)$ has the corresponding entry of `remaining` equal to 0. But this means that the production rule had been added to `queue`, and thus that `terminates`$[i]$ had been updated to `true`, contradiction.

Thus at the end of the algorithm we have that the array `terminates` tells us, for each symbol, whether they can produce a word in $\mathcal{A}^*$ or not. In particular, as the initial symbol $S$ is indexed as 0, we get the correct output `terminates`$[0]$.

Let's now discuss the complexity of the algorithm. The initialization of `occurrences`, `terminates`, `remaining`, `queue` are done in time $O(\|\mathcal{P}\|)$, $O(|\mathcal{N}|)$, $O(\|\mathcal{P}\|)$, $O(|\mathcal{P}|)$ respectively. During all the update cycles, the operations of taking one production rule from `queue`, checking and updating the value of `terminates`, and adding a new production rule to `queue`, are done at most once for each production rule, so the total time required is $O(|\mathcal{P}|)$; the operation of updating `remaining` due to a given occurrence $(c, d)$ of a certain non-terminal symbol is done at most once for each symbol in each of the words $u$ for $(N \to u) \in \mathcal{P}$, so the total time required is $O(\|\mathcal{P}\|)$. □

When the algorithm of Proposition 5.4.2 gives an affirmative answer, we would like to algorithmically and efficiently build a word belonging to the language; however, we point out that the length of the shortest word in the language can be exponential in $\|\mathcal{P}\|$, and thus any algorithm that explicitly writes down such a word runs in exponential time. One possible way of going around this problem is to give as output, instead of the full explicit word belonging to the language, some other piece of information, that allows us to build such word in a straightforward manner, while at the same time being shorter.

**Proposition 5.4.3.** *There is an algorithm that, given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ with non-empty language $\mathcal{L}$, produces a list of production rules $(N_1 \to u_1), ..., (N_m \to u_m) \in \mathcal{P}$ such that:*

*(i) The symbols $N_1, ..., N_m$ are pairwise distinct and $N_m = S$ is the initial symbol.*

*(ii) The word $u_i$ contains only terminal symbols and possibly the symbols $N_1, ..., N_{i-1}$.*

*In particular, starting at $S$ and applying the production rules in the list (in any order) produces a word in the language $\mathcal{L}$. The algorithm runs in time $O(\|\mathcal{P}\|)$.*

*Proof.* We run the same algorithm as in the proof of Proposition 5.4.2 but with a few changes.

(i) We initialize an array `list`, which at the beginning is empty, but that will contain the list of (the indices of) the production rules that is required as output. During the `Update` step, if we have taken the index $j$ from `queue` and if we update `terminates`$[i]$ from `false` to `true`, then at the same time we also add $j$ to the end of `list`.

(ii) At the end of the algorithm we give as output the array `list`, truncated at the (unique) occurrence of a production rule for the symbol $S$. □

We are now interested in producing a word in the language that has minimum **size**. In order to keep flexible our notion of **size**, we assume we are given a function $\sigma : \mathcal{A} \to \mathbb{N}$ that assigns to each letter $a$ in our alphabet a non-negative integer number $\sigma(a)$; for a word $w = w_1...w_{l(w)} \in \mathcal{A}^*$ we define $\sigma(w) := \sum_{i=1}^{l(w)} \sigma(w_i)$.

**Definition 5.4.4.** *Let $(\mathcal{N}, \mathcal{P}, S)$ be a context-free grammar.*

*(i) For a non-terminal symbol $N \in \mathcal{N}$ define $\sigma(N) := \min\{\sigma(w) : w \in \mathcal{A}^*$ with $N \stackrel{*}{\to} w\}$.*

*(ii) For a word $u = u_1...u_{l(u)} \in (\mathcal{A} \cup \mathcal{N})^*$ define $\sigma(u) := \sum_{i=1}^{l(u)} \sigma(u_i)$.*

**Proposition 5.4.5.** *There is an algorithm that, given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ with non-empty language $\mathcal{L}$, and a function $\sigma : \mathcal{A} \to \mathbb{N}$, produces a list of couples $(N_1 \to u_1, \tau_1), ..., (N_m \to u_m, \tau_m) \in \mathcal{P} \times \mathbb{N}$ such that:*

*(i) The symbols $N_1, ..., N_m$ are pairwise distinct and $N_m = S$ is the initial symbol.*

*(ii) We have $\sigma(N_i) = \tau_i$.*

*(iii) The word $u_i$ contains only terminal symbols and possibly the symbols $N_1, ..., N_{i-1}$.*

*(iv) The sum of $\sigma(c)$, for each symbol $c$ in $u_i$, is equal to $\tau_i$.*

*In particular $\tau_m = \sigma(S)$ is the minimum possible size for a word in the language. Moreover, starting at $S$ and applying the production rules in the list (in any order) produces a word of minimum possible size in the language $\mathcal{L}$. The algorithm runs in time $O(\|\mathcal{P}\| \log |\mathcal{P}|)$.*

*Proof.* We run the same algorithm as in the proof of Proposition 5.4.2 but with a few changes.

(i) The queue `queue` contains couples $(j, \tau)$ where $j$ is the index of a production rule and $\tau \in \mathbb{N}$. Instead of a first-in first-out queue, this time the elements of `queue` are ordered in increasing order on $\tau$. Whenever we insert a new couple to `queue`, we make sure to preserve the ordering; whenever we extract an element from `queue`, we take one with $\tau$ smallest.

(ii) We initialize an array `list`, which at the beginning is empty, but that will contain the list that is required as output. During the `Update` step, if we have taken the couple $(j, \tau)$ from `queue` and we update `terminates`$[i]$ from `false` to `true`, then at the same time we also add the couple $(j, \tau)$ to `list`.

(iii) We initialize an array `size` of length $|\mathcal{P}|$: if the $j$-th production rule is $N \to u$ then we initialize `size`$[j]$ to be the sum of $\omega(c)$ for each terminal symbol $c$ in $u$

(with repetition). During the `Update` step, if we have taken the couple $(j, \tau)$ from `queue`, whenever we decrease `remaining`$[c]$ by one we also increase `size`$[c]$ by $\tau$. If `remaining`$[c]$ goes down to zero, then we add the couple $(c,$ `size`$[c])$ to `queue`.

(iv) At the end of the algorithm we give as output the array `list`, possibly truncated at the (unique) occurrence of a production rule for the symbol $S$.

Each production rule is added at most once to `queue`, and the steps of adding a couple to `queue` is done in time $O(\log |\mathcal{P}|)$. The rest of the operations of the algorithms are performed in total time $O(\|\mathcal{P}\|)$. Thus the total running time of the algorithm is $O(\|\mathcal{P}\| \log |\mathcal{P}|)$. □

If, for any reason, we have information that the language contains at least one word of bounded length, then there is an algorithm to explicitly write down such a word.

**Proposition 5.4.6.** *There is an algorithm that, given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ and an integer $r \geq 0$, produces the following:*

*(i) A list of all non-terminal symbols $N \in \mathcal{N}$ that can derive a word of length at most $r$.*

*(ii) For each $N$ in the list (i), the minimum possible length $l_N$ and a word $w_N \in \mathcal{A}^*$ of length $l_N$ such that $w_N$ can be produced by means of a derivation starting at $N$.*

*The algorithm produces the list (i) and the lengths $l_N$ in time $O(\|\mathcal{P}\|r)$ and the words $w_N$ in time $O(\|\mathcal{P}\|r^2)$.*

*Proof.* The algorithm is again similar to the one in the proof of Proposition 5.4.2, but with a few differences.

(i) We initialize `occurrences` exactly as in Proposition 5.4.2.

(ii) We create three arrays `terminates` and `length` and `word` of length $|\mathcal{N}|$. For $i = 0, ..., |\mathcal{N}| - 1$ we initialize `terminates`$[i]$ = `false` and `length`$[i] = 0$ and `word`$[i]$ to be the empty string.

(iii) We create two arrays `remaining` and `size` of length $|\mathcal{P}|$. For $j = 0, ..., |\mathcal{P}| - 1$, if $(N, u)$ is the $j - th$ production rule, then we initialize `remaining`$[j]$ to be the number of non-terminal symbols in $u$ and `size`$[j]$ to be the number of terminal symbols in $u$.

(iv) We create a queue `queue` that will contain some indices of some production rules. We initialize `queue` to be empty.

(v) We create an integer $s$ and we initialize $s = 0$.

We now describe the `Update` step. We take (and remove) an index $j$ from `queue`, and we consider the $j$-th production rule $(N, u)$, where $N$ is the $i$-th non-terminal symbol. If `terminates`$[i]$ is `true` we immediately terminate the `Update` step, otherwise we proceed. We set `terminates`$[i]$ = `true`, `length`$[i] = s$ and `word`$[i]$ to be the word obtained from $u$ by substituting each occurrence of each non-terminal symbol with the corresponding entry of `word` (i.e. we read $u$ and whenever we encounter an occurrence

of the $i'$-th non-terminal symbol we substitute that occurrence with `word`$[i']$). We read the list `occurrences`$[i]$: when we read the couple $(c, d)$, we decrease by 1 the number `remaining`$[c]$, we increase by `length`$[i]$ the number `size`$[c]$, and if `remaining`$[c]$ is 0 and `size`$[j]$ is $s$ then we add $c$ to `queue`.

The algorithm now runs as follows: it initializes the data structures (i), (ii), (iii), (iv), (v) as described above. Then for each $s = 0, ..., r$ it runs the following: it initializes `queue` by adding all the indices $j$ such that `remaining`$[j] = 0$ and `size`$[j] = s$, and then it runs the `Update` step until `queue` becomes empty. At the end of this process, the algorithm terminates and outputs the list of all non-terminal symbols such that `terminates`$[i]$ = `true`, and for each of them the integer `length`$[i]$ and the string `word`$[i]$.

At any point during the algorithm, if `terminates`$[i]$ is `true` then `length`$[i]$ and `word`$[i]$ contain the minimum length of a word and a word of that length that can be produced from the $i$-th non-terminal symbol. At any point during the algorithm, if the $j$-th production rule is $(N, u)$, then `remaining`$[j]$ is the number of non-terminal symbols that are contained in $u$ and still have `terminates` equal to `false`; similarly, `size`$[j]$ is the number of terminal symbols in the word $u$ to which we sum also, for each occurrence of a non-terminal symbol with `terminates` equal to `true`, the length `length` of the shortest word that we can produce from that symbol. The idea is that during the iteration with a certain value of $s$, we find all the non-terminal symbols that can produce a word of length at most $s$; whenever we increase $s$ to $s + 1$ we reset `queue` by rechecking all the production rules with `remaining` equal to 0, because it is possible that one of these production rules has `remaining` equal to 0 but `size` equal to $s + 1$, so that it has been ignored up to this point. The proof that the algorithm works and the computation of the complexity are completely analogous to the proof of Proposition 5.4.2. □

### 5.4.2 Unambiguous grammars

We are interested in context-free grammars with the additional property of being unambiguous. This means that each word in the corresponding language can be obtained in an essentially unique way; this is made precise as follows.

**Definition 5.4.7.** *Let $(\mathcal{N}, \mathcal{P}, S)$ be a context-free grammar. We say that a derivation*

$$S \to_{k_0,(S,u_0)} v_1 \to_{k_1,(N_1,u_1)} v_2 \to ... \to_{k_{r-1},(N_{r-1},u_{r-1})} v_r$$

*is **leftmost** if each substitution $v_i \to_{k_i,(N_i,u_i)} v_{i+1}$ acts on the leftmost non-terminal symbol of $v_i$.*

**Definition 5.4.8.** *A context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ is called **unambiguous** if every word in the corresponding context-free language can be obtained by means of a unique leftmost derivation.*

We point out that, even if two grammars produce the same language, it is possible that one is ambiguous while the other is not. There are also languages which are context-free, but which don't have any unambiguous grammar: such languages are called *inherently ambiguous*.

The reason why we are interested in unambiguous grammars is that it helps controlling the growth of the corresponding language; this will be explained more in detail in Section 5.4.8.

*Remark.* We point out that the notion of unambiguous grammar is related to a special kind of push-down automaton, the *deterministic push-down automata*. To be precise, if a language is recognized by a deterministic push-down automaton, then it admits an unambiguous grammar (but the converse is false in general). In Propositions 5.4.13 and 5.4.18 we will provide two grammars which are unambiguous: the reason behind this is that the corresponding language can be recognized by means of a deterministic push-down automaton.

### 5.4.3 The intersection of a context-free language and a regular language

Let $\mathcal{A}$ be an alphabet. Proposition 5.1.9 tells us that the intersection of a context-free language $\mathcal{L}$ and a regular language $\mathcal{R}$ is a context-free language $\mathcal{L} \cap \mathcal{R}$. We provide here an explicit description of a grammar for $\mathcal{L} \cap \mathcal{R}$ in terms of a grammar for $\mathcal{L}$ and of an automaton for $\mathcal{R}$; we do so in order to provide bounds on the size of this grammar, as well as on the time required to algorithmically produce it; we also give particular attention to the extra property of being unambiguous.

**Proposition 5.4.9.** *There is an algorithm that, given a context-free grammar $(\mathcal{N}, \mathcal{P}, S)$ for $\mathcal{L}$ and a finite automaton $(Q, \delta, Q_0, Q_f)$ for $\mathcal{R}$, produces a context-free grammar $(\mathcal{N}', \mathcal{P}', S')$ for $\mathcal{L} \cap \mathcal{R}$. Moreover the algorithm satisfies the following:*
*(i) $\|\mathcal{P}'\|$ is $O(\|\mathcal{P}\||Q|^{2+2\mathrm{ram}(\mathcal{P})})$ and $\mathrm{ram}(\mathcal{P}') = \mathrm{ram}(\mathcal{P})$.*
*(ii) The algorithm runs in time $O(\|\mathcal{P}\||Q|^{4+2\mathrm{ram}(\mathcal{P})})$.*
*(iii) If the $(Q, \delta, Q_0, Q_f)$ is deterministic, the algorithm runs in time $O(\|\mathcal{P}\||Q|^{2+2\mathrm{ram}(\mathcal{P})})$.*
*(iv) If $(\mathcal{N}, \mathcal{P}, S)$ is unambiguous and $(Q, \delta, Q_0, Q_f)$ is deterministic, then $(\mathcal{N}', \mathcal{P}', S')$ is unambiguous.*

The grammar has non-terminal symbols $N_{p \to q}$ for $N \in \mathcal{N}$ and $p, q \in Q$. The grammar has an extra symbol $S'$ which we set as initial symbol.

Suppose we are given a production rule $(N \to w_0 A_1 w_1 A_2 ... A_k w_k) \in \mathcal{P}$ with $w_0, ..., w_k \in \mathcal{A}^*$ and $A_1, ..., A_k \in \mathcal{N}$; suppose we are given $p, q, p_1, q_1, ..., p_k, q_k \in Q$; suppose $\delta(p, w_0) \ni p_1$ and $\delta(q_i, w_i) \ni p_{i+1}$ for $i = 1, ..., k-1$ and $\delta(q_k, w_k) \ni q$. Then we define a production rule in $\mathcal{P}'$

$$N_{p \to q} \ \to \ w_0 (A_1)_{p_1 \to q_1} w_1 ... (A_k)_{p_k \to q_k} w_k$$

For each $q_0 \in Q_0$ and $q_f \in Q_f$ we define a production rule in $\mathcal{P}'$

$$S' \ \rightarrow \ S_{q_0 \rightarrow q_f}$$

The rest of this section is dedicated to the proof that this grammar has the desired properties.

**Lemma 5.4.10.** *Let $(Q, \delta, Q_0, Q_f)$ be a finite automaton. Let $w = w'w''$ with $w', w'' \in \mathcal{A}^*$ be a word and let $p, q \in Q$ be such that $\delta(p, w) \ni q$. Then there is a state $r \in Q$ such that $\delta(p, w') \ni r$ and $\delta(r, w'') \ni q$.*

*Proof.* Immediate from the definitions. □

**Lemma 5.4.11.** *For $w \in \mathcal{A}^*$ we have that $N_{p \rightarrow q} \stackrel{*}{\rightarrow} w$ in the grammar $\mathcal{P}'$ if and only if the following two conditions hold:*
*(i) $N \stackrel{*}{\rightarrow} w$ in the grammar $\mathcal{P}$.*
*(ii) $\delta(p, w) \ni q$ in the automaton.*

*Proof.* We prove that if $N \stackrel{*}{\rightarrow} w$ and $\delta(p, w) \ni q$ then $N_{p \rightarrow q} \stackrel{*}{\rightarrow} w$, by induction on the length of the derivation $N \stackrel{*}{\rightarrow} w$. The base step, when the derivation has length one, is trivial. For the inductive step, suppose $N \rightarrow w_0 A_1 w_1 A_2 ... A_k w_k \stackrel{*}{\rightarrow} w_0 v_1 w_1 v_2 ... v_k w_k = w$, meaning that $A_i \stackrel{*}{\rightarrow} v_i$ for $i = 1, ..., k$, and suppose $\delta(p, w) = q$. Using repeatedly Lemma 5.4.10 we find states $p_1, q_1, ..., p_k, q_k \in Q$ such that $\delta(p, w_0) \ni p_1$, $\delta(p_i, v_i) \ni q_i$ for $i = 1, ..., k$, $\delta(q_i, w_i) \ni p_{i+1}$ for $i = 1, ..., k-1$ and $\delta(q_k, w_k) \ni q$. By definition of the grammar $\mathcal{P}'$ we have $N_{p \rightarrow q} \rightarrow w_0 (A_1)_{p_1 \rightarrow q_1} w_1 (A_2)_{p_2 \rightarrow q_2} ... (A_k)_{p_k}$ and by inductive hypothesis we have $(A_i)_{p_i \rightarrow q_i} \stackrel{*}{\rightarrow} v_i$. It follows that $N_{p \rightarrow q} \stackrel{*}{\rightarrow} w$ as desired.

We prove that if $N_{p \rightarrow q} \stackrel{*}{\rightarrow} w$ then $N \stackrel{*}{\rightarrow} w$ and $\delta(p, w) \ni q$, by induction on the length of the derivation $N_{p \rightarrow q} \stackrel{*}{\rightarrow} w$. The base step, when the derivation has length one, is trivial. For the inductive step, suppose $N_{p \rightarrow q} \rightarrow w_0 (A_1)_{p_1 \rightarrow q_1} w_1 (A_2)_{p_2 \rightarrow q_2} ... (A_k)_{p_k \rightarrow q_k} w_k \stackrel{*}{\rightarrow} w_0 v_1 w_1 v_2 ... v_k w_k = w$, meaning that $(A_i)_{p_i \rightarrow q_i} \stackrel{*}{\rightarrow} v_i$ for $i = 1, ..., k$. By definition of the grammar $\mathcal{P}'$ we have $N \rightarrow w_0 A_1 w_1 A_2 ... A_k w_k$ and by inductive hypothesis we have $A_i \stackrel{*}{\rightarrow} v_i$ for $i = 1, ..., k$, yielding that $N \stackrel{*}{\rightarrow} w$. By definition of the grammar $\mathcal{P}'$ we have that $\delta(p, w_0) \ni p_1$ and $\delta(q_i, w_i) \ni p_{i+1}$ for $i = 1, ..., k-1$ and $\delta(q_k, w_k) \ni q$. By inductive hypothesis we have that $\delta(p_i, v_i) \ni q_i$ for $i = 1, ..., k$. It follows that $\delta(p, w) = \delta(p, w_0 v_1 w_1 v_2 ... v_k w_k) \supseteq \delta(p_1, v_1 w_1 v_2 ... v_k w_k) \supseteq \delta(q_1, w_1 v_2 ... v_k w_k) \supseteq ... \supseteq \delta(q_k, w_k) \ni q$, and the conclusion follows. □

It immediately follows that the context-free grammar $(\mathcal{N}', \mathcal{P}', S')$ produces the language $\mathcal{L} \cap \mathcal{R}$. Each production rule in $\mathcal{P}$ gives us at most $|Q|^{2+2\mathrm{ram}(\mathcal{P})}$ production rules in $\mathcal{P}'$, each of the same length and with the same number of non-terminal symbols; this easily proves that $\|\mathcal{P}'\|$ is $O(\|\mathcal{P}\||Q|^{2+2\mathrm{ram}(\mathcal{P})})$ and $\mathrm{ram}(\mathcal{P}') = \mathrm{ram}(\mathcal{P})$.

In order to estimate the complexity of explicitly performing the above construction, we first need to estimate the complexity of the membership problem, for a word of

length $l$, to the language generated by the automaton $(Q, \delta, Q_0, Q_f)$. According to Section 4.3.3 of [HMU06], the membership problem can be solved in time $O(|Q|^2 l)$. Now, for every production rule $(N, u) = (N, w_0 A_1 w_1 A_2 ... A_k w_k) \in \mathcal{P}$ and for every $p, q, p_1, q_1, ..., p_k, q_k \in Q$, we have to test the (equivalent of) membership problem for the words $w_0, ..., w_k$: this can be done in time $O(|Q|^2(l(w_0) + ... + l(w_k)))$ which is at most $O(|Q|^2(1 + l(u)))$; repeating for all choices of $p, q, p_1, q_1, ..., p_k, q_k \in Q$ can be done in time $O(|Q|^{4+2k}(1 + l(u)))$; repeating for all production rules $(N, u) \in \mathcal{P}$ can be done in time $O(|Q|^{4+2\mathrm{ram}(\mathcal{P})}\|\mathcal{P}\|)$. In the case the $(Q, \delta, Q_0, Q_f)$ is deterministic, according to Section 4.3.3 of [HMU06], the membership problem for a word of length $l$ can be solved in time $O(l)$ instead of $O(|Q|^2 l)$. The same computation as above gives complexity $O(|Q|^{2+2\mathrm{ram}(\mathcal{P})}\|\mathcal{P}\|)$.

**Lemma 5.4.12.** *If $(\mathcal{N}, \mathcal{P}, S)$ is unambiguous and $(Q, \delta, Q_0, Q_f)$ is deterministic, then $(\mathcal{N}', \mathcal{P}', S')$ is unambiguous.*

*Proof.* Suppose we have a leftmost derivation $N_{p\to q} \to ... \to w$ in $(\mathcal{N}', \mathcal{P}', S')$. Removing the labels gives a leftmost derivation $N \to ... \to w$ in $(\mathcal{N}, \mathcal{P}, S)$, and thus the productions of this derivation are uniquely determined. Since the automaton is deterministic, the labels $p_i, q_i$ to be added in each of these productions are uniquely determined too. Thus the leftmost derivation $N_{p\to q} \to ... \to w$ is uniquely determined too. This shows that $(\mathcal{N}', \mathcal{P}', S')$ is unambiguous. □

The proof of Proposition 5.4.9 is thus complete.

### 5.4.4 A grammar for the language of the trivial element

The result of this section is a particular case of the result of next Section 5.4.5; we think it's useful to examine the proof in this easier case first.

Let $F_n$ be a free group generated by $a_1, ..., a_n$ and consider the alphabet $\mathcal{A} = \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$. Let $\mathcal{L}(1) \subseteq \mathcal{A}^*$ be the language of all words representing the trivial element $1 \in F_n$.

**Proposition 5.4.13.** *The language $\mathcal{L}(1)$ is context-free and has a grammar $(\mathcal{N}, \mathcal{P}, S)$ such that:*

*(i) $\|\mathcal{P}\|$ is $O(n)$ and* $\mathrm{ram}(\mathcal{P}) = 2$.

*(ii) $(\mathcal{N}, \mathcal{P}, S)$ is unambiguous.*

The grammar has non-terminal symbols $S, A_i, \overline{A}_i$ for $i = 1, ..., n$. The initial symbol is $S$ and the production rules are as follows:

$$
\begin{array}{lcllllll}
S & \to & \epsilon & | & a_j\overline{A}_j S & | & \overline{a}_j A_j S & & \text{for } j = 1, ..., n. \\
A_i & \to & a_i & | & \overline{a}_i A_i A_i & | & a_j \overline{A}_j A_i & | \quad \overline{a}_j A_j A_i & \text{for } j = 1, ..., n \text{ with } j \neq i. \\
\overline{A}_i & \to & \overline{a}_i & | & a_i \overline{A}_i \overline{A}_i & | & a_j \overline{A}_j \overline{A}_i & | \quad \overline{a}_j A_j \overline{A}_i & \text{for } j = 1, ..., n \text{ with } j \neq i.
\end{array}
$$

The rest of this section is dedicated to the proof that this grammar has the desired properties. When we say that $w'$ is a **proper** initial segment of $w$ we mean $w' \neq \epsilon$ and $w' \neq w$. The following is a standard lemma about cancellations in words, and the proof will be omitted.

**Lemma 5.4.14.** *Let $w, w_r \in \mathcal{A}^*$ be non-empty words such that $w \equiv w_r$ in $F_n$ and $w_r$ is reduced. If the first letter of $w$ and the first letter of $w_r$ are different, then there is a proper initial segment $w'$ of $w$ such that $w' \equiv 1$ in $F_n$.*

**Lemma 5.4.15.** *For $w \in \mathcal{A}^*$ and $i = 1, ..., n$ we have that $A_i \xrightarrow{*} w$ (resp. $\overline{A}_i \xrightarrow{*} w$) if and only if the following two conditions hold:*

*(i) $w \equiv a_i$ in $F_n$ (resp. $w \equiv \overline{a}_i$).*

*(ii) No proper initial segment $w'$ of $w$ satisfies $w' \equiv a_i$ in $F_n$ (resp. $w' \equiv \overline{a}_i$).*

*Proof.* We prove that if $A_i \xrightarrow{*} w$ then (i) and (ii) hold, by induction on the length of the derivation. The base step, when the derivation has length one, is trivial. Suppose $A_i \xrightarrow{*} w$ and suppose the first production applied is $A_i \to \overline{a}_j A_j A_i$ for some $j \in \{1, ..., n\}$ (the other case is analogous): then we can write $w = \overline{a}_j xy$ with $A_j \xrightarrow{*} x$ and $A_i \xrightarrow{*} y$. By inductive hypothesis $x \equiv a_j$ and $y \equiv a_i$, so that $w \equiv a_i$, yielding (i). Suppose a proper initial segment $w'$ of $w$ satisfies $w' \equiv a_i$: then it is either of the form $w' = \overline{a}_j x'$ where $x'$ is an initial segment of $x$, or of the form $w' = \overline{a}_j xy'$ where $y'$ is a proper initial segment of $y'$. If $w' = \overline{a}_j x'$ then we have $\overline{a}_j x' \equiv a_i$ and thus by Lemma 5.4.14 we find a proper initial segment $x''$ of $x$ such that $\overline{a}_j x'' \equiv 1$, giving $x'' \equiv a_j$ and contradicting the inductive hypothesis. If $w' = \overline{a}_j xy'$ then we have $\overline{a}_j xy' \equiv a_i$ and thus $y' \equiv a_i$, contradicting again the inductive hypothesis. This yields (ii).

We prove that if (i) and (ii) holds for $w$ then $A_i \xrightarrow{*} w$, by induction on the length of $w$. The base step, when $w$ has length one, is trivial. If $w$ has length bigger than one, then by (ii) the first letter of $w$ can't be $a_i$; suppose the first letter of $w$ is $\overline{a}_j$ for some $j \in \{1, ..., n\}$ (the other case is analogous). Since $w \equiv a_i$ and the first letter of $w$ is $\overline{a}_j$, by Lemma 5.4.14 we can find a decomposition $w = \overline{a}_j xy$ with $x \equiv a_j$; without loss of generality we can also assume that $x$ has minimum possible length for such a decomposition. We have that $x \equiv a_j$ and no proper initial segment $x'$ of $x$ satisfies $x' \equiv a_j$, and thus by inductive hypothesis we must have $A_j \xrightarrow{*} x$. Since $w \equiv a_i$ and $w = \overline{a}_j xy$ we obtain that $y \equiv a_i$; since no proper initial segment $w'$ of $w$ satisfies $w' \equiv a_i$, it follows that no proper initial segment $y'$ of $y$ satisfies $y' \equiv a_i$; it follows by inductive hypothesis that $A_i \xrightarrow{*} y$. But then we have $A_i \to \overline{a}_j A_j A_i \xrightarrow{*} \overline{a}_j x A_i \xrightarrow{*} \overline{a}_j xy = w$ as desired. □

**Lemma 5.4.16.** *For $w \in \mathcal{A}^*$ we have that $S \xrightarrow{*} w$ if and only if $w \equiv 1$ in $F_n$.*

*Proof.* We prove that if $S \xrightarrow{*} w$ then $w \equiv 1$, by induction on the length of the derivation. The base step, when the derivation has length one, is trivial. Suppose $S \xrightarrow{*} w$ and suppose the first production applied is $S \to \overline{a}_i A_i S$ (the other case is

analogous): then we can write $w = \overline{a}_i xy$ with $A_i \xrightarrow{*} x$ and $S \xrightarrow{*} y$. By Lemma 5.4.15 we have $x \equiv a_i$ and by inductive hypothesis we have $y \equiv 1$, yielding $w \equiv 1$ as desired.

We prove that if $w \equiv 1$ then $S \xrightarrow{*} w$, by induction on the length of the word. The base step, when the word has length zero, is trivial. Suppose $w \equiv 1$ and without loss of generality assume that $w$ begins with $\overline{a}_j$. Decompose $w = \overline{a}_j xy$ where $x$ is the shortest possible such that $x \equiv a_j$. By Lemma 5.4.15 we have $A_j \xrightarrow{*} x$ and by inductive hypothesis we have $S \xrightarrow{*} y$, yielding $S \to \overline{a}_j A_j S \xrightarrow{*} \overline{a}_j xS \xrightarrow{*} \overline{a}_j xy = w$ as desired. □

**Lemma 5.4.17.** *The above grammar is unambiguous.*

*Proof.* Suppose we are given a leftmost derivation $S = v_0 \to v_1 \to ... \to v_{r-1} \to v_r = w$ for a certain word $w \in \mathcal{A}^*$. We fix $k \in \{0, ..., r-1\}$ and we want to prove that the production rule to be applied on $v_k$ is forced. Write $v_k = w'Np$ with $w' \in \mathcal{A}^*$ and $N \in \mathcal{N}$ and $p \in (\mathcal{A} \cup \mathcal{N})^*$, and notice that $w'$ must be an initial segment of $w$. Suppose $N = A_i$ for some $i \in \{1, ..., n\}$ (the case $N = \overline{A}_i$ is analogous). Each production rule for $A_i$ begins with a terminal symbol, so we must have $w' \neq w$: we call $c \in \mathcal{A}$ the next letter of $w$, so that $w'c$ is an initial segment of $w$. But for each $c \in \mathcal{A}^*$ there is exactly one production rule for $A_i$ that begins with $c$, and thus we are forced to apply that production rule.

It is easy to show by induction that each of $v_0, ..., v_{r-1}$ contains exactly one occurrence of $S$, and exactly at the end. Suppose $N = S$; if $w' = w$ then we are forced to apply $(S, \epsilon)$M; otherwise we have that $w'c$ is an initial segment of $w$ for exactly one $c \in \mathcal{A}$, and we are forced to apply the unique production rule for $S$ that begins with $c$. This proves that the grammar is unambiguous. □

The proof of Proposition 5.4.13 is thus complete.

### 5.4.5 A grammar for the language of a kernel

The following proposition is a generalization of Proposition 5.4.13, obtained by looking at the proof of Proposition 5.1.10, but with particular care to the size and to the unambiguity of the grammars involved.

Let $F_n$ be a free group generated by $a_1, ..., a_n$ and let $\mathcal{A} = \{a_1, ..., a_n, \overline{a}_1, ..., \overline{a}_n\}$. Let $F_m$ be a free group generated by $b_1, ...b_m$ and let $\mathcal{B} = \{b_1, ..., b_m, \overline{b}_1, ..., \overline{b}_m\}$.

Let $\phi : F_m \to F_n$ be a homomorphism. For $k = 1, ..., m$, let $\beta_k, \overline{\beta}_k \in \mathcal{A}^*$ be the reduced words representing $\phi(b_k), \phi(\overline{b}_k)$ respectively, so that $\overline{\beta}_k$ is the formal inverse of $\beta_k$. Denote with $\|\phi\| = \max\{l(\beta_1), ..., l(\beta_m)\}$.

**Proposition 5.4.18.** *Let $\mathcal{L}(\ker(\phi)) \subseteq \mathcal{B}^*$ be the language of all words representing an element of $\ker(\phi) \subseteq F_m$. Then the language $\mathcal{L}(\ker(\phi))$ is context-free and has a grammar $(\mathcal{N}, \mathcal{P}, S)$ such that:*

*(i)* $\|\mathcal{P}\|$ *is* $O(nm^3\|\phi\|^3)$ *and* $\mathrm{ram}(\mathcal{P}) = 2$.
*(ii)* $(\mathcal{N}, \mathcal{P}, S)$ *is unambiguous.*

Let $\Lambda = \{\lambda \in \mathcal{A}^* : \lambda \text{ is a final segment of some of } \beta_1, ..., \beta_m, \overline{\beta}_1, ..., \overline{\beta}_m\}$, including the empty final segment $\epsilon$ and the full final segments $\beta_1, ..., \beta_m, \overline{\beta}_1, ..., \overline{\beta}_m$. The grammar has non-terminal symbols $S^\lambda, A_i^{\lambda,\mu}, \overline{A}_i^{\lambda,\mu}$ for $i = 1, ..., n$ and $\lambda, \mu \in \Lambda$. The initial symbol is $S^\epsilon$ and the production rules are as follows:

$$
\begin{array}{lcll}
S^\epsilon & \to & \epsilon \mid b_k S^{\beta_k} \mid \overline{b}_k S^{\overline{\beta}_k} & \text{for } k = 1, ..., m. \\
S^{a_i\lambda} & \to & \overline{A}_i^{\lambda,\nu} S^\nu & \text{for } \nu \in \Lambda. \\
S^{\overline{a}_i\lambda} & \to & A_i^{\lambda,\nu} S^\nu & \text{for } \nu \in \Lambda.
\end{array}
$$

$$
\begin{array}{lcll}
A_i^{\epsilon,\mu} & \to & b_k A_i^{\beta_k,\mu} \mid \overline{b}_k A_i^{\overline{\beta}_k,\mu} & \text{for } k = 1, ..., m. \\
A_i^{a_i\lambda,\lambda} & \to & \epsilon & \\
A_i^{a_j\lambda,\mu} & \to & \overline{A}_j^{\lambda,\nu} A_i^{\nu,\mu} & \text{for } j = 1, ..., n \text{ with } j \neq i \text{ and } \nu \in \Lambda. \\
A_i^{\overline{a}_j\lambda,\mu} & \to & A_j^{\lambda,\nu} A_i^{\nu,\mu} & \text{for } j = 1, ..., n \text{ and } \nu \in \Lambda.
\end{array}
$$

$$
\begin{array}{lcll}
\overline{A}_i^{\epsilon,\mu} & \to & b_k \overline{A}_i^{\beta_k,\mu} \mid \overline{b}_k \overline{A}_i^{\overline{\beta}_k,\mu} & \text{for } k = 1, ..., m. \\
\overline{A}_i^{\overline{a}_i\lambda,\lambda} & \to & \epsilon & \\
\overline{A}_i^{\overline{a}_j\lambda,\mu} & \to & A_j^{\lambda,\nu} \overline{A}_i^{\nu,\mu} & \text{for } j = 1, ..., n \text{ with } j \neq i \text{ and } \nu \in \Lambda. \\
\overline{A}_i^{a_j\lambda,\mu} & \to & \overline{A}_j^{\lambda,\nu} \overline{A}_i^{\nu,\mu} & \text{for } j = 1, ..., n \text{ and } \nu \in \Lambda.
\end{array}
$$

Notice that there are no production rules for the symbols $A_i^{a_i\lambda,\mu}$ and $\overline{A}_i^{\overline{a}_i\lambda,\mu}$ if $\mu \neq \lambda$. The rest of this section is dedicated to the proof that the above grammar has the desired properties.

Define $\psi : \mathcal{B}^* \to \mathcal{A}^*$ as the unique map such that $\psi(b_k) = \beta_k$ and $\psi(\overline{b}_k) = \overline{\beta}_k$ for $k = 1, ..., m$ and such that $\psi$ preserves concatenations.

**Lemma 5.4.19.** *For* $u \in \mathcal{A}^*$ *and* $i = 1, ..., n$ *we have that* $A_i^{\lambda,\mu} \xrightarrow{*} u$ *(resp.* $\overline{A}_i^{\lambda,\mu} \xrightarrow{*} u$*) if and only if the following three conditions hold:*

*(i)* $\lambda \cdot \psi(u) = \alpha \cdot \mu$ *for some* $\alpha \in \mathcal{A}^*$ *with* $\alpha \equiv a_i$ *in* $F_n$ *(resp.* $\alpha \equiv \overline{a}_i$*).*

*(ii) No proper initial segment* $\alpha'$ *of* $\alpha$ *satisfies* $\alpha' \equiv a_i$ *in* $F_n$ *(resp.* $\alpha' \equiv \overline{a}_i$*).*

*(iii) The word* $\alpha$ *is not an initial segment of* $\lambda \cdot \psi(u')$ *for any proper initial segment* $u'$ *of* $u$.

*Proof.* We prove that if $A_i^{\lambda,\mu} \xrightarrow{*} u$ then (i), (ii) and (iii) hold, by induction on the length of the derivation. The base step, when the derivation has length one, is trivial. Suppose $A_i^{\epsilon,\mu} \xrightarrow{*} u$ and suppose without loss of generality the first production rule used is $A_i^{\epsilon,\mu} \to b_k A_i^{\beta_k,\mu}$: then we can write $u = b_k u_1$ with $A_i^{\beta_k,\mu} \xrightarrow{*} u_1$. By inductive hypothesis we have that $\beta_k \cdot \psi(u_1) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii). But then we have $\epsilon \cdot \psi(u) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii), as desired.

Suppose $A_i^{\overline{a}_j\lambda,\mu} \stackrel{*}{\to} u$ for some $j \in \{1,...,n\}$: then the first production used is of the form $A_i^{\overline{a}_j\lambda,\mu} \to A_j^{\lambda,\nu}A_i^{\nu,\mu}$ and we can write $u = u_1u_2$ with $A_j^{\lambda,\nu} \stackrel{*}{\to} u_1$ and $A_i^{\nu,\mu} \stackrel{*}{\to} u_2$. By inductive hypothesis we can write $\lambda \cdot \psi(u_1) = \alpha_1 \cdot \nu$ where $\alpha_1 \equiv a_j$ satisfies (ii) and (iii); similarly we can write $\nu \cdot \psi(u_2) = \alpha_2 \cdot \mu$ where $\alpha_2 \equiv a_i$ satisfies (ii) and (iii). But then we have that $\overline{a}_j\lambda \cdot \psi(u) = \overline{a}_j\alpha_1\alpha_2 \cdot \mu$ where $\overline{a}_j\alpha_1\alpha_2 \equiv a_i$; property (ii) for $\alpha_1$ and for $\alpha_2$, together with Lemma 5.4.14, implies that $\overline{a}_j\alpha_1\alpha_2$ satisfies (ii); property (iii) for $\alpha_1$ and for $\alpha_2$ easily implies property (iii) for $\overline{a}_j\alpha_1\alpha_2$, as desired.

The case $A_i^{a_j\lambda,\mu} \stackrel{*}{\to} u$ for some $j \in \{1,...,n\}$ with $j \neq i$ is completely analogous. The case $A_i^{a_i\lambda,\mu} \stackrel{*}{\to} u$ can only happen if $\lambda = \mu$ and only in the base step of the induction.

We prove that if $u$ satisfies conditions (i), (ii) and (iii) for some $i \in \{1,...,n\}$ and $\lambda, \mu \in \Lambda$, then $A_i^{\lambda,\mu} \stackrel{*}{\to} u$; we proceed by induction on $l(u) + l(\lambda)$. The base step, where $u = \epsilon$ and $\lambda = \epsilon$, is true since (i) can't hold.

Suppose $\lambda = \epsilon$ and $\epsilon \cdot \psi(u) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii). Without loss of generality assume that $u = b_ku_1$ for some $k \in \{1,...,m\}$. We have that $\beta_k \cdot \psi(u_1) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii): by inductive hypothesis we have $A_i^{\beta_k,\mu} \stackrel{*}{\to} u_1$. It follows that $A_i^{\epsilon,\mu} \stackrel{*}{\to} u$, as desired.

Suppose $\lambda = \overline{a}_j\lambda_1$ for some $j \in \{1,...,n\}$ and suppose $\lambda \cdot \psi(u) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii). Using Lemma 5.4.14 we can write $\alpha = \overline{a}_j\alpha_1\alpha_2$ where $\alpha_1 \equiv a_j$, and without loss of generality we can assume that $\alpha_1$ is the shortest with this property. Write also $u = u_1u_2$ where $u_1$ is the shortest such that $\lambda \cdot \psi(u_1)$ contains $\overline{a}_j\alpha_1$ as an initial segment, and write $\lambda \cdot \psi(u_1) = \overline{a}_j\alpha_1 \cdot \nu$ for some $\nu \in \Lambda$. Now we have that $\lambda_1 \cdot \psi(u_1) = \alpha_1 \cdot \nu$ where $\alpha_1 \equiv a_j$ satisfies (ii) and (iii); we also have that $\nu \cdot \psi(u_2) = \alpha_2 \cdot \mu$ where $\alpha_2 \equiv a_i$ satisfies (ii) and (iii). By inductive hypothesis we have that $A_j^{\lambda_1,\nu} \stackrel{*}{\to} u_1$ and $A_i^{\nu,\mu} \stackrel{*}{\to} u_2$ and thus $A_i^{\lambda,\mu} \to A_j^{\lambda_1,\nu}A_i^{\nu,\mu} \stackrel{*}{\to} u$, as desired.

The case $\lambda = \overline{a}_j\lambda_1$ for some $j \in \{1,...,n\}$ with $j \neq i$ is analogous.

Suppose $\lambda = a_i\lambda_1$ and suppose $\lambda \cdot \psi(u) = \alpha \cdot \mu$ where $\alpha \equiv a_i$ satisfies (ii) and (iii). Since the first letter of $\lambda$ is $a_i$, the same must hold for $\alpha$ too, and thus property (ii) implies $\alpha = a_i$. Now property (iii) implies $u = \epsilon$, and it follows that $\lambda_1 = \mu$. The production rule $A_i^{a_i\lambda_1,\lambda_1} \to \epsilon$ gives us the conclusion. □

**Lemma 5.4.20.** *For $u \in \mathcal{B}^*$ we have that $S^\lambda \stackrel{*}{\to} u$ if and only if $\lambda \cdot \psi(u) \equiv 1$ in $F_n$.*

*Proof.* We prove that if $S^\lambda \stackrel{*}{\to} u$ then $\lambda \cdot \psi(u) \equiv 1$, by induction on the length of the derivation. The base step, when the length of the derivation is one, is trivial. Suppose $S^\epsilon \stackrel{*}{\to} u$, and without loss of generality write $u = b_ku_1$ for some $k \in \{1,...,m\}$: then the first production applied must be $S^\epsilon \to b_kS^{\beta_k} \stackrel{*}{\to} b_ku_1$, and by inductive hypothesis on $S^{\beta_k}$ we are done. Suppose $S^{a_i\lambda} \stackrel{*}{\to} u$: then the first production applied must be of the form $S^{a_i\lambda} \to \overline{A}_i^{\lambda,\nu}S^\nu$ and the conclusion follows from Lemma 5.4.19 and from the inductive hypothesis on $S^\nu$.

We prove that if $\lambda \cdot \psi(u) \equiv 1$ then $S^\lambda \stackrel{*}{\to} u$, by induction on $l(u) + l(\lambda)$. The base step, when $u = \epsilon$ and $\lambda = \epsilon$, is trivial. Suppose $\lambda = \epsilon$ and $\psi(u) \equiv 1$, and

without loss of generality assume that $u = b_k u_1$ for some $k \in \{1, ..., m\}$: then we have $\beta_k \cdot \psi(u_1) \equiv 1$ and thus by inductive hypothesis $S^{\beta_k} \xrightarrow{*} u_1$, which yields $S^\epsilon \to b_k S^{\beta_k} \xrightarrow{*} b_k u_1$, as desired.

Finally, suppose $\lambda = \overline{a}_i \lambda_1$ for some $i \in \{1, ..., n\}$ (the case $\lambda = a_i \lambda_1$ being analogous) and suppose $\lambda \cdot \psi(u) \equiv 1$. Then we have $\lambda_1 \cdot \psi(u) \equiv a_i$ and let $\alpha$ be the shortest initial segment of $\lambda_1 \cdot \psi(u)$ which satisfies $\alpha \equiv a_i$. Write $u = u_1 u_2$ so that $\lambda_1 \cdot \psi(u_1)$ contains $\alpha$ as initial segment, and $u_1$ is the shortest with this property. Then we can write $\lambda_1 \cdot \psi(u_1) = \alpha \cdot \nu$ for some $\nu \in \Lambda$ and we have that $\alpha \equiv a_i$ and $\alpha$ satisfies properties (ii) and (iii) of Lemma 5.4.19: it follows that $A_i^{\lambda_1, \nu} \xrightarrow{*} u_1$. Since $\nu \cdot \psi(u_2) \equiv 1$, by inductive hypothesis we have $S^\nu \xrightarrow{*} u_2$ and thus $S^\lambda \to A_i^{\lambda_1, \nu} S^\nu \xrightarrow{*} u_1 u_2$, as desired. □

**Lemma 5.4.21.** *The above grammar is unambiguous.*

*Proof.* Suppose we are given a leftmost derivation $S^\epsilon = v_0 \to v_1 \to ... \to v_{r-1} \to v_r = u$ for a certain word $u \in \mathcal{B}^*$ and we want to prove that the production rule to be applied on $v_k$ is forced, for $k = 0, ..., r-1$. Write $v_k = u' N p$ with $u' \in \mathcal{B}^*$ and $N \in \mathcal{N}$ and $p \in (\mathcal{B} \cup \mathcal{N})^*$, and notice that $u'$ must be an initial segment of $u$, let's say $u = u' u''$.

Suppose $N = A_i^{\epsilon, \mu}$ for some $i \in \{1, ..., n\}$ and $\mu \in \Lambda$. Each production for $A_i^{\epsilon, \mu}$ begins with a terminal symbol, and thus $u''$ is non-empty and we can call $d \in \mathcal{B}$ the first letter of $u''$. But there is exactly one production rule for $A_i^{\epsilon, \mu}$ that begins with $d$, and so we are forced to apply that production rule.

Suppose $N = A_i^{c\lambda, \mu}$ for some $i \in \{1, ..., n\}$ and $c \in \mathcal{A}$ and $c\lambda, \mu \in \Lambda$. If $c = a_i$ then there is at most one derivation for $A_i^{a_i \lambda, \mu}$ and thus we have no choice. If $c = \overline{a}_j$ for some $j \in \{1, ..., n\}$, then we only have derivations of the form $A_i^{\overline{a}_j \lambda, \mu} \to A_j^{\lambda, \nu} A_i^{\nu, \mu}$ and we need to prove that $\nu \in \Lambda$ is uniquely determined. Let's say that during the rest of the derivation the symbol $A_j^{\lambda, \nu}$ derives a word $t \in \mathcal{B}^*$: we must have that $t$ is an initial segment of $u''$ and thus $\lambda \cdot \psi(t)$ is an initial segment of $\lambda \cdot \psi(u'')$. But we have $\lambda \cdot \psi(t) = \alpha \cdot \nu$ where $\alpha \in \mathcal{A}^*$ represents $a_j \in F_n$ and properties (ii) and (iii) of Lemma 5.4.19 hold; property (ii) uniquely determines $\alpha$ to be the shortest initial segment of $\lambda \cdot \psi(u'')$ representing $a_j \in F_n$; property (iii) uniquely determines $t$ to the the shortest initial segment of $u''$ such that $\lambda \cdot \psi(t)$ contains $\alpha$. Since $\lambda \cdot \psi(t) = \alpha\nu$ we have that $\nu$ is uniquely determined too, as desired. The same argument works if $c = a_j$ for some $j \in \{1, ..., n\}$ and $j \neq i$.

In a similar way we deal with the case $N = \overline{A}_i^{\lambda, \mu}$ for $i \in \{1, ..., n\}$ and $\lambda, \mu \in \Lambda$.

It is easy to show by induction that each of $v_0, ..., v_{r-1}$ contains exactly one occurrence of one of $\{S^\lambda : \lambda \in \Lambda\}$, and exactly at the end. Suppose $N = S^\epsilon$: if $u' = u$ then we are forced to use the production rule $S^\epsilon \to \epsilon$, otherwise we look at the first letter of $u''$ and this forces our choice of production rule to use. Suppose $N = S^{a_i \lambda}$ with $i \in \{1, ..., n\}$ and $a_i \lambda \in \Lambda$: then we must use the production rule $S^{a_i \lambda} \to \overline{A}_i^{\lambda, \nu} S^\nu$ and

$\nu \in \Lambda$ is uniquely determined with the same argument as above. The case $N = S^{\overline{a}_i \lambda}$ with $i \in \{1, ..., n\}$ and $\overline{a}_i \lambda \in \Lambda$ is completely analogous.

This proves that the grammar is unambiguous. □

From the explicit description of the grammar, it is very easy to estimate that $\|\mathcal{P}\|$ is at most $O(nm|\Lambda|) + O(n|\Lambda|^3)$. We estimate $|\Lambda|$ as $O(m\|\phi\|)$ and we obtain that $\|\mathcal{P}\|$ is $O(nm^3\|\phi\|^3)$. The proof of Proposition 5.4.18 is thus complete.

### 5.4.6 Complexity of the problem of the existence of equations

We are ready to provide upper bounds on the running time of the algorithms of Theorems 5.2.2 and 5.2.4. Notice that all the running times provided in this section are polynomial in the input data.

Let $F_n$ be a free group on $n$ generators. Let $H \leq F_n$ be a finitely generated subgroup of rank $r$, and let $g \in F_n$ be any element. Define $\varphi_g : H * \langle x \rangle \to F_n$ be the map that is the inclusion on $H$ and that sends $x$ to $g$; in particular we have $\mathfrak{I}_g = \ker \varphi_g$. Let $L = \|\varphi_g\| = \max\{l(h_1), ..., l(h_r), l(g)\}$ where we mean the length of the reduced words representing those elements.

**Theorem 5.4.22.** *We have the following:*

*(i) The algorithm in Theorem 5.2.2 can be constructed so as to run in time $O(nr^9L^3)$.*

*(ii) Moreover, this algorithm will output data that is sufficient for a description of an explicit equation (following the procedure described in Proposition 5.4.3).*

*(iii) If the ideal $\mathfrak{I}_g$ contains a non-trivial equation of degree at most $D$, then the algorithm explicitly writes down a non-trivial equation in time $O(nr^9L^7D^2)$.*

*Proof.* According to Proposition 5.4.18, we have an (unambiguous) grammar $(\mathcal{N}, \mathcal{P}, S)$ for $\mathcal{L}(\mathfrak{I}_g)$ such that $\|\mathcal{P}\|$ is $O(nr^3L^3)$ and $\mathrm{ram}(\mathcal{P}) = 2$. For the regular language $\mathcal{R}$ of all non-empty reduced words in $\{h_1, ..., h_r, x, \overline{h}_1, ..., \overline{h}_r, \overline{x}\}$, it is easy to provide a deterministic finite automaton $(Q, \delta, \{q_0\}, Q_f)$ such that $|Q|$ is $O(r)$. Thus we can produce a context-free grammar $(\mathcal{N}', \mathcal{P}', S')$ for the (unambiguous) language $\mathcal{L}(\mathfrak{I}_g) \cap \mathcal{R}$ such that $\|\mathcal{P}'\|$ is $O(nr^9L^3)$. The first statement of Theorem 5.4.22 now follows from Propositions 5.4.2 and 5.4.3.

If the ideal contains an equation of degree at most $D$, then by Theorem 4.2.2 we have that $\mathfrak{I}_g$ must contain a non-trivial equation of length $O(L^2D)$. It follows from Proposition 5.4.6 that the algorithm writes down an explicit equation in time $O(nr^9L^7D^2)$. □

**Theorem 5.4.23.** *We have the following:*

*(i) The algorithm in Theorem 5.2.4 can be constructed so as to run in time $O(nr^{15}L^3d^6)$.*

*(ii) Moreover, this algorithm will output data sufficient for a description of an explicit equation (following the procedure described in Proposition 5.4.2).*

*(iii) In addition, this algorithm explicitly writes down an equation in $\mathfrak{J}_{g,d}$ in time $O(nr^{15}L^{11}d^{10})$.*

*Proof.* Completely analogous to the proof of Theorem 5.4.22. Instead of Theorem 4.2.2 we use Theorem 4.3.10. □

### 5.4.7 The minimum possible degree for equations in an ideal

**Theorem 5.4.24** ([Asc22a])**.** *There is an algorithm that, given $H \leq F_n$ finitely generated and $g \in F_n$, computes the minimum possible degree for an equation in $\mathfrak{J}_g$. The algorithm runs in time $O(nr^9L^3\log(nrL))$.*

*Proof.* We take the unambiguous grammar for $\mathcal{L}(\mathfrak{J}_g)$ as described in Proposition 5.4.18 and we run the algorithm of Proposition 5.4.5; as size function we use $\sigma : \{h_1, ..., h_r, x, \overline{h}_1, ..., \overline{h}_r, \overline{x}\} \to \mathbb{N}$ given by $\sigma(h_i) = \sigma(\overline{h}_i) = 0$ for $i = 1, ..., r$ and $\sigma(x) = \sigma(\overline{x}) = 1$. □

We now discuss the problem of finding an estimate on the minimum possible degree $d_{\min}$ for a non-trivial equation in $\mathfrak{J}_g$. A good bound on $d_{\min}$ leads to a good bound on the running time of the algorithm of Theorem 5.4.22. We first need a preliminary lemma.

**Proposition 5.4.25.** *Let $F_m$ be a finitely generated free group with a fixed basis $c_1, ..., c_m$. Let $\psi : F_m \to F_m$ be an automorphism and suppose each of $\psi(c_1), ..., \psi(c_m)$ has length at most $C$ when written in the basis $c_1, ..., c_m$. Then there is an inner automorphism $\rho : F_m \to F_m$ such that each of $\rho \circ \psi^{-1}(c_1), ..., \rho \circ \psi^{-1}(c_m)$ has length at most $K_m C^{M_m}$ when written in the basis $c_1, ..., c_m$, for some constants $K_m, M_m$ independent on $\psi$.*

*Proof.* Immediately follows from Corollary 4.6 of [LSV15]. The constants $K_m, M_m$ are the ones introduced in [LSV15] too. □

Mantaining the notation of Section 5.4.6, fix $F_n$ and $H \leq F_n$ finitely generated and $g \in F_n$; denote with $h_1, ..., h_r$ a fixed basis for $H$, with $\varphi_g : H * \langle x \rangle \to F_n$ the evaluation map (whose kernel is $\mathfrak{J}_g$), and with $L = \|\varphi_g\| = \max\{l(h_1), ..., l(h_r), l(g)\}$.

**Theorem 5.4.26.** *If it is non-trivial, the ideal $\mathfrak{J}_g$ contains a non-trivial equation of degree at most $2K_{r+1}L^{M_{r+1}}$.*

*Proof.* Using the algorithm of Theorem 1.3.4, we find a basis $c_1, ..., c_{r+1}$ for $H * \langle x \rangle$ such that each of $h_1, ..., h_r, x$ written as a reduced word in $c_1, ..., c_{r+1}$ has at length at most $L$; moreover, the ideal $\mathfrak{J}_g$ is generated (as normal subgroup) by words in $c_1, ..., c_{r+1}$ of length at most 2.

Take the automorphism $\psi : H * \langle x \rangle \to H * \langle x \rangle$ with $\psi(c_1) = h_1, ..., \psi(c_r) = h_r, \psi(c_{r+1}) = x$. Each of $\psi(c_1), ..., \psi(c_{r+1})$ has length at most $L$ when written in the basis $c_1, ..., c_{r+1}$. By Proposition 5.4.25, there is a conjugation automorphism $\rho : H * \langle x \rangle \to H * \langle x \rangle$ such that each of $\psi^{-1} \circ \rho(c_1), ..., \psi^{-1} \circ \rho(c_{r+1})$ has length at most $K_{r+1} L^{M_{r+1}}$ when written in the basis $c_1, ..., c_{r+1}$.
In particular we have that each of $\psi(\psi^{-1} \circ \rho(c_1)), ..., \psi(\psi^{-1} \circ \rho(c_{r+1}))$ has length at most $K_{r+1} L^{M_{r+1}}$ when written in the basis $\psi(c_1), ..., \psi(c_{r+1})$; this means that each of $\rho(c_1), ..., \rho(c_{r+1})$ has length at most $K_{r+1} L^{M_{r+1}}$ when written in the basis $h_1, ..., h_r, x$. We can take an equation which is a word of length at most two in $\rho(c_1), ..., \rho(c_{r+1})$, and thus is a word of length at most $2K_{r+1} L^{M_{r+1}}$ in $h_1, ..., h_r, x$. In particular its degree is at most $2K_{r+1} L^{M_{r+1}}$, as desired. □

One may hope to improve the bound of Theorem 5.4.26 to a bound which is polynomial in both $r, L$. Unfortunately this isn't possible, as shown in Section 5.5.4, where we provide examples of ideals $\mathfrak{I}_g$ where $r = n + 1$ and $L \approx p^2$ and the minimum possible degree for a non-trivial equation is at least $\approx p^n$, which is $\approx L^{\frac{r-1}{2}}$.

### 5.4.8 Complexity of the problem of computing the growth rates

**Theorem 5.4.27.** *There is an algorithm as in Theorem 5.3.5 that runs in polynomial time.*

*Proof.* According to Proposition 5.4.18 we can build, in polynomial time in $H, g, d$, a grammar $(\mathcal{P}, \mathcal{N}, S)$ for $\mathcal{L}(\mathfrak{I}_{g,d})$ with polynomial size $\|\mathcal{P}\|$. The algorithm of Theorem 5.3.2 runs in polynomial time too, according to [GKRS08]. We run such algorithm on such grammar, and the conclusion follows. □

We point out that most of the complexity of the algorithm of Theorem 5.3.2 comes from the possible presence of ambiguity in the grammar. If we are only interested in unambiguous grammars, then the algorithm can be made much easier and much faster. Since our grammar is obtained using Propositions 5.4.18 and 5.4.9, it is unambiguous, and so we are able to apply the easier version of the algorithm.

## 5.5 Examples

We now provide a few examples. In each of them we have a free group $F_n$, a subgroup $H \leq F_n$ and an element $g \in F_n$ that depends on $H$; we are interested in studying the growth rate of the functions $\rho_{g,d}$ according to Theorem 5.3.5. Some of the examples are taken from Section 4.4.

In Section 5.5.1 we deal with the case where $\text{rank}(H) = 1$. We show that $\rho_{g,d}(M)$ has polynomial growth rate for every $d \in D_g$, in contrast with Theorem 5.3.9.

In Example 5.5.2 we show that it is possible that the function $\rho_{g,d}(M)$ has polynomial growth rate of degree zero, i.e. it is constant for all $M$ big enough.

We also provide the additional Example 5.5.4, where the minimum degree in $D_g$ is big compared to the size of the subgroup $H$.

### 5.5.1 Cyclic subgroups

We here deal with the case where $\text{rank}(H) = 1$. As shown in Section 4.4.1, we can assume that $F_n = \langle a \rangle$ and that $H = \langle h \rangle$ where $h = a^m$ with $m \geq 1$, and that $g = a^k$ with $k \geq 0$ coprime with $m$.

Every cyclically reduced equation in $\mathfrak{I}_g$ has the form $w = h^{e_0} x^{f_1} h^{e_1} ... h^{e_{s-1}} x^{f_s} h^{e_s}$ with $s \geq 1$ and $e_1, ..., e_{s-1}, f_1, ..., f_s \in \mathbb{Z} \setminus \{0\}$ and $e_0, e_s \in \mathbb{Z}$ with the same sign (possibly one or both of them equal to zero). When we substitute $h = a^m$ and $x = a^k$ we obtain the trivial word, and thus we must have $(e_0 + ... + e_s)m + (f_1 + ... + f_s)k = 0$. The degree of such equation is $|f_1| + ... + |f_s|$.

We want to estimate $\rho_{g,d}(M)$ and thus we restrict our attention to equations of degree $d$ and length at most $M$. This implies that $s \leq d$ and gives only a finite number of possible choices for the numbers $f_1, ..., f_s$. Once $s, f_1, ..., f_s$ are fixed, we must choose $e_0, ..., e_s$ such that $e_0 + ... + e_s = -(f_1 + ... + f_s)k/m$ and such that $|e_0| + ... + |e_s| \leq M - d$. The number of possible choices is polynomial of degree at most $s$ in $M$.

This shows that $\rho_{g,d}(M)$ has polynomial growth of degree at most $d$ for every $m \geq 1$ and $k \geq 0$ and $d \in D_g$.

### 5.5.2 An ideal with a finite number of minimum-degree equations

As in Section 4.4.2, let $F_2 = \langle a, b \rangle$ and consider the subgroup $H = \langle h_1, h_2 \rangle$ with $h_1 = ba$ and $h_2 = ab^2\overline{a}$ and the element $g = a$. In this case the ideal $\mathfrak{I}_g$ is the normal subgroup generated by the equation $\overline{x}h_2 x x \overline{h}_1 x \overline{h}_1$ and we have $D_g = \{d : d \geq 4 \text{ even}\}$.

The function $\rho_{g,4}(M)$ is constant for all $M$ big enough, since $\mathfrak{I}_g$ only contains one equation of degree 4 up to inverse and conjugations, which is the generator of the ideal. The function $\rho_{g,6}(M)$ has polynomial growth of degree at least one. The function $\rho_{g,d}(M)$ has exponential growth for every $d \geq 8$ even, by Lemma 5.3.7.

### 5.5.3 An ideal where the minimum-degree equations have linear growth

As in Section 4.4.3, let $F_2 = \langle a, b\rangle$ and consider the subgroup $H = \langle h_1, h_2\rangle$ with $h_1 = b$ and $h_2 = ababa$. Let $g = a$. The ideal $\mathfrak{I}_g$ is the normal subgroup generated by $\overline{h}_2 x h_1 x h_1 x$ and we have $D_g = \{d : d \geq 2 \text{ integer}\}$.

The function $\rho_{g,2}(M)$ has linear growth, since a complete list of the degree 2 equations, up to inverses and conjugations, is given by $[(h_2h_1)^i, xh_1]$ for $i \in \mathbb{Z} \setminus \{0\}$. The function $\rho_{g,3}(M)$ has polynomial growth of degree at least one. The function $\rho_{g,d}(M)$ has exponential growth for $d \geq 4$, by Lemma 5.3.7.

### 5.5.4 An ideal with no small-degree equations

The idea of this example is taken from Section 4.1 of [LSV15], but some changes were needed to fit our purposes. Fix a natural number $p$ and set $u(y, x) = xyx^2yx^3...yx^{p+1}$.

For $n \geq 2$ let $F_{n+1} = \langle a_1, ..., a_n, b\rangle$. Consider the subgroup $H = \langle h_1, ..., h_n, h'\rangle$ where $h_1 = a_1$ and $h_i = a_i \cdot (u(a_{i-1}, b))^{-1}$ for $i = 2, ..., n$ and $h' = \overline{a}_n\overline{b}$. Consider the element $g = b$.

Consider the alphabet $\mathcal{H} = \{h_1, \overline{h}_1, ..., h_n, \overline{h}_n, h', \overline{h'}, x, \overline{x}\}$. Define $\widehat{a}_1, ..., \widehat{a}_n \in \mathcal{H}^*$ as follows: we set $\widehat{a}_1 = h_1$, and we define inductively $\widehat{a}_i = h_i u(\widehat{a}_{i-1}, x)$. Consider the natural projection map $\eta : \mathcal{H}^* \to H * \langle x\rangle$ defined as at the beginning of Section 5.1.3; consider also the evaluation homomorphism $\varphi_g : H * \langle x\rangle \to F_{n+1}$ which is the identity on $H$ and with $\varphi_g(x) = g$. It is easy to show by induction that $\varphi_g(\eta(\widehat{a}_i)) = a_i$. It is also easy to prove by induction that

(a) The word $\widehat{a}_i$ only contains the letters $h_1, ..., h_i, x$.

(b) For every two consecutive letters in $\widehat{a}_i$, at least one is $x$.

(c) The word $\widehat{a}_i$ has length $\frac{p^2+3p+4}{2} \cdot \frac{p^{i-1}-1}{p-1} + p^{i-1}$.

Finally, define the element $w \in \mathcal{H}^*$ given by $w = h'x\widehat{a}_n$. In Figure 5.1 we can see the core graph $\text{core}_*(H)$. By means of the algorithm of Theorem 1.3.4 we can prove that $\mathfrak{I}_g$ is generated, as a normal subgroup of $H * \langle x\rangle$, by the single element $\eta(w)$.

We now want to apply Theorem 4.4 of [LS01] to the ideal $\mathfrak{I}_g$. We take the symmetrized set $W$ given by the cyclic permutations of $w$ and of $\overline{w}$, and we want to prove that the small cancellation property $C'(1/6)$ holds. A cyclic permutation of $w$ and a cyclic permutation of $\overline{w}$ can't have an initial segment in common, since $w$ only contains the letters $h_1, ..., h_n, h', x$. Suppose $w', w''$ are distinct cyclic permutations of $w$ with a common initial segment $u$. We observe that, since $w$ only contains one occurrence of $h'$, $u$ can't contain $h'$. We can write

$$w = h'xh_n \cdot xh_{n-1}u(\widehat{a}_{n-2}, x) \cdot x^2h_{n-1}u(\widehat{a}_{n-2}, x) \cdot ... \cdot x^ph_{n-1}u(\widehat{a}_{n-2}, x) \cdot x^{p+1}$$

and it is easy to see that, between two consecutive occurrences of $h_{n-1}$, there is always a different number of letters. This implies that $u$ contains at most one

occurrence of $h_{n-1}$, and in particular $l(u)$ is at most $\approx \frac{2}{p} \cdot l(w)$. For $p$ big enough this proves that $C'(1/6)$ holds in this case.

By Dehn's algorithm (Theorem 4.4 of [LS01]), every equation in $\mathfrak{I}_g$ has a subword of length $\frac{1}{2}|w|$ in common with a cyclic permutation of $w$ or $\overline{w}$. For any two consecutive letters of $w$, at least one is $x$; a subword of length $\frac{1}{2}|w|$ has to contain at least $\frac{1}{4}|w| - 1$ letters $x$ or $\overline{x}$. This shows that every non-trivial equation in $\mathfrak{I}_g$ has degree at least

$$\frac{p^2+3p+4}{8} \cdot \frac{p^{n-1}-1}{p-1} + \frac{1}{4}p^{n-1} - \frac{1}{2}$$

which, for $p$ big enough, is at least $p^n/8$.

Thus in this case we have a subgroup $H$ with $n+1$ generators, each of length at most $\frac{p^2+5p+6}{2}$, and where the minimum degree for a non-trivial equation in $\mathfrak{I}_g$ is at least $p^n/8$.

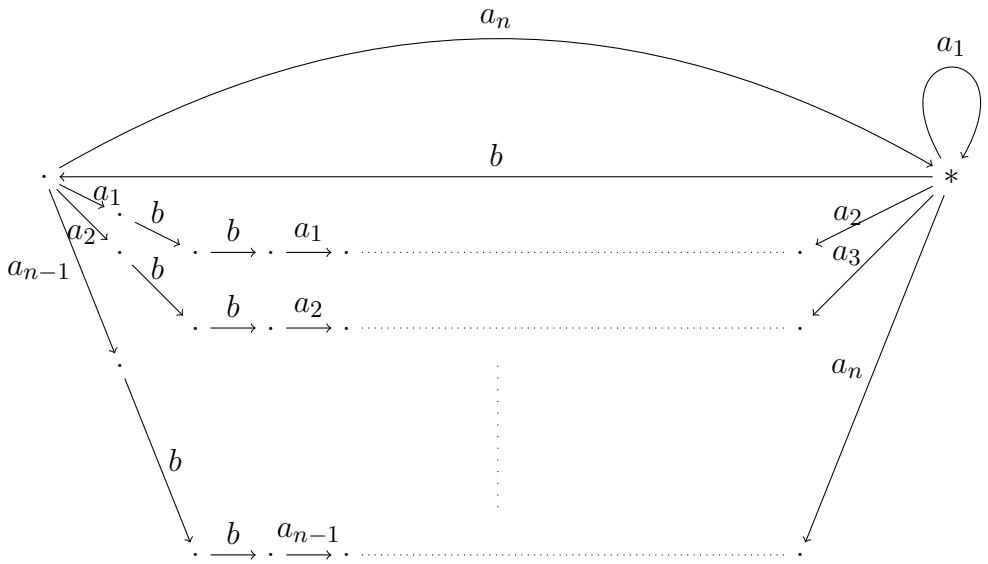


Figure 5.1: The core graph core$(H)$ of the subgroup $H$ of Section 5.5.4.

## 5.6 Equations in more variables

We point out that most of the results of this chapter can be generalized to equations in more than one variable. We gather the statements of these results in this section, but since the proofs are completely analogous to the one-variable case, they will be omitted. Let $F_n$ be a free group generated by $n$ elements $a_1, ..., a_n$.

Let $H \leq F_n$ be a finitely generated subgroup and let $\langle x_1 \rangle, \langle x_2 \rangle, ..., \langle x_m \rangle \cong \mathbb{Z}$ be infinite cyclic groups with generators $x_1, ..., x_m$ respectively. Let $(g_1, ..., g_m) \in (F_n)^m$ be an $m$-tuple of elements that depends on $H$; see Section 4.5 for the definition of the

ideal $\mathfrak{I}_{g_1,...,g_m}$, as well as for the definition of multi-degree $(d_1, ..., d_m)$ of an equation $w \in \mathfrak{I}_{g_1,...,g_m}$. Theorems 5.2.1, 5.2.2 and 5.4.22 can be generalized as follows:

**Theorem 5.6.1.** *The set $\mathfrak{I}_{g_1,...,g_m}$ is context-free as a subset of $H * \langle x_1 \rangle * ... * \langle x_m \rangle$. There is an algorithm that, given $H, g_1, ..., g_m$, tells us whether $\mathfrak{I}_{g_1,...,g_m}$ contains a non-trivial equation or not, and in case also produces data sufficient for an explicit description of a non-trivial equation in $\mathfrak{I}_{g_1,...,g_m}$ (as specified in Proposition 5.4.5). The algorithm runs in polynomial time.*

As in the one-variable case, we consider the subset $\mathfrak{J}_{g_1,...,g_m,(d_1,...,d_m)}$ of $\mathfrak{I}_{g_1,...,g_m}$ given by all the equations of multi-degree $(d_1, ..., d_m)$. Theorems 5.2.3, 5.2.4 and 5.4.23 can be generalized as follows:

**Theorem 5.6.2.** *The set $\mathfrak{J}_{g_1,...,g_m,(d_1,...,d_m)}$ is context-free as a subset of $H * \langle x_1 \rangle * ... * \langle x_m \rangle$. There is an algorithm that, given $H, g_1, ..., g_m, d_1, ..., d_m$, tells us whether the set $\mathfrak{J}_{g_1,...,g_m,(d_1,...,d_m)}$ is empty or not, and in case explicitly writes down an element of $\mathfrak{J}_{g_1,...,g_m,(d_1,...,d_m)}$. The algorithm runs in polynomial time.*

It is also possible to generalize Theorem 5.4.26 as follows:

**Theorem 5.6.3.** *The ideal $\mathfrak{I}_{g_1,...,g_m}$ contains a non-trivial equation of multi-degree $(d_1, ..., d_m)$ with $d_1 + ... + d_m \leq 2K_{r+m}L^{M_{r+m}}$.*

As in Section 5.3.2, we can define the functions $\rho_{g_1,...,g_m}(M)$ and $\rho_{g_1,...,g_m,(d_1,...,d_m)}(M)$ as the growth functions of the languages $\mathcal{L}_{\mathrm{cr}}(\mathfrak{I}_{g_1,...,g_m})$ and $\mathcal{L}_{\mathrm{cr}}(\mathfrak{J}_{g_1,...,g_m,(d_1,...,d_m)})$ respectively. The analogous of Theorems 5.3.4, 5.3.5 and 5.4.27 hold:

**Theorem 5.6.4.** *The function $\rho_{g_1,...,g_m}(M)$ has exponential growth.*

**Theorem 5.6.5.** *The function $\rho_{g_1,...,g_m,(d_1,...,d_m)}(M)$ has either exponential growth, or polynomial growth of degree $k$ for some $k \in \mathbb{N}$. Moreover, there is an algorithm that, in polynomial time, tells us which case takes place, and in the second case computes the degree $k$ of the growth.*

# Bibliography


[AM22] D. Ascari and F. Milizia. Weakly bounded cohomology classes and a counterexample to a conjecture of Gromov, 2022. arXiv:2207.03972.

[Asc21] D. Ascari. A fine property of Whitehead's algorithm, 2021. arXiv:2110.11936.

[Asc22a] D. Ascari. Ideals of equations for elements in a free group and context-free languages, 2022. arXiv:2211.10276.

[Asc22b] D. Ascari. Ideals of equations for elements in a free group and Stallings folding, 2022. arXiv:2207.04759.

[BB22] M. Bestvina and M. R. Bridson. Rigidity of the complex of free factors, 2022.

[BF14] M. Bestvina and M. Feighn. Hyperbolicity of the Complex of Free Factors. *Adv. Math.*, 256:104–155, 2014.

[BFH20] M. Bestvina, M. Feighn, and M. Handel. A McCool Whitehead Type Theorem for Finitely Generated Subgroups of $\mathsf{Out}(F_n)$, 2020.

[BG02] M. R. Bridson and R. H. Gilman. Context-free languages of sub-exponential growth. *Journal of Computer and System Sciences*, 64(2):308–310, 2002.

[BH92] M. Bestvina and M. Handel. Train tracks and automorphisms of free groups. *Annals of Mathematics*, 135(1):1–51, 1992.

[BL18] B. Beeker and N. Lazarovich. Stallings' folds for cube complexes. *Israel Journal of Mathematics*, 227(1):331–363, 2018.

[BM12] O. Bogopolski and O. Maslakova. An efficient algorithm for finding a basis of the fixed point subgroup of an automorphism of a free group. *International Journal of Algebra and Computation*, 26, 04 2012.

[BMR99] Gilbert Baumslag, Alexei Myasnikov, and Vladimir Remeslennikov. Algebraic geometry over groups i. algebraic sets and ideal theory. *Journal of Algebra*, 219(1):16–79, 1999.

[CG10] A. Clifford and R. Z. Goldstein. Subgroups of Free Groups and Primitive Elements. *J. Group Theory*, 13(4):601–611, 2010.

[CV86] M. Culler and K. Vogtmann. Moduli of Graphs and Automorphisms of Free Groups. *Invent. Math.*, 84(1):91–119, 1986.

[DV96] W. Dicks and E. Ventura. The group fixed by a family of injective endomorphisms of a free group. *Contemp. Math.*, 195:1–81, 1996.

[Ger84] S. M. Gersten. On Whitehead's Algorithm. *Bull. Amer. Math. Soc. (N.S.)*, 10(2):281–284, 1984.

[Ger87] S. M. Gersten. Fixed points of automorphisms of free groups. *Advances in Mathematics*, 64(1):51–85, 1987.

[GKRS08] P. Gawrychowski, D. Krieger, N. Rampersad, and J. Shallit. Finding the growth rate of a regular language in polynomial time. In *Developments in Language Theory*, pages 339–358, 2008.

[HMU06] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation - 3rd ed.* 2006.

[HW19] M. Heusener and R. Weidmann. A Remark on Whitehead's Cut-Vertex Lemma. *J. Group Theory*, 22(1):15–21, 2019.

[Inc01] R. Incitti. The growth function of context-free languages. *Theoretical Computer Science*, 255(1):601–605, 2001.

[KM04] O. Kharlampovich and A. Miasnikov. Effective JSJ decompositions. 08 2004.

[KM06] O. Kharlampovich and A. Myasnikov. Elementary theory of free non-abelian groups. *Journal of Algebra*, 302(2):451–552, 2006.

[KWM05] I. Kapovich, R. Weidmann, and A. Myasnikov. Foldings, graphs of groups and the membership problem. *International Journal of Algebra and Computation*, 15(01):95–128, 2005.

[LS01] R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory*. Classics in Mathematics. Springer-Verlag, Berlin, 2001. Reprint of the 1977 edition.

[LSV15] M. Ladra, P. V. Silva, and E. Ventura. Bounding the gap between a free group (outer) automorphism and its inverse. *Collectanea Mathematica*, 67(3):329–346, feb 2015.

[LW21] L. Louder and H. Wilton. Negative immersions for one-relator groups, 2021.

[Mak83] G. S. Makanin. Equations in a free group. *Mathematics of the USSR-Izvestiya*, 21(3):483, 1983.

[MKS04] W. Magnus, A. Karrass, and D. Solitar. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*. Dover books on mathematics. Dover Publications, 2004.

[MS83] D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26(3):295–310, 1983.

[MV04] A. Martino and E. Ventura. A description of auto-fixed subgroups in a free group. *Topology*, 43(5):1133–1164, 2004.

[Pud13] D. Puder. Primitive words, free factors and measure preservation. *Israel Journal of Mathematics*, 201(1):25–73, oct 2013.

[PW14] D. Puder and C. Wu. Growth of Primitive Elements in Free Groups. *J. London Math. Soc.*, 90(1):89–104, May 2014.

[Rap58] E. S. Rapaport. On Free Groups and their Automorphisms. *Acta Math.*, 99:139–163, 1958.

[Raz85] A. A. Razborov. On systems of equations in a free group. *Mathematics of the USSR-Izvestiya*, 25(1):115, feb 1985.

[Ros01] A. Rosenmann. On rank, root and equations in free groups. *Internat. J. Algebra Comput.*, 11(3):375–390, 2001.

[Ros13] A. Rosenmann. On the intersection of subgroups in free groups: Echelon subgroups are inert. *Groups - Complexity - Cryptology*, 5(2):211–221, 2013.

[RV21] A. Rosenmann and E. Ventura. Dependence and algebraicity over subgroups of free groups, 2021.

[Sel06] Z. Sela. Diophantine geometry over groups vi: The elementary theory of a free group. *Geometric and Functional Analysis*, 16:707–730, 06 2006.

[Sta83] J. R. Stallings. Topology of Finite Graphs. *Invent. Math.*, 71(3):551–565, 1983.

[Whi36a] J. H. C. Whitehead. On Certain Sets of Elements in a Free Group. *Proc. London Math. Soc. (2)*, 41(1):48–56, 1936.

[Whi36b] J. H. C. Whitehead. On Equivalent Sets of Elements in a Free Group. *Ann. of Math. (2)*, 37(4):782–800, 1936.

[Wil18] Henry Wilton. Essential surfaces in graph pairs, 2018.