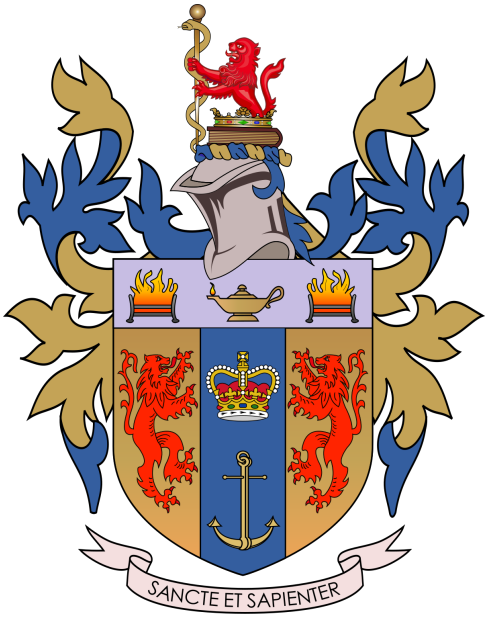
# Algorithmic problems in group theory

A thesis presented for the degree of
Bachelor of Science (Honours)
by

**Francesco Nishanil Chotuck**

Supervised by
Dr. David Sheard




Department of Mathematics
King's College London
United Kingdom
March 2024

# Abstract

In 1911, Max Dehn introduced the first three decision problems in group theory: the word, conjugacy, and isomorphism problems. This thesis delves into the word and conjugacy problems within finitely presented groups, examining their geometric representations in Cayley graphs and Van Kampen diagrams. It investigates Dehn's algorithms for the fundamental group of closed orientable surfaces with genus $g \geq 2$. Furthermore, it delves into the Novikov-Boone-Britton theorem, which illustrates the undecidability of the word problem for general finitely presented groups.

*Dedico questa tesi a*
*Francesco (D'Assisi) Morgese*

# Acknowledgements

I am deeply indebted to my supervisor, Dr. David Sheard, whose unwavering commitment to the subject has ignited my own enthusiasm and passion for the research at hand. Dr. Sheard has not only fostered my appreciation for the subject but has also pushed me to think geometrically, a skill that has significantly enriched the content of this thesis. Many of the intricate figures and geometric representations found within these pages are a testament to his guidance and expertise. Dr. Sheard's keen eye for geometric detail and his insistence on precision have been instrumental in shaping the visual narrative of this work, and I am profoundly grateful for his invaluable contributions in this regard. Dr. Sheard has demonstrated an unparalleled dedication to the field, consistently showcasing his profound knowledge, insight, and genuine excitement for the subject. His contagious enthusiasm has served as a catalyst, inspiring me to delve deeper into the intricacies of the topic and to approach my research with a newfound fervour and curiosity. Moreover, I am profoundly grateful to Dr. Sheard for bringing this subject to my attention and for recognising its potential as a focal point for academic inquiry. His keen insight and discerning eye for promising research avenues have been instrumental in shaping the trajectory of my academic journey, and I am deeply appreciative of the opportunity to explore this subject under his guidance.

Furthermore, Dr. Sheard has not only excelled as a supervisor but has also demonstrated exceptional prowess as an educator in guiding me through the material of this thesis. His instructional sessions, meticulously tailored to elucidate the intricate concepts and theories pertinent to my research, consistently proved to be exceptional. Dr. Sheard provided insightful explanations and clarifications that significantly enriched my comprehension of the subject. Additionally, his module, "Geometry of Surfaces", serves as a notable testament to his excellence in academia. Through the adept use of animations, visual aids, and illustrative examples, Dr. Sheard skilfully conveyed the complexities of geometric concepts, fostering within me a profound intuition for the subject. His unwavering dedication to creating immersive and informative learning experiences has left an indelible mark on my academic journey. I am profoundly grateful for the privilege of learning from such an outstanding educator and for Dr. Sheard's invaluable contributions to my intellectual growth and development as a scholar.

Beyond his academic prowess, I am also indebted to Dr. Sheard for his pastoral mentorship and support. In addition to guiding me through the intricacies of research, Dr. Sheard has provided a supportive and nurturing environment in which I have been able to thrive both academically and personally. His genuine care and concern for my well-being have been a source of comfort and reassurance during challenging times, and I am deeply grateful for his unwavering support and encouragement. Dr. Sheard has not only been a mentor but also a trusted confidant and friend, and I am immensely fortunate to have had the opportunity to work with someone of such integrity, compassion, and wisdom.

Equally deserving of my profound appreciation is my partner, Hannah M. Butler, whose steadfast love, support, and understanding have been the bedrock of my journey throughout this research endeavour. Hannah has been a constant companion through the highs and lows, offering encouragement, empathy, and confidence in my abilities. Her consistent backing and empathy have endowed me with the fortitude and resilience essential for navigating the rigours of academic research while tending to personal commitments and obligations. Hannah's mere presence has served as both solace and inspiration, underscor-

ing the significance of nurturing personal and professional relationships in the pursuit of my aspirations. Her steadfast support, sacrifices, and unshakeable faith in my capabilities have served as a perpetual fount of motivation, propelling me to surpass my limits and pursue excellence in all endeavour. I am profoundly grateful for Hannah's unwavering love, support, and companionship, and I hold dear the privilege of embarking on this journey with her at my side. Meeting her has been a blessing, and her presence in my life is akin to a cherished gift that I am endlessly thankful for.

I would like to extend a heartfelt acknowledgement to my dear friend, Madalena de Gouveia Magalhães, whose unwavering support and friendship have been a cornerstone of my journey throughout this research project. Madalena has been more than just a confidant; she has been a constant source of encouragement, understanding, and inspiration. During moments of doubt or difficulty, Madalena has provided a listening ear and offered words of wisdom that have helped me navigate the challenges of academic pursuits with grace and determination. Her unwavering belief in my abilities and her genuine enthusiasm for my success have fuelled my motivation and reminded me of the importance of perseverance and resilience in the face of adversity. Beyond her role as a friend, Madalena has also contributed valuable insights and perspectives to my research, enriching the depth and breadth of my work. Whether through engaging discussions, brainstorming sessions, or simply sharing in the joys and frustrations of the research process, Madalena has been a trusted collaborator, whose input has been instrumental in shaping the trajectory of my research. Furthermore, Madalena's contagious passion for learning and her boundless curiosity have served as a constant source of inspiration, motivating me to push the boundaries of my own intellectual pursuits and embrace new challenges with enthusiasm and vigour. For all these reasons and more, I am profoundly grateful to Madalena for her unwavering friendship, support, and companionship throughout this research journey. I am immensely fortunate to have her by my side as I embark on this next chapter of my academic and personal growth.

Furthermore, I would like to acknowledge the contributions of, Julian Caliao, Hagieben Krishnamoorthy, Aryan Mangal, Arthur Aimone, Jan Olucha Fuentes and Sam Tomlin, who have supported me in various capacities, whether through discussions, feedback, or moral support. Your collective input has been invaluable in shaping the outcome of this research, and I am grateful for your generosity and kindness.

# Contents

# List of Figures

The author created all the figures in this thesis using TikZ, except for Figure 5.

# 1 Introduction

## 1.1 Historical background

The concept of decision problems was initially introduced by David Hilbert and Wilhem Ackermann in 1928 [22], known as the "Entscheidungsproblem" in German, which translates to 'decision problem'. This problem inquires whether there exists a systematic method capable, in principle, of determining the truth or falsity of any given mathematical statement.

Hilbert and Ackermann's decision problem forms part of their broader programme to formalise all of mathematics, aiming to provide a robust foundation for mathematical truths. They posited that every mathematical problem should be decidable, implying that we can determine the truth or falsity of a proposition using an *algorithm*. The Entscheidungsproblem represented a central aspect of this vision, suggesting that a universal method for solving mathematical problems would imply the completeness, consistency, and decidability of mathematics.

However, the resolution of the Entscheidungsproblem came with negative responses from mathematicians and logicians such as Alonzo Church, Alan Turing, and Kurt Gödel in the 1930s [11, The Rise and Fall of the *Entscheidungsproblem*]. Turing in particular, introduced the *Turing machine*, a theoretical model of computation, to argue against the existence of a universal algorithm for the Entscheidungsproblem. He illustrated this with the halting problem, which questions whether a program, given its description and input, will halt or run indefinitely [34]. Turing's proof of the halting problem's undecidability exposed the limitations of computation and laid the foundation for computability theory. Gödel's incompleteness theorem further influenced their work, asserting that in any sufficiently complex formal system capable of expressing arithmetic, there exist true statements unprovable within that system [16]. These seminal contributions delineated the boundaries of computability, revealing certain mathematical statements as inherently undecidable, as demonstrated by Gödel's incompleteness theorems and Turing's proof of the halting problem's undecidability.

Concurrently, in the field of group theory, Max Dehn proposed similar decision problems around 1911 [12], known as the 'word' and 'conjugacy' problem for groups. Dehn's problem inquired whether an algorithm could determine whether two given elements in a group are equal or conjugate respectively. Dehn anticipated the potential difficulty of these problems for a general group and thus focused on studying these problems for the fundamental groups of closed orientable surfaces of genus $g \geq 2$. Remarkably, he managed to resolve both the word and conjugacy problems for these groups in 1912 [13].

Significant progress has been made in addressing these problems. Following Dehn's pioneering work, his student Wilhelm Magnus laid the groundwork for further advancements [37]. Martin Greendlinger expanded the scope of groups for which the word problem is decidable by demonstrating the continued efficacy of Dehn's algorithm for a class of groups meeting certain *small cancellation conditions* [17]. In recent years, Roger C. Lyndon and Paul E. Schupp used geometric arguments to extend the decidability of both the word and conjugacy problems for groups satisfying specific small cancellation conditions [36]. Their contributions represent a significant step forward in understanding and resolving these problems in group theory.

The word problem for a general group presentation is undecidable, meaning there is no algorithm capable of deciding it for all groups. This was independently proven by Pyotr Novikov (1955) [44], William W. Boone (1959) [4], and John L. Britton (1958) [6]. Novikov's groundbreaking work drew from mathematical logic and computability theory, ingeniously encoding the halting problem of a Turing machine within a specific group to demonstrate the undecidability of the word problem. Meanwhile, Boone and Britton pursued alternative paths, employing techniques rooted in combinatorial group theory to construct groups with inherently undecidable word problems. The undecidability of the conjugacy problem was later established by Alexander A. Fridman [15] in 1969.

The significance of Dehn's problems lies not only in their direct challenge to computational group theory but also in their implications for algorithm theory and decidability. Mathematicians have endeavoured for decades to find general solutions to these problems, contributing to the advancement of several mathematical domains, including algebraic topology, combinatorial group theory, and algorithmic logic.

## 1.2 Structure of the thesis

This thesis explores the decidability of the word and conjugacy problems within specific groups, along with the algorithms used to address them. It begins by laying the groundwork in Section 2, defining fundamental concepts such as group presentations. Section 3 provides a geometric interpretation of the word problem using Cayley graphs. In Section 4, algorithms by Dehn for fundamental groups of closed orientable surfaces with genus $g \geq 2$ are discussed. Section 4.3 presents Van Kampen diagrams and their role in extending the decidability of the decision problem to groups satisfying the small cancellation condition. Lastly, Section 5 introduces the Novikov-Boone-Britton theorem, establishing the undecidability of the word problem. Additionally, Appendix A showcases Python algorithms the author has developed for addressing the word and conjugacy problems in free groups of finite rank.

## 1.3 Preliminaries

The terms *decision problem, algorithm*, *word* and *conjugacy* problems have been used casually, with some possessing intuitive understanding. We now provide formal definitions.

For the purposes of our discussion, we will employ the following definitions of algorithm. For a more precise definition consult [38].

**Definition 1.1.** An **algorithm** is a finite set of well-defined instructions designed to solve a problem or accomplish a specific task. It provides procedure that, when executed on an input, yields the answer 'yes' or 'no'.

**Definition 1.2** ([29, Page 7])**.** A **decision problem** is any arbitrary yes-or-no question on specified sets of inputs. It is classified as **decidable** if there exists an algorithm that provides either a 'yes' or 'no' answer; otherwise, it is deemed as **undecidable**.

Recall, for a group $G$ and a subset $X$ of $G$, the group $G$ is generated by $X$ if each element of $G$ can be represented as a finite concatenation of elements from $X$ and their inverses (with the empty product interpreted as the identity element of $G$). This principle implies the existence of a mapping $\pi : X \to G$ such that the image of $X$ under $\pi$ forms a generating set for $G$. Throughout our discussion, we will denote the concatenation of

elements in $X$ as *words*, and when we speak of the equality of words within the group $G$, we are referring to the image of the word under the mapping $\pi$ [31, Chapter I, Section 12]. We will provide a more formal definition of a word in Definition 2.2.

Now, we will present Dehn's "identity" and "transformation" problems as originally stated in [12], while referencing the English translation by John Stillwell in [55, On infinite discontinuous groups]. The first problem is articulated as follows.

> **The identity problem**. An element of the group is given as a product of generators. One is required to give a method whereby it may be decided in a finite number of steps whether this element is the identity or not.

Now, for the second problem.

> **The transformation problem**. Any two elements $S$ and $T$ of the group are given. A method is sought for deciding the question whether $S$ and $T$ can be transformed into each other, i.e. whether there is an element $U$ of the group satisfying the relation $S = UTU^{-1}$.

We will rephrase these decision problems using a modern perspective, with a particular emphasis on classes of groups known as *finitely generated* and *finitely presented*. These groups possess a finite set of generators and relators, as formally defined in Definition 2.27.

**Question 1.3** (Word problem)

Let $G$ be a group with a finite set of generators $X$. Does there exist an algorithm that decides, given two words in $X$ whether they are equal in $G$?

**Remark 1.4.** This problem is also known as the 'generalised word problem'.

It is crucial to highlight that the word problem is equivalent to Dehn's identity problem. This equivalence arises from the option to select a word $w = uv^{-1}$ for the identity problem, which asks if $uv^{-1} = 1$ in $G$ which is identical to asking whether $u = v$ holds in $G$, which is precisely the statement the word problem.

We will now reformulate the transformation problem to align with our terminology.

**Question 1.5** (Conjugacy problem)

Let $G$ be a group with a finite set of generators $X$. Does there exist an algorithm that decides, given two words in $X$ whether they are conjugate in $G$?

In fact, the decidability of the conjugacy problem implies the decidability of the word problem: if we can decide whether a word is conjugate to the word representing the identity element, then we can conclude that the word is equal to the identity in $G$. However, for the sake of completeness, we address both problems separately, as there are instances in certain groups where the conjugacy problem remains undecidable while the word problem is decidable [15].

# 2 Free groups and presentations

Our exploration begins with the study of free groups. In this preliminary stage, we lay the foundation by defining essential concepts necessary for the construction of a free group. For this, we will refer to [14, Pages 203–205].

## 2.1 Definitions

**Definition 2.1.** Let $X$ be a set whose elements will be called **letters**. We define the set of **inverse letters** to be $X^{-1} = \{x^{-1} : x \in X\}$ where $x^{-1}$ is merely a symbol.

We can think of the set $X^{\pm} = X \cup X^{-1}$ as an alphabet and use it to construct a free group. With this notion we must define words from the alphabet.

**Definition 2.2.** A **word** is a finite (possibly empty) string of letters from the alphabet $X^{\pm}$. We will use 1 to denote the **empty word**. Let $(X^{\pm})^*$ denote the set of words from the alphabet $X^{\pm}$.

**Example 2.3.** Let $X = \{x_1, x_2, x_3\}$ be a set of letters. Then, the set of inverse letters $X^{-1}$ would be $X^{-1} = \{x_1^{-1}, x_2^{-1}, x_3^{-1}\}$. Consider the alphabet $X^{\pm}$, a word from this alphabet could be constructed as follows:

$$w = x_1 x_2 x_3^{-1} x_1^{-1} x_2.$$

**Definition 2.4.** The **length** of a word $w$ is the number of letters in this word which we denote by $|w|$. We set the length of the empty word to 0.

Our objective is to ensure that these words satisfy the axioms of a group. In order to account for the simplification that occurs when adjacent inverse elements in a group interact, we consider expressions of the form $xx^{-1}$ and $x^{-1}x$ as equivalent to the identity element. To establish this notion rigorously, we introduce the concept of a *reduced* word.

**Definition 2.5.** A word in $(X^{\pm})^*$ is **reduced** if it contains no pair of consecutive letters of the form $xx^{-1}$ or $x^{-1}x$.

We lack a method for the derivation of reduced words. Consequently, we introduce a procedure to achieve this.

**Definition 2.6.** The **elementary reduction** of a word $w \in (X^{\pm})^*$ is the deletion of all pairs of consecutive letters of the form $xx^{-1}$ or $x^{-1}x$.

**Example 2.7.** The words

$$1,\ x_1x_2,\ x_{12}x_{32}^{-1}x_2^{-1}$$

are reduced, whereas the word

$$x_1x_5^{-1}x_5x_6$$

is not reduced.

By introducing the concepts of reduced words and word reduction, we can establish an equivalence relation that serves as the basis for constructing the operation associated with a free group.

**Definition 2.8.** For two words $w, w' \in (X^{\pm})^*$ we say that $w'$ is equivalent to $w$ if it can be obtained from $w$ after a finite number of insertions and deletions of $xx^{-1}$ or $x^{-1}x$.

**Proposition 2.9.** The relation defined in Definition 2.8 is an equivalence relation.

*Proof.* We check the axioms of an equivalence relation are satisfied.

*Reflexivity.* Let $w \in (X^{\pm})^*$. No insertions or deletions are required to obtain $w$ from itself.

*Symmetry.* Let $w \sim w'$. Consider the same sequence of insertions and deletions that transforms $w$ into $w'$. Since each operation is reversible (deleting $xx^{-1}$ can be reversed by inserting $xx^{-1}$ and vice versa), we can simply reverse the operations in the sequence which they were carried out.

*Transitivity.* Suppose $w \sim w'$ and $w' \sim v$. This means $w$ can be transformed into $w'$, and $w'$ can be transformed into $v$ through sequences of elementary reductions and insertions. By combining these two sequences of transformations, $w$ can be transformed into $v$. Therefore, $w \sim v$. □

It is important to observe that we have not specified a particular sequence for inserting or deleting pairs $xx^{-1}$ and $x^{-1}x$ when determining the equivalence of two words. For instance, consider the word:

$$w = xyxx^{-1}yx^{-1}xyx.$$

This word is equivalent to $xy^3x$, but we can arrive at this reduction by first deleting the pair $x^{-1}x$ and then the pair $xx^{-1}$, or vice versa. Similarly, we could alternatively insert pairs of these forms to $xy^3x$, in any order, and obtain $w$. Thus, the order in which we execute insertions and deletions to show two words are equivalent is inconsequential. This flexibility stems from the fact that there exists only one reduced form for any given word, a fact demonstrated in the subsequent proposition.

**Proposition 2.10** ([14, Proposition 7.17])
Any word $w \in (X^{\pm})^*$ is equivalent to a unique reduced word.

*Proof.* We prove the following.

*Existence.* We will use induction on the length of a word. For a word of length 0 and 1 the statement is trivially true. Now, suppose the statement is true for words of length $k$ and consider a word of length $k+1$, say $w = x_1 \cdots x_{k+1}$ where each $x_i \in X^{\pm}$. By the induction hypothesis, there exists a reduced word $v = y_1 \cdots y_n$ for $n \leq k$ with each $y_j \in X^{\pm}$ such that $x_2 \cdots x_{k+1} \sim v$. To prove $w$ is equivalent to a reduced word we consider the following cases: if $x_1 \neq y_1^{-1}$, then $x_1 v$ is inherently reduced, and consequently, $w \sim x_1 v$. Conversely, if $x_1 = y_1^{-1}$, we eliminate the pair $x_1 y_1$, yielding $x_1 v \sim y_2 \cdots y_n$. Given that $v$ was originally reduced, the resultant word $y_2 \cdots y_n$ is reduced thus, $w \sim y_2 \cdots y_n$. In both scenarios, $w$ is equivalent to a reduced word.

*Uniqueness.* Let $F(X)$ denote the set of reduced words in the set $X^{\pm}$. For each element $x \in X^{\pm}$, we introduce the mapping

$$\begin{aligned} L_x : F(X) &\to F(X) \\ y_1 \cdots y_k &\mapsto \begin{cases} xy_1 \cdots y_k & \text{if } x \neq y_1^{-1} \\ y_2 \cdots y_k & \text{if } x = y_1^{-1}. \end{cases} \end{aligned}$$

We also define $L_w$ for any word $w = x_1 \cdots x_n$ as the composition $L_{x_1} \circ \cdots \circ L_{x_n}$ and for the empty word 1, we define $L_1$ as the identity map id. Note that $L_x \circ L_{x^{-1}}$ is the identity

map for every $x \in X^{\pm}$, which means that applying $L_x$ and $L_{x^{-1}}$ in succession results in no change to the word. It follows that, $v \sim w$ implies $L_v = L_w$.

We prove by induction on the length that if $w$ is reduced, then $w = L_w(1)$. This statement is clearly true for words of length 0 and 1. Assuming its validity for reduced words of length $n$, consider a reduced word $w$ of length $n+1$. We can express $w = xu$, where $x \in X^{\pm}$, and $u$ is a reduced word of length $n$ that does not begin with $x^{-1}$, meaning $L_x(u) = xu$ and $L_1(u) = u$. Therefore, $L_w(1) = L_x \circ L_u(1) = L_x(u) = xu = w$.

To prove uniqueness, it suffices to show that if $w \sim v$ and both are reduced, then $w = v$. This follows from the fact that if $v \sim w$ then $L_v = L_w$, leading to $L_v(1) = L_w(1)$, and since $v_1 w$ are reduced then $v = L_v(1) = L_w(1) = w$ that is, $v = w$. □

We use the following proposition to show that the closure axiom in a group is maintained when performing multiplication with equivalent words.

**Proposition 2.11** ([2, Proposition 7.9.3])**.** Products of equivalent words are equivalent i.e. if $w \sim w'$ and $v \sim v'$ then $wv \sim w'v'$.

*Proof.* Consider the reduced forms of two words $w$ and $v$, denoted as $w_0$ and $v_0$ respectively. To determine the reduced form of the product $wv$, we first independently reduce $w$ to $w_0$ and $v$ to $v_0$. The product $wv$ is then initially transformed into $w_0 v_0$. If $w_0 v_0$ is already in its reduced form, our process is complete. However, if $w_0 v_0$ is not yet fully reduced, we continue to apply elementary reduction operations until we achieve a reduced word. If $w \sim w'$ and $v \sim v'$, the same process, when applied to $w'v'$ will proceed through the intermediate form $w_0 v_0$ so, it leads to the same reduced word. □

## 2.2 Construction of free group

In commonly encountered groups, elements exhibit specific relationships. For instance, in the cyclic group $C_n$, we observe that for an element $a \in C_n$ we have $a^n = 1$, and similarly, in the dihedral group $D_n$, a reflection has order of 2, so $b^2 = 1$. These relations among group elements are characteristic of many familiar groups.

However, in the context of free groups, a distinguishing feature is the absence of such inherent relationships between elements. This absence of constraints on element combinations gives rise to the term 'free'. This property enables more versatile and unrestricted group constructions.

**Definition 2.12.** By Proposition 2.10 we can identify the set $F(X)$ as the set $(X^{\pm})^*/\sim$. The **free group** over $X$ is the set $F(X)$ with the binary operation $*$ defined by $w * v$ is the unique reduced word equivalent to the word $wv$.

> **Remark 2.13.** Elements of the group $F(X)$ can be represented as equivalence classes denoted by $[w] = \{w' \in (X^{\pm})^*/\sim : w \sim w'\}$. For simplicity and convenience, we will often use the notation $w$ to refer to the equivalence class $[w]$ when denoting elements in $F(X)$.

Given our ability to associate words with elements of the free group, from now on we will use the term 'words' to refer to elements of the free group as well.

**Definition 2.14.** The cardinality of $X$ is called the **rank** of the free group $F(X)$. The trivial free group is a group with $X = \varnothing$ so, has rank 0.

Given the absence of inherent relationships between its elements, one might question whether a free group adheres to the closure axiom of a group. Having defined the term 'free group', we prove that this definition indeed gives rise to a group structure.

**Proposition 2.15** ([14, Page 205])
The free group over $X$ is a group.

*Proof.* We check the axioms of a group are satisfied.

*Identity.* We designate the empty word 1 as the identity element of the group $F(X)$. This choice is justified by the fact that when any word is concatenated with the empty word, it remains unchanged.

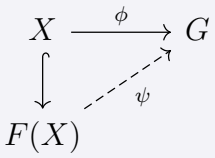
*Inverse.* The inverse of a reduced word $w = x_{i_1}^{\varepsilon_1} \cdots x_{i_k}^{\varepsilon_k}$, where $\varepsilon_h \in \{-1, 1\}$ and $x_{i_j} \in (X^{\pm})^*$, is expressed as

$$w^{-1} = x_{i_k}^{-\varepsilon_1} \cdots x_{i_1}^{-\varepsilon_1}.$$

*Associativity.* The operation $*$ is defined as concatenating words and reducing them to their unique reduced forms. Concatenation of strings is associative, and the reduction process leads to a unique outcome regardless of the order of concatenation. □

Free groups possess a universal property, which is a distinctive method of characterising an object or structure by their unique and fundamental interactions and relationships with other objects. This concept is illustrated below.

**Theorem 2.16** (Universal property of free groups [14, Proposition 7.21])
A map $\phi : X \to G$ from the set $X$ to a group $G$ can be extended group homomorphism $\psi : F(X) \to G$ in a unique way. Therefore, we have the following commutative diagram i.e. starting from any point and following the direction of the arrows leads to the same endpoint.




**Remark 2.17.** We can employ the universal property to provide an alternative definition of free groups: a group $F$ is said to be **free** over a subset $X \subseteq F$ if, for any group $G$ and any mapping $\phi : X \to G$, there exists a unique homomorphism $\psi : F \to G$ such that $\psi(x) = \phi(x)$ holds for all $x \in X$ [27, Chapter I, Definition 1]. This approach offers a succinct and elegant means of characterising free groups while accentuating their essential attributes.

*Proof.* We must prove that $\psi$ exists and is unique.

*Existence.* For any word $w \in F(X)$, given by $w = x_1^{\varepsilon_1} x_2^{\varepsilon_2} \dots x_k^{\varepsilon_k}$, we define the map $\psi : F(X) \to G$ as follows:

$$\psi(w) = \phi(x_1)^{\varepsilon_1} \phi(x_2)^{\varepsilon_2} \cdots \phi(x_k)^{\varepsilon_k}.$$

Now, consider another element $v \in F(X)$, represented as $v = y_1^{\eta_1} y_2^{\eta_2} \cdots y_j^{\eta_j}$. Applying $\psi$ to the product $wv$, we obtain:

$$\begin{aligned}\psi(wv) &= \psi(x_1^{\varepsilon_1} x_2^{\varepsilon_2} \cdots x_k^{\varepsilon_k} y_1^{\eta_1} y_2^{\eta_2} \cdots y_j^{\eta_j}) \\ &= \phi(x_1)^{\varepsilon_1} \phi(x_2)^{\varepsilon_2} \cdots \phi(x_k)^{\varepsilon_k} \phi(y_1)^{\eta_1} \phi(y_2)^{\eta_2} \cdots \phi(y_j)^{\eta_j} \\ &= \psi(w)\psi(v).\end{aligned}$$

Therefore, $\psi$ is a homomorphism.

*Uniqueness.* In pursuit of a contradiction, suppose there exists another homomorphism $\Psi : F(X) \to G$ which extends $\phi$ and, is distinct from $\psi$. By the nature of homomorphisms, for any element $w = x_1^{\varepsilon_1} x_2^{\varepsilon_2} \cdots x_k^{\varepsilon_k} \in F(X)$ it follows that

$$\begin{aligned}\Psi(w) &= \Psi(x_1^{\varepsilon_1})\Psi(x_2^{\varepsilon_2}) \cdots \Psi(x_k^{\varepsilon_k}) \\ &= \phi(x_1)^{\varepsilon_1} \phi(x_2)^{\varepsilon_2} \cdots \phi(x_k)^{\varepsilon_k} \\ &= \psi(w).\end{aligned}$$

Hence, $\psi$ is the unique homomorphism. □

In the following theorem, we establish a pivotal connection between the isomorphism of free groups and the cardinality of their generating sets.

**Theorem 2.18** ([49, 2.1.4.])**.** Let $F(X), F(Y)$ be free groups on $X, Y$ respectively. Then $F(X)$ is isomorphic to $F(Y)$ if and only if $|X| = |Y|$.

*Proof.* We prove each direction in turn.

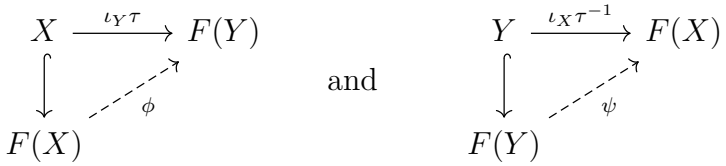
Proof of ($\Rightarrow$). (The proof of this implication is adapted from [14, Proposition 7.37]). Let $\Phi : F(X) \to F(Y)$ be an isomorphism and let us define the set $N_X = \langle w^2 : w \in F(X) \rangle$. It follows that $N_X$ is a normal subgroup of $F(X)$, because for any $w \in F(X)$ and $n \in N_X$, the conjugate $wnw^{-1}$ will also be in $N_X$. Indeed, for any $n \in N_X$, it is possible to express $n$ as $k^2$. Consequently, $wk^2w^{-1} = (wkw^{-1})^2$ belongs to $N_X$. The normality of $N_X$ allows us to construct the quotient group $F(X)/N_X$. Since $\Phi$ is an isomorphism it will map the entire subgroup $N_X$ to the normal subgroup generated by the squares of elements in $F(Y)$, which we denote by $N_Y$. This mapping induces an isomorphism

$$\begin{aligned}\Psi : F(X)/N_X &\to F(Y)/N_Y \\ [w] &\mapsto [\Phi(w)]\end{aligned}$$

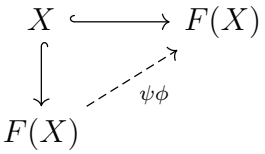
for all $w \in F(X)$. The existence of $\Psi$ implies that $F(X)/N(X)$ and $F(Y)/N(Y)$ have the same cardinality. The cardinality of $F(X)/N_X$ is determined by the number of distinct cosets of $N_X$ in $F(X)$. Since $N_X$ is generated by the squares of elements in $F(X)$, each coset corresponds to a unique square element in $N_X$. Therefore, the cardinality of $F(X)/N_X$ is equal to the number of distinct squares in $N_X$, which is the number of generators in $X$. Consequently, the cardinality of $F(X)/N_X$ is the same as the cardinality of the generating set $X$. Hence, $|X|$ must equal $|Y|$.

Proof of ($\Leftarrow$). Let $\iota_X : X \to F(X)$ and $\iota_Y : Y \to F(Y)$ be inclusion maps and, let $\tau : X \to Y$ be a bijection. The existence of such a map is guaranteed by the assumption that $|X| = |Y|$. We can construct the following maps $\iota_Y \tau : X \to F(Y)$ and

$\iota_X\tau^{-1} : Y \to F(X)$. By Theorem 2.16 we then have the following commutative diagrams




where $\phi$ and $\psi$ are homomorphisms. Therefore, $\psi\phi\iota_X = \psi\iota_Y\tau = \iota_X\tau^{-1}\tau = \iota_X$ i.e. the following diagram




commutes. However, the identity map $\mathrm{id}_{F(X)} : F(X) \to F(X)$ will also make this diagram commute; since the extending homomorphism is unique we must have $\psi\phi = \mathrm{id}_{F(X)}$. Using a similar argument yields $\phi\psi = \mathrm{id}_{F(Y)}$. Hence, $\phi : F(X) \to F(Y)$ is an isomorphism. □

**Remark 2.19.** This theorem tackles the isomorphism problem for free groups, which asks if there exists an algorithm capable of deciding if two given groups are isomorphic. In free group, of finite rank the algorithm entails comparing the cardinalities of their respective sets of generators.

Building on Theorem 2.18, for finite rank free groups we introduce a new notation $F_n$, for $n \in \mathbb{N} = \{0, 1, 2, \ldots\}$. Furthermore, the theorem emphasises that the isomorphism of free groups is solely dependent on the cardinality of their generating sets, not on the nature of the elements within these sets. This highlights a key characteristic of free groups: their structure is determined by the number of generators rather than their specific identities.

**Example 2.20.** According to Theorem 2.18, all free groups of rank 1 are indeed isomorphic. Specifically, these groups are abelian groups which are isomorphic to the group $\mathbb{Z}$.

Utilising the universal property, we introduce a pivotal result that lays the groundwork for our further investigation, particularly in the context of defining a *presentation* of a group.

**Corollary 2.21** (Von Dyck's theorem [14, Corollary 7.22])
Every group is the quotient of a free group.

*Proof.* Let $G$ be a group and set $X = G$. If $\phi : G \to G$ is the identity map then, by Theorem 2.16, $\phi$ can be extended to the unique homomorphism $\psi : F(G) \to G$. Hence, $G$ is isomorphic to $F(G)/\ker(\psi)$. □

The theorem underscores the universal nature of free groups, establishing them as the fundamental building blocks for characterising and understanding any group. This universality simplifies the classification of groups, enriching our understanding of the relationships between them.

Furthermore, we can employ this theorem to develop algorithms that decide decision problems within the quotient of a free group instead of the group itself. This approach may be more convenient, even though we do not intend to utilise this particular approach.

## 2.3 Presentations of groups

Group presentations are a foundational element of group theory, arising when there is a need to define a group using particular generators and the relationships among them. Group presentations serve as a powerful tool for succinctly describing a group's structure and properties by specifying both the set of generators and the connections between them.

**Definition 2.22** ([23, Definition 2.51])**.** If $R$ is a non-empty subset of a group $G$, the **normal closure** of $R$ is the intersection of all the normal subgroups of $G$ which contain $R$. We denote this by $\langle\langle R\rangle\rangle$.

Since the normal closure is the intersection of a collection of normal subgroups of a group, it is evident that it is itself a subgroup. Furthermore, it is normal, as all of its elements can be expressed in the form $\prod_{i=1}^{n} g_i r_i g_i^{-1}$ for some $g_i \in G$ and $r_i \in R$. In fact, we can write $\langle\langle R\rangle\rangle = \langle gRg^{-1} : g \in G\rangle$. With this understanding, we are now prepared to define the presentation of a group.

**Remark 2.23.** In fact $\langle\langle R\rangle\rangle$ is the smallest normal subgroup containing $R$.

**Definition 2.24** ([23, Definition 2.51])**.** Let $G$ be a group. A **presentation** for $G$ is a pair $\langle X \mid R\rangle$ where $X$ is a subset of $G$ and $R$ is a set of words in $X$ such that

$$G \cong F(X)/\langle\langle R\rangle\rangle.$$

The elements of $X$ are the generators of $G$, while elements $r \in R$ are called **relators**, and equations of the form $r = 1$ referred to as **relations**.

**Remark 2.25.** Occasionally, for the sake of convenience, we may adopt a shorthand notation and express a group $G = \langle X \mid R\rangle$.

Consequently, these presentations allow us to represent any element within $G$ as a finite concatenation $x_1 x_2 \cdots x_n$, with each $x_i$ representing an element of $X^{\pm}$. A significant implication of this result is that any word $w$ in $F(X)$ maps to the identity element in $G$ under the map $F(X) \to F(X)/\langle\langle R\rangle\rangle$ if and only if $w$ belongs to $\langle\langle R\rangle\rangle$.

The choice of defining a presentation as $F(X)/\langle\langle R\rangle\rangle$ is a natural one. Initially, $F(X)$ represents a free group generated by elements in $X$, devoid of non-trivial relations among its generators, allowing for unrestricted element generation from $X$, except the relation $xx^{-1} = x^{-1}x = 1$, mandated to fulfil the group axioms. By quotienting out $\langle\langle R\rangle\rangle$ we effectively incorporate these specified relations into the unconstrained group. Each element in $\langle\langle R\rangle\rangle$ becomes the identity element in our presentation, thus defining the relations among the generators. As a result, $F(X)/\langle\langle R\rangle\rangle$ produces the group that emerges from applying these relations to $F(X)$, where $X$ serves as the generators, ensuring the preservation of the relations defined in $R$.

By Corollary 2.21, every group is isomorphic to the quotient of a free group. However, a fundamental question arises: can every group be described through a presentation? The answer is yes, as we will demonstrate in the following proposition.

**Proposition 2.26** ([26, Chapter 4, Proposition 1])**.** Every group has a presentation.

*Proof.* Let $G$ be a group, and let $X$ be a subset of $G$. Suppose $X$ generates $G$ with the map $\phi : X \to G$ such that $\phi(x) = x$ for all $x \in X$. We can use Theorem 2.16 to extend the homomorphism $\phi$ to a homomorphism $\psi : F(X) \to G$. Therefore, $G$ is isomorphic to $F(X)/\ker(\psi)$. Given that $\ker(\psi)$ is a normal subgroup of $F(X)$, we can select a subset $R \subset \ker(\psi)$ such that $R$ normally generates $\ker(\psi)$, making $R$ the smallest normal subgroup of $F(X)$ containing $R$. Consequently, $G$ is isomorphic to $F(X)/\langle\langle R \rangle\rangle$. □

Now that presentations have been defined, we can proceed to define a specific type of presentation.

**Definition 2.27.** Let $G$ be a group.

- We say $G$ is **finitely generated** if there exists a presentation $\langle X \mid R \rangle$ for $G$ such that $X$ is a finite set.
- We say $G$ is **finitely presented** if there is a presentation $\langle X \mid R \rangle$ for $G$ with both $X$ and $R$ finite sets.

**Example 2.28**

Here we present some examples of presentations.

1. The dihedral group $D_n$ of order $2n$ can be presented as $\langle a, b \mid a^2 = b^2 = (ab)^n \rangle$, where $n \geq 2$; furthermore, by defining $x = ab$, we obtain an alternative presentation:
$$\langle a, x \mid a^2 = x^n, a^{-1}xa = x^{-1} \rangle,$$
illustrating that presentations of groups are not unique. [49, Example of Presentations, Page 51].
2. A presentation for the free group of rank $n \in \mathbb{N}$ generated by the set $X$ is commonly denoted as $\langle X \mid \varnothing \rangle$, although some texts use $\langle X \mid \rangle$.
3. The group $G$ generated by $a$ and $b$, with relators defined as
$$c_n = x_n^{-1} b x_n b,$$
where $x_n = a^{-(w_n-2)} b a^{w_n-2}$ and $w_n = 2^{(2^n)} + 1$ for $n \in \mathbb{N} \setminus \{0\}$, is not a finitely presented group [43, Theorem 13].

## 2.4 Tietze transformations

We can perform operations on presentations, a set of manipulations known as *Tietze transformations* [56]. These transformations, also commonly referred to as Tietze moves, are named in honour of the German mathematician Heinrich Tietze, renowned for his substantial contributions to the fields of group theory and topology. Tietze transformations entail elementary alterations of a group presentation which preserve the isomorphism class of the group defined by the presentation. This section follows [26, Pages 41–47].

**Definition 2.29.** For a group with the presentation $\langle X \mid R \rangle$, the subsequent operations are referred to as the **Tietze transformations**.

(T1). Adjoin a relator:

$$\langle X \mid R \rangle \to \langle X \mid R \cup \{r\} \rangle$$

where $r \in \langle\langle R \rangle\rangle \setminus R$.

(T2). Remove a relator:

$$\langle X \mid R \rangle \to \langle X \mid R \setminus \{r\} \rangle$$

where $r \in R \cap \langle\langle R \setminus \{r\} \rangle\rangle$.

(T3). Adjoin a generator:

$$\langle X \mid R \rangle \to \langle X \cup \{y\} \mid R \cup \{y^{-1}w\} \rangle$$

where $y \notin X$ and $w \in F(X)$.

(T4). Remove a generator:

$$\langle X \mid R \rangle \to \langle X \setminus \{y\} \mid R \setminus \{y^{-1}w\} \rangle$$

where $y \in X, w \in \langle X \setminus \{y\} \rangle$ and $y^{-1}w \in R$ is the only element of $R$ involving $y$.

**Remark 2.30.** Notice that T1 and T2, as well as T3 and T4, are mutually inverse transformations.

In Tietze transformation T3, it may initially appear unusual that we need to include the relation $y^{-1}w$, but it serves a crucial purpose in specifying how the new generator $y$ interacts with the existing generators. Essentially, it provides a rule that allows us to replace $y$ with $w$ when $y$ appears alongside other generators in a product. This ensures not only the consistency of $y$ within the group's existing structure but also preserves the closure property of the group. To illustrate this concept further, consider a group with the following presentation

$$\langle a \mid a^4 \rangle.$$

Now, suppose we introduce a new generator $y$ to the group. If we do not include the relation $y^{-1}w$ in the presentation, the new group presentation would appear as follows:

$$\langle a, y \mid a^4 \rangle.$$

However, this presentation lacks information about how $y$ interacts with the existing generator $a$, leading to ambiguity in defining group operations involving $y$.

By including the relation $y^{-1}a^2$, the presentation becomes:

$$\langle a, y \mid a^4, y^{-1}a^2 \rangle.$$

Now, we have explicitly specified that $y$ behaves as if it were equal to $a^2$ when involved in group operations. For example, when calculating $ya$, we can apply the relation as follows:

$$ya = a^2a = a^3.$$

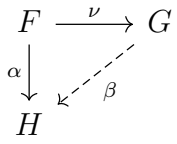
With Tietze transformations defined, we now begin by stating two lemmas essential for proving how these operations preserve a group's isomorphism class.

**Lemma 2.31** ([26, Chapter 4, Lemma 1])**.** Let $F, G, H$ be groups, and let $\nu : F \to G$ and $\alpha : F \to H$ be homomorphisms such that

1. $\mathrm{Im}(\nu) = G$ and,
2. $\ker(\nu) \subseteq \ker(\alpha)$.

Then there exists a homomorphism $\beta : G \to H$ such that $\beta \circ \nu = \alpha$.

*Proof.* We intend to establish the existence of a homomorphism $\beta$ that ensures the commutativity of the following diagram:




To do so, we start by selecting $g \in G$, and then we choose $f \in F$ such that $\nu(f) = g$. We define $\beta(g) = \alpha(f)$. It is crucial to note that because the image of $\nu$ precisely covers $G$, we are guaranteed to find a pre-image for any $g$. Moreover, the second condition implies that $\beta$ is well-defined:

$$\begin{aligned} \nu(f') = \nu(f) &\iff \nu(f')(\nu(f))^{-1} = 1_G \\ &\iff f'f^{-1} \in \ker(\nu) \\ &\implies f'f^{-1} \in \ker(\alpha) \\ &\implies \alpha(f') = \alpha(f). \end{aligned}$$

The value of $\alpha(f)$ remains consistent, regardless of the choice of $f \in \nu^{-1}(g)$ (in this context, $\nu^{-1}(g)$ denotes the pre-image of $g$ under $\nu$).

Now, to establish that $\beta$ is indeed a homomorphism, let us consider $g, g' \in G$ and $f, f' \in F$ such that $\nu(f) = g$ and $\nu(f') = g'$. Since $\nu$ is a homomorphism, we have $\nu(ff') = gg'$, and it follows that:

$$\begin{aligned} \beta(gg') &= \alpha(ff') && \text{(by the definition of } \beta) \\ &= \alpha(f)\alpha(f') && \text{(as } \alpha \text{ is a homomorphism)} \\ &= \beta(g)\beta(g'). \end{aligned}$$

Finally, for each $f \in F$, it is important to note that $f \in \nu^{-1}(\nu(f))$. This implies that the value of $\beta$ applied to $\nu(f)$ is precisely $\alpha(f)$. Therefore, we have $\beta \circ \nu = \alpha$. □

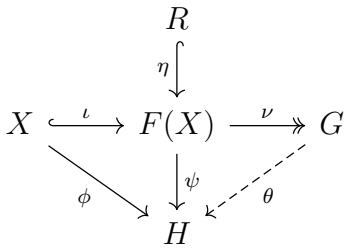
**Lemma 2.32** ([26, Chapter 4, Proposition 3])**.** Suppose we are given $G = \langle X \mid R \rangle$, a group $H$ and a mapping $\phi : X \to H$. Then $\phi$ extends to a homomorphism $\theta : G \to H$ if and only if, for every relator $r \in R$ the result of replacing every occurrence of $x \in X$ with $\phi(x)$ yields the identity of $H$.

**Note 2.33.** This condition ensures that the way the generators relate to each other in $G$ is mirrored by their images in $H$ under the map $\phi$. If this mirroring does not occur, then $\phi$ cannot be a homomorphism, because a homomorphism must preserve the group structure.

**Remark 2.34.** This resembles the assertion of Theorem 2.16 but applies to a general group that is not free. Due to $H$ not being free thus, containing relations, the conclusion drawn in this lemma is less conclusive compared to that of Theorem 2.16.

*Proof.* We prove each direction separately.

Proof of ($\Leftarrow$). Consider $G$ with the presentation $\langle X \mid R\rangle$. We consider the following commutative diagram:




Here, both $\nu$ and $\psi$ are canonical maps derived from Theorem 2.16. Next, we can rephrase the substitution condition as $\eta(R) \subseteq \ker(\psi)$. Since $\ker(\psi)$ is a normal subgroup of $F(X)$, and $\langle\langle R\rangle\rangle = \ker(\nu)$, this condition is equivalent to $\ker(\nu) \subseteq \ker(\psi)$. Both conditions stated in Lemma 2.31 are fulfilled. Therefore, by applying the lemma, we can extend $\psi$ to a homomorphism $\theta$.

Proof of ($\Rightarrow$). The existence of such a $\phi$ implies that $R \subseteq \langle\langle R\rangle\rangle \subseteq \ker(\nu) \subseteq \ker(\nu \circ \theta) = \ker(\psi)$. □

**Proposition 2.35** ([26, Chapter 4, Proposition 5])**.** Let $G = \langle X \mid R\rangle$ and suppose that $w, r \in F(X)$ with $r \in \langle\langle R\rangle\rangle \setminus R$. If $y$ is a symbol not in $X$, then both the canonical maps

$$X \to \langle X \mid R \cup \{r\}\rangle$$
$$X \to \langle X \cup \{y\} \mid R \cup \{y^{-1}w\}\rangle$$

extend to isomorphism with domain $G$.

*Proof.* By Lemma 2.32, these mappings extend to homomorphisms of $G$, denoted as

$$\phi : \langle X \mid R\rangle \to \langle X \mid R \cup \{r\}\rangle$$
$$\psi : \langle X \mid R\rangle \to \langle X \cup \{y\} \mid R \cup \{y^{-1}w\}\rangle.$$

To establish the isomorphism of $\phi$ and $\psi$ it suffices to identify their inverses. Notably, the mappings

$$X \to \langle X \mid R\rangle$$
$$X \cup \{y\} \to \langle X \mid R\rangle$$

also extend to homomorphisms:

$$\phi' : \langle X \mid R \cup \{r\}\rangle \to \langle X \mid R\rangle$$
$$\psi' : \langle X \cup \{y\}\rangle \to \langle X \mid R\rangle.$$

In fact, $\phi'$ and $\psi'$ serve as the inverses of the corresponding mappings of $\phi$ and $\psi$ respectively. It is evident that the composition of these mappings in either direction with their respective inverses results in the identity map. □

We have shown that group presentations are not unique, and a single Tietze transformation suffices to modify a given presentation while preserving the group's isomorphism. Consequently, employing a finite sequence of Tietze transformations enables us to relate two distinct presentations. We illustrate this relation in the following theorem.

**Theorem 2.36** (Tietze's theorem [36, Chapter II, Proposition 2.1])
Two finite presentations $\langle X \mid R\rangle$ and $\langle Y \mid Q\rangle$ define isomorphic groups if and only if they are related by a finite sequence of Tietze transformations.

*Proof.* We prove each direction in turn.

Proof of ($\Rightarrow$). Consider a group $G$ and let us define two mappings: $\alpha : F(X) \to G$ and $\beta : F(Y) \to G$. Without loss of generality, we suppose that $X$ and $Y$ are disjoint sets. However, if they are not disjoint, we can relabel the overlapping generators in order to make the sets disjoint. Let $Z = X \cup Y$ and define the map

$$\begin{aligned} \phi : Z &\to G \\ z &\mapsto \begin{cases} \alpha(z) & \text{if } z \in X \\ \beta(z) & \text{if } z \in Y \end{cases} \end{aligned}$$

By Theorem 2.16 this map extends to a homomorphism $\psi : F(Z) \to G$. For each element $x \in X$, we choose the word $w_x \in F(Y)$ such that $\psi(x) = \psi(w_x)$, and we let $S_1$ be the finite set of words $s_x = x^{-1}w_x$. Similarly, for each $y \in Y$ choose $w_y \in F(X)$ such that $\phi(y) = \phi(w_y)$, and define $S_2$ to be the finite set of words $s_y = y^{-1}w_y$. To incorporate all the elements from $Y$ into the presentation $\langle X \mid R\rangle$, we use a sequence of T3 transformations. This process is finite because $Y$ is a finite set. As a result, we transition from the presentation $\langle X \mid R\rangle$ to $\langle Z \mid R \cup S_2\rangle$, as the T3 transformations introduce relations in the form $y^{-1}w_y$, which precisely correspond to the set $S_2$. Consequently, we can deduce that $\ker(\psi) = \langle\langle R \cup S_2\rangle\rangle$. Given the definitions of $Q$ and $S_2$, it becomes evident that $\psi(Q \cup S_1) = 1$ and that $Q \cup S_1 \subseteq R \cup S_2$. Therefore, by using a finite sequence of Tietze transformations, we can incorporate elements from $Q \cup S_1$ one by one into $R \cup S_2$, resulting in a presentation $\langle Z \mid P\rangle = \langle R \cup Q \cup S_1 \cup S_2\rangle$. We have established that we can transition from $\langle X \mid R\rangle$ to $\langle Z \mid P\rangle$ through a finite sequence of Tietze transformations. By employing a similar symmetrical approach, we can also move from $\langle Y \mid Q\rangle$ to $\langle Z \mid P\rangle$. Consequently, this implies that we can pass from $\langle X \mid R\rangle$ to $\langle Y \mid Q\rangle$ using a finite sequence of Tietze transformations.

Proof of ($\Leftarrow$). By Proposition 2.35 a single Tietze transformation between group presentations preserves the isomorphism class. Therefore, by repeatedly applying this argument, we can establish the validity of the statement. □

## 2.5 Decidability of the decision problems

We are now prepared to solve the word and conjugacy problems in free groups and obtain an interesting result regarding the decidability of these decision problems in finitely presented groups.

**Theorem 2.37** ([37, Corollary 1.2.1.])
The word problem is decidable in a finite rank free group.

*Proof.* For this proof, we will examine the free group of finite rank with the presentation $\langle X \mid \varnothing \rangle$. Later, as demonstrated in Theorem 2.42, we will establish that analysing this presentation alone is sufficient.

For a free group with rank $n = 0$, there are no generators, and the group consists only of the identity element. In this case, the word problem is trivial, and any word $w$ represents the identity element. However, when $n \geq 1$ as demonstrated by Proposition 2.10, every word in the group is equivalent to its unique reduced form. Therefore, to solve the word problem, we construct the following algorithm:

1. given a word $w$ in $F_n$, apply the process of elementary reduction to obtain its unique reduced form, denoted as $w'$;
2. verify if the reduced word $w'$ is equal to the identity word in $F_n$.

Since the length of the input word $w$ is finite, the process of elementary reduction will eventually terminate, ensuring that the algorithm will terminate. □

Before we delve into the examination of the conjugacy problem, it is essential to introduce the concept of *cyclically reduced* elements within free groups.

**Definition 2.38.** A word is **cyclically reduced** if it is reduced and the first and last letter of $w$ are not inverses of each other.

**Remark 2.39.** Equivalently, a word is cyclically reduced if it is reduced and if all of its cyclic permutations are also reduced.

Cyclically reduced elements play a pivotal role in the development of our algorithm, as they have specific properties that significantly influence our understanding of conjugacy.

**Lemma 2.40** ([36, Chapter I, Proposition 2.14])**.** Let $F(X)$ be a free group. We have the following.

1. Every word $w \in F(X)$ is conjugate to a cyclically reduced word.
2. Two cyclically reduced words are conjugate if and only if they are cyclic permutations of each other.

*Proof.* We prove each statement separately.

1. Let $w \in F(X)$ represented as $w = x_1 x_2 \cdots x_n$. Firstly, if $w$ is already cyclically reduced, there is nothing more to prove as it is already conjugate to itself, with the identity element as the conjugating element. Now, let $g = x_{n-k} \cdots x_n$ be the longest terminal word such that $x_n = x_1^{-1}, x_{n-1} = x_2^{-1}, \ldots, x_{n-k} = x_{k+1}^{-1}$. Consequently, we can write $w$ as $w = g^{-1} v g$ where $v = x_{k+2} \cdots x_{n-k-1}$ is a cyclically reduced word.
2. Suppose we have two conjugate elements $w$ and $w' = g^{-1} w g$. To establish the validity of this statement, we shall employ induction on the length of $g$. We begin with the base case where $|g| = 0$. Here, $g$ must be the identity element so, the words are equal. Now, let us assume that the statement holds for all $|g| \leq k$. We assume that $w$ is a cyclically reduced word and $|w'| = |w|$. Given that $w' = g^{-1} w g$, it is not

a reduced word and, consequently, not cyclically reduced since its length exceeds that of $w$. Therefore, there must be a cancellation between $g^{-1}$ and $w$ or between $w$ and $g$, but not both. We clarify this statement by examining the scenario where $g = q_1 \cdots q_m$ and $w = w_1 \cdots w_n$. We can write

$$w' = (q_m^{-1} \cdots q_1^{-1}) w_1 \cdots w_n (q_1 \cdots q_m),$$

any cancellation occurring simultaneously at both the beginning and end of $w$ would imply that $w_1 = q_1$ and $w_n = q_1^{-1}$. However, such an implication contradicts the assumption that $w$ is a cyclically reduced word.

If there is cancellation between $g^{-1}$ and $w$, we can express $g$ as $g = yd$ and $w$ as $w = yu$, where $y \in (X^{\pm})^*$ and $|d| = k$. We can represent $w'$ as follows:

$$w' = g^{-1} w g = (yd)^{-1}(yu)(yd) = d^{-1} y^{-1} yu(yd) = d^{-1}(uy)d.$$

It is important to note that $d^{-1}(uy)d$ is a cyclic permutation of $w$ because both $d$ and $uy$ are reduced words. In the second case, we can establish that $w'$ is a cyclic permutation of $w$ using a similar argument. Therefore, the result follows by induction. □

**Theorem 2.41** ([36, Chapter II, Proposition 2.14])
The conjugacy problem is decidable in a finite rank free group.

*Proof.* As discussed in the proof of Theorem 2.37, it is only necessary to consider the presentation of a free group, given by $\langle X \mid \varnothing \rangle$.

For the case where $n = 0$, the proof mirrors that found in Theorem 2.37. On the other hand, for $n \geq 1$, given two words $w, v \in F_n$, we construct the following algorithm to check if they are conjugate.

1. If $w$ and $v$ are not already in their cyclically reduced forms, apply the reduction process to obtain $w'$ and $v'$.
2. Examine whether $w'$ and $v'$ are cyclic permutations of each other, as determined by Lemma 2.40:
   - if $w'$ and $v'$ are cyclic permutations, then $w$ and $v$ are conjugate in $F_n$;
   - if $w'$ and $v'$ are not cyclic permutations, then $w$ and $v$ are not conjugate in $F_n$.

Each step of the algorithm will terminate since the words have finite length hence, the overall algorithm will halt. □

In the appendix, specifically in Appendix A, we present Python implementations of the algorithms.

We have examined the decidability of decision problems in free groups with presentation $\langle X \mid \varnothing \rangle$. The following theorem demonstrates that the choice of presentation is inconsequential in the decidability of the decision problems.

**Theorem 2.42** ([36, Chapter II, Proposition 2.2])
Let $\langle X \mid R \rangle$ be a finite presentation of a group $G$ and, let $\langle Y \mid Q \rangle$ be another finite presentation of the group $G$. We have the following.

1. If the word problem is decidable for $\langle X \mid R \rangle$ then it is decidable for $\langle Y \mid Q \rangle$.
2. If the conjugacy problem is decidable for $\langle X \mid R \rangle$ then it is decidable for $\langle Y \mid Q \rangle$.

*Proof.* We will prove the statements separately. The core approach of this proof involves treating the elements within $F(Y)$ as if they were elements of $F(X)$ and subsequently applying the algorithm available for $F(X)$, leveraging our knowledge of their decidability in that context.

1. Consider a function $\phi : Y \to F(X)$ such that for each $y \in Y$, we have $\phi(y)$ representing the same element of $G$ as $y$. We can extend the map $\phi$ to a homomorphism $\widetilde{\phi} : F(Y) \to F(X)$ with the same property. Since $Y$ is finite, we have a finite table of values for $\phi(y)$. Therefore, we also have a finite table of values for $\widetilde{\phi}(w)$ for $w \in F(Y)$. To decide if two given elements $u$ and $v$ of $F(Y)$ represent the same element of $G$, we need to evaluate $\widetilde{\phi}(u)$ and $\widetilde{\phi}(v)$ and apply the algorithm to decide the word problem for the presentation $\langle X \mid R \rangle$ to check if $\widetilde{\phi}(u) = \widetilde{\phi}(v)$. Since we can compute $\widetilde{\phi}(u)$ and $\widetilde{\phi}(v)$ using the finite tables of values, and the word problem is decidable for the presentation $\langle X \mid R \rangle$, we can determine whether $u$ and $v$ represent the same element in $G$.
2. The proof is analogous to the one for the word problem but applies the algorithm to decide the conjugacy problem for the presentation $\langle X \mid R \rangle$ instead. □

The results we have established demonstrate that the decidability of the decision problems we are studying is an intrinsic property of the group itself and is not contingent on the specific presentation chosen for the group.

# 3 Cayley graphs

In this section, our focus shifts to the conversion of groups into geometric constructs referred to as *Cayley graphs* [7]. We will utilise Cayley graphs as instruments to delve into whether the word problem is decidable.

We will explore the relationship between words and paths in Cayley graphs, highlighting how closed paths signify a decidable word problem. Leveraging this connection, we will explore the constructability of an infinite Cayley graph, proposing that if it can be completed in finite time, it indicates a decidable word problem.

## 3.1 Review of graph theory

We begin this section by reviewing some fundamental concepts in graph theory. We will refer to [32, Pages 53–61] for establishing key definitions and principles.

**Definition 3.1.** A **graph** is a pair $\Gamma = (V, E)$ of disjoints sets where $E$ is a set of subsets of $V$ that contains exactly two elements, i.e.

$$E \subset V = \{e : e \subset V \text{ and } |e| = 2\}.$$

The elements of $V$ are the **vertices**, the elements of $E$ are the **edges** of $\Gamma$.

In essence, graphs provide an alternative perspective on symmetric relations and are typically employed as a means to represent these relations.

**Definition 3.2.** Let $\Gamma = (V, E)$ be a graph.

- We say that two vertices $u, v \in V$ are **neighbours** or **adjacent** if they are joined by an edge i.e. if $\{u, v\} \in E$.
- The number of neighbours of a vertex is the **degree** of this vertex.

**Example 3.3.** We define the set of vertices as $V = \{1, 2, 3, 4, 5\}$, and the set of edges as $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}\}$. The graphical representations of the resulting graph $\Gamma = (V, E)$ can be observed in Figure 1.

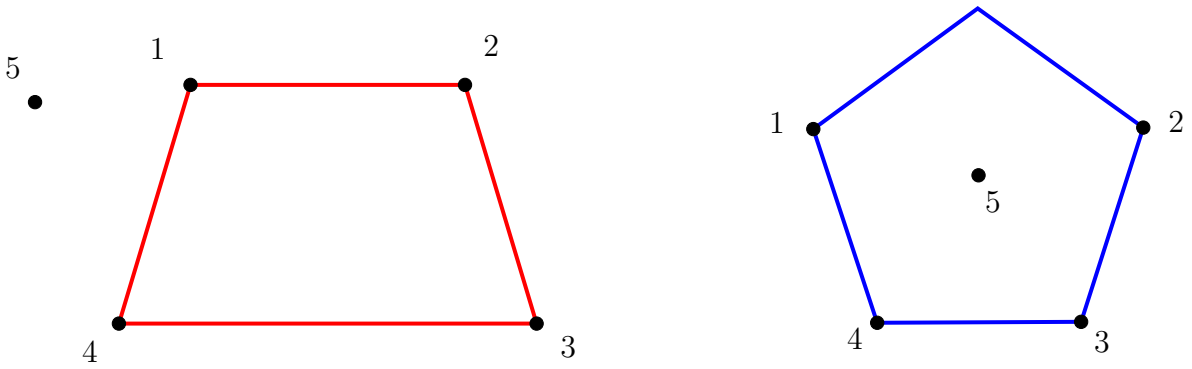



Figure 1: Illustration of two graphs with the same defining vertices and edges.

It is important to note that different graphical representations may seem distinct but ultimately convey the same underlying graph structure, underscoring the inherent combinatorial nature of graphs.

We introduce a set of geometric terminology specifically designed for the representation and analysis of graphs.

**Definition 3.4.** Let $\Gamma = (V, E)$ be a graph

- Let $n \in \mathbb{N} \cup \{\infty\}$. A **path** in $\Gamma$ of length $n$ is a sequence $v_0, \ldots, v_n$ of different vertices $v_0, \ldots, v_n \in V$ with the property that $\{v_j, v_{j+1}\} \in E$ for all $j \in \{0, \ldots, n-1\}$. If $n < \infty$ then we say this path **connects** the vertices $v_0$ and $v_n$.
- The graph $\Gamma$ is called **connected** if every pair of its vertices can be connected by a path in $\Gamma$.

**Example 3.5.** Examining the graph presented in Figure 2, we observe that it is a connected graph.

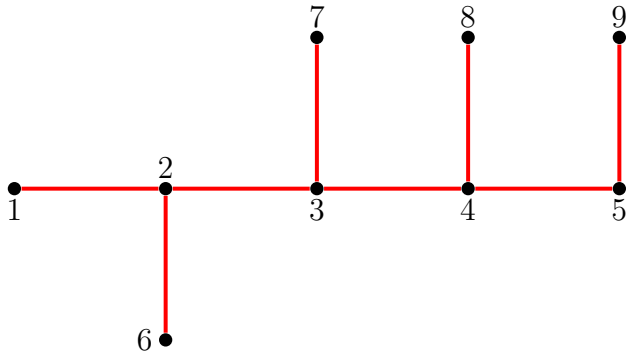



Figure 2: A connected graph.

Furthermore, the sequence $1, 2, 3$ forms a path, whereas the sequence $3, 7, 8$ does not constitute a path.

## 3.2 Cayley graphs

Building upon our comprehension of graph theory, we are now ready to formally define a Cayley graph.

**Definition 3.6.** Let $G$ be a group and let $X \subset G$ be a generating set of $G$. The **Cayley graph** of $G$ with respect to the generating set $X$, is the graph $\mathrm{Cayley}(G, X)$ whose

- set of vertices is $G$, and whose
- set of edges is $\{\{g, gx\} : g \in G, x \in (X^{\pm}) \setminus \{1\}\}$.

**Example 3.7.** In Figure 3, we illustrate a finite section of the Cayley graph for the integers, generated by the element 1. It is important to note that in this context, 1 does not represent the identity element, as the identity element for integers is 0.

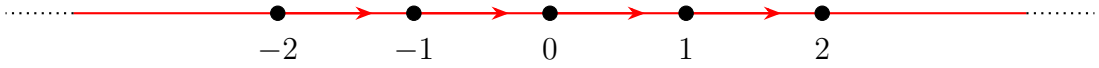



Figure 3: $\mathrm{Cayley}\,(\mathbb{Z}, \{1\})$.

Now, in Figure 4, we introduce the same Cayley graph, but this time, the generating set consists of $\{1, 3\}$.

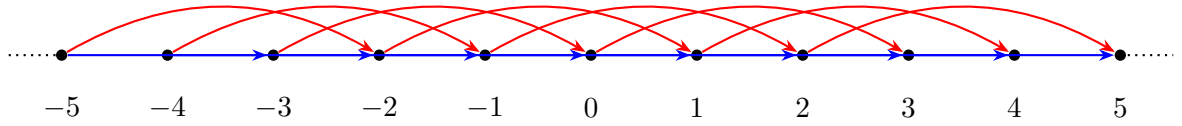



Figure 4: Cayley($\mathbb{Z}, \{1, 3\}$).

**Example 3.8**

In Figure 5, we present a finite section of the Cayley graph associated with the group $F_2$. The absence of group relations in $F_2$ leads to a pattern that resembles a fractal.

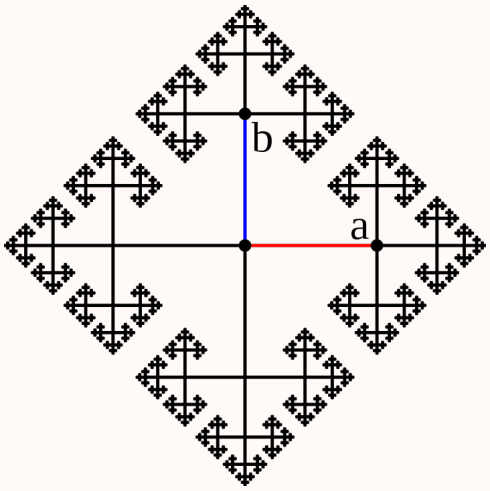



Figure 5: Cayley($F_2, \{a, b\}$).
Image: Dbenbenn (CC0).

The Cayley graph of $F_2$ with generating set $\{a, b\}$ can be described as a tree, which is a connected graph that does not have any cycles. In this context, a cycle is defined as a path $v_0, v_1, \ldots, v_{n-1}$ within a graph $\Gamma = (V, E)$, with $v_{n-1}, v_n \in E$, and this condition remains valid for all $n \in \mathbb{N}$ with $n > 2$.

### 3.2.1 Words and paths

The primary focus of this thesis has revolved around the study of words. In this context, we will examine the representations of words within a Cayley graph, following the discussion presented in [40, Pages 37–39].

Consider a group $G$ generated by the set $X$. For any given word $w \in (X^{\pm})^*$, there is a corresponding path within the Cayley graph Cayley$(G, X)$. This path initiates from the vertex that represents the identity element and subsequently traverses the edges of Cayley$(G, X)$ following the sequence dictated by $w$.

**Example 3.9.** For instance, consider the group $\mathbb{Z}^2$ generated by $x = (1, 0)$ and $y = (0, 1)$. In this context, the word $w = xxyxy^{-1}xy^{-1}xyx$ represents the path depicted in Figure 6.

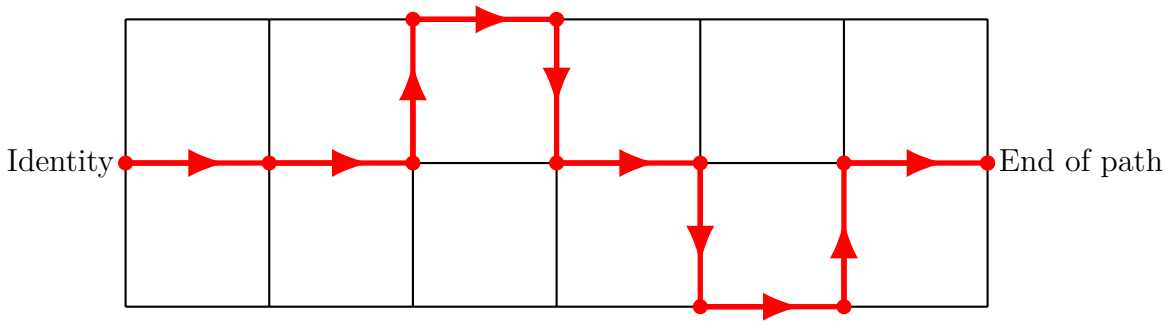



Figure 6: The path $p_w$ associated to the word $w = xxyxy^{-1}xy^{-1}xyx$.

We will denote the path described by a word $w$ as $p_w$. Therefore, given a word $w$, the path $p_w : [0, 1] \to \text{Cayley}(G, X)$ will connect the vertex corresponding to the identity with the vertex corresponding to the group element obtained by computing the product of $w$ in terms of $G$.

Conversely, every finite edge path in a Cayley graph $\text{Cayley}(G, X)$ corresponds to a word in the generators and their inverses. To retrieve the word, one reads the labels of the edges being traversed and appends an inverse if they move in the opposite direction of the edge orientation. Thus, it becomes evident that for a given group $G$ and a finite generating set $X$, there exists a one-to-one correspondence between:

1. finite paths in the Cayley graph $\text{Cayley}(G, X)$ that originate from the vertex representing the identity, and
2. words in $(X^{\pm})^*$.

## 3.3 Distance in graphs

Before discussing the word problem in a Cayley graph, it is necessary to first establish a concept of distance within the graph.

**Definition 3.10** ([46, 2.4])**.** Let $\Gamma = (V, E)$ be a connected graph. The **distance** between two vertices $u, v \in V$ the length of the shortest path joining.

In the definition above, it is necessary for $\Gamma$ to be connected in order to ensure the presence of paths connecting two points. By setting the length of the edges comprising these paths to 1, we establish that the length is a non-negative integer, thus ensuring the existence of a shortest-length path.

This distance definition remains applicable in the context of a Cayley graph. As the vertices of a Cayley graph are in correspondence with group elements, we have the ability to establish a concept of distance between vertices in the Cayley graph, thereby defining a distance function over the group's set of elements. This can be entirely articulated in terms of group elements, without explicit reference to the Cayley graph, by utilising the concept known as the *word metric*.

**Definition 3.11** ([40, Page 163])**.** Let $G$ be a group. The map

$$\begin{aligned} d : G \times G &\to \mathbb{N} \\ (g, h) &\mapsto \left|g^{-1}h\right|. \end{aligned}$$

is called the **word metric**.

This definition is the most natural: if $w \in (X^{\pm})^*$ is a word representing $g^{-1}h$, then

$$g^{-1}h = w \Rightarrow h = gw \quad \text{in } G.$$

If we commence at the vertex that represents $g$, the word $w$ denotes the path within the Cayley graph necessary to reach the vertex representing $h$. Consequently, the length of $g^{-1}h$ corresponds precisely to the length of the shortest path between $g$ and $h$.

**Proposition 3.12**
The word metric is a metric.

*Proof.* Let $G$ be a group generated by $X$. To demonstrate that $d$ satisfies the axioms of a metric, we will select arbitrary elements $g, h, k \in G$.

*Non-negativity.* We have $d(g,h) = |gh^{-1}|$ which represents the length of some element in $G$. This length is always non-negative.

*Non-degeneracy.* Suppose $d(g,h) = 0$. Then $|g^{-1}h| = 0$, and the only element in $G$ with length zero is 1. Therefore, $g^{-1}h = 1$ which implies $g = h$. Conversely, if $g = h$, then $d(g,h) = d(g,g) = |g^{-1}g| = |1| = 0$.

*Symmetry.* First, note that if $|g| = n$ then $|g^{-1}| = n$. Therefore, we can prove symmetry as follows:

$$d(g,h) = \left|g^{-1}h\right| = \left|\left(g^{-1}h\right)^{-1}\right| = \left|h^{-1}g\right| = d(h,g).$$

*Triangle inequality.* Let $d(g,h) = n$ and $d(h,k) = m$. Then, there exist some $x_i, y_i \in X^{\pm}$ such that:

$$g^{-1}h = x_1 \cdots x_n \quad \text{and} \quad h^{-1}k = y_1 \cdots y_m.$$

Then, by associativity

$$\begin{aligned} g^{-1}k = g^{-1}(hh^{-1})k &= (g^{-1}h)(h^{-1}k) \\ &= (x_1 \cdots x_n)(y_1 \cdots y_m). \end{aligned}$$

Hence, $d(g,k) \leq n + m = d(g,h) + d(h,k)$. □

## 3.4 The word problem

In Section 3.2.1, we demonstrated a one-to-one correspondence between words within the generating set of a group and paths within the Cayley graph, commencing from the vertex associated with the identity element. Given this correspondence, we can restate the word problem as presented in the following proposition.

**Proposition 3.13** ([40, Proposition 5.7])**.** Given a group $G$ and a finite generating set $X$, the word problem is decidable if and only if one can decide which words $w$ correspond to closed paths $p_w$ in the Cayley graph.

*Proof.* The discussion in Section 3.2.1 reveals that each word

$$w = x_{i_1}^{\varepsilon_1} \cdots x_{i_k}^{\varepsilon_k}$$

where $\varepsilon_h \in \{-1, 1\}$ and $x_{i_j} \in X^{\pm}$, defines a path from the vertex representing the identity in the Cayley graph, denoted as $v_{\text{id}}$, to the vertex representing the end of the word,

denoted as $v_w$. By traversing the labels $x_{i_1}, \ldots, x_{i_k}$ sequentially, following or opposing the direction of the arrows based on whether the exponent is $+1$ or $-1$, it becomes evident that $w = 1$ if and only if the path forms a closed loop. □

This proposition also addresses cases involving paths not originating from the identity, as we can establish a path from the identity to the base vertex of any such closed paths. For instance, consider a closed path denoted by $h$ representing a sequence of letters, originating from an arbitrary vertex say $v_h$, which does not represent the identity. We can construct a path from the vertex representing the identity, $v_{\text{id}}$, to $v_h$, representing a word, denoted as $g$. Combining these two paths forms another closed path. Traversing this combined path from $v_{\text{id}}$ following its orientation leads us to $v_h$. Given $h$ is a closed path, we return to $v_h$, and traversing $g$ in the opposite orientation brings us back to $v_{\text{id}}$. Consequently, the word $ghg^{-1} = 1$ in the group. This process is illustrated in Figure 7, where we illustrate the traversal of the path $p_{ghg^{-1}}$ (in green).

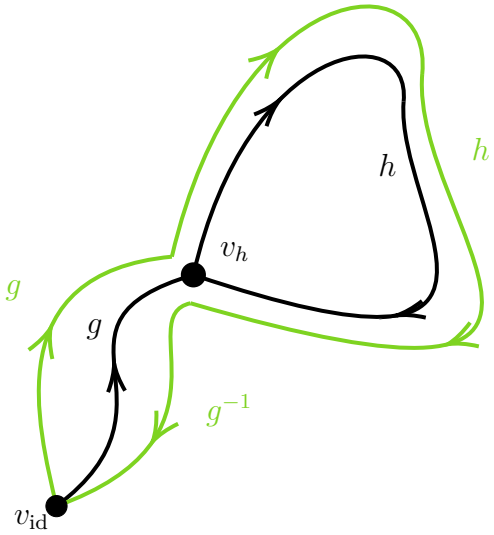



Figure 7: The traversal of the path $p_{ghg^{-1}}$.

Proposition 3.13 demonstrates that the word problem can be solved in groups where the Cayley graph is finite. However, as we explore infinite Cayley graphs, exemplified by free groups, we encounter new challenges. To establish the decidability of the word problem in groups with infinite Cayley graphs, we need to develop a deeper understanding of how to navigate these unbounded structures. Thus, it becomes necessary to establish a precise definition of when it is feasible to assert that a Cayley graph *can be constructed*. To achieve this, a reasonable criterion would be the ability to construct Cayley graph components of arbitrarily large size. The concept of 'arbitrarily large components' is precisely encapsulated by the notion of *balls* and *spheres* within a Cayley graph.

**Definition 3.14** ([40, Definition 9.9])**.** Given $g \in G$ and $n \in \mathbb{N}$ we define the following:

- the **sphere** of radius $n$ is $\mathcal{S}_n(g) = \{h \in G : d(g, h) = n\}$;
- the **ball** of radius $n$ is $\mathcal{B}_n(g) = \{h \in G : d(g, h) \leq n\}$.

**Remark 3.15.** In a connected graph $\Gamma$, the concepts of spheres and balls are entirely equivalent. However, instead of relying on the word metric, we use the distance as defined in Definition 3.10.

**Example 3.16.** Consider the group $\mathbb{Z}^2$ generated by the set $X = \{(0,1),(1,0)\}$. In Figure 8, we illustrate the sphere $\mathcal{S}_2((0,0))$ and the ball $\mathcal{B}_2((0,0))$ for the Cayley graph of $\mathbb{Z}^2$ with respect to $X$.

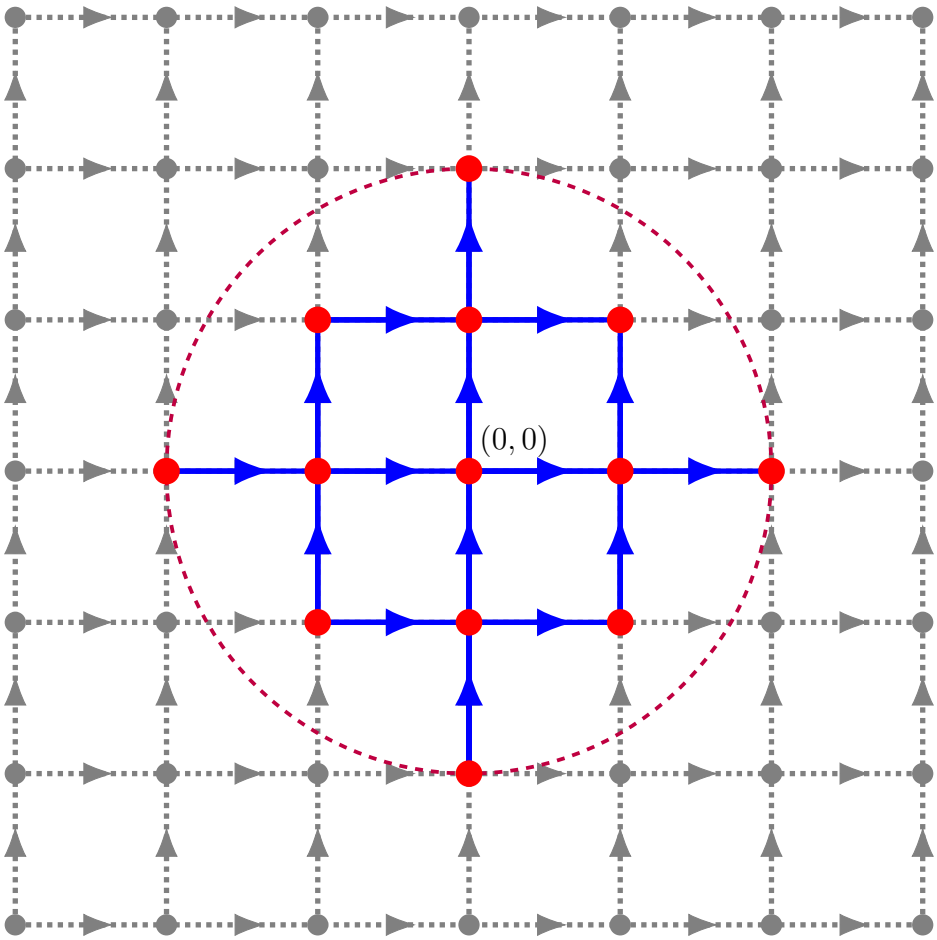



Figure 8: The ball and sphere of radius 2 for Cayley$(\mathbb{Z}^2, \{(1,0),(0,1)\})$.

The larger vertices (in red) represent the points on the sphere $\mathcal{S}_2((0,0))$, while the combination of solid arrows (in blue) and larger vertices (in red) collectively represents the ball $\mathcal{B}_2((0,0))$. Additionally, a dashed circle (in purple) is included to illustrate the 'shape' of the ball.

We are now prepared to establish the precise definition of a *constructible* Cayley graph.

**Definition 3.17.** A Cayley graph is **constructible** if given $n \in \mathbb{N}$ one can construct $\mathcal{B}_n(1)$ in a finite amount of time.

**Theorem 3.18** ([40, Theorem 5.10])
Let $G$ be a group with finite generating set $X$. Then the word problem is decidable if and only if the Cayley graph Cayley$(G, X)$ is constructible.

**Remark 3.19.** Theorem 3.18 presents an alternative proof for Theorem 2.42. Since the construction of the Cayley graph of $G$ depends on the chosen generating set, and given that there may be multiple generating sets for $G$, it follows that various Cayley graphs of $G$ can be constructed. However, Theorem 3.18 holds for any chosen generating set of $G$. In other words, the word problem is decidable for any finite presentation of $G$ with the generating set corresponding to the one chosen for constructing the Cayley graph.

*Proof.* We will prove both implications separately.

Proof of ($\Rightarrow$). We will use induction on $n$ to demonstrate that the ball $\mathcal{B}_n(1)$ in the Cayley graph is constructible. The ball $\mathcal{B}_0(1)$ consists of a single vertex corresponding to the identity element. This trivially exists, and thus, $\mathcal{B}_0(1)$ is constructible. Assume that $\mathcal{B}_n(1)$ has been constructed for some $n \geq 0$, our goal is to construct $\mathcal{B}_{n+1}(1)$.

Let $v$ be a vertex in $\mathcal{B}_n(1)$. Each vertex in $\text{Cayley}(G, X)$ is incident with exactly $k$ edges labelled by generators $X^{\pm}$. Thus, $v$ is already incident with $k$ edges labelled by $X^{\pm}$. To construct $\mathcal{B}_{n+1}(1)$, we need to add an oriented edge labelled $x$ to $v$ for each $x \in (X^{\pm}) \setminus \text{LABELS}_n(v)$, where $\text{LABELS}_n(v)$ is the subset of $X^{\pm}$ corresponding to edges attached to $v$ in $\mathcal{B}_n(1)$.

We must determine which vertices are connected to $v$ by these missing edges. To do this, we use an ordered list of vertices in the sphere $\mathcal{S}_n(1) = \{v_1, v_2, \ldots, v_l\}$. For each $v_i$, we associate a word $w_i$ representing the path from 1 to $v_i$ in $\mathcal{B}_n(1)$. For $v_1$, we list the generators in $(X^{\pm}) \setminus \text{LABELS}_n(v_1) = \{x_1, x_2, \ldots, x_m\}$. We check if $w_1 x_i w_j^{-1} = 1$ in $G$ for each $j$. If this condition is met, the edge labelled $x_i$ connects $v_1$ to $v_j$.

For $i > 1$, we consider three cases for the edges connected to missing labels $x_j$:

1. if $w_i x_j w_m^{-1} = 1$ in $G$, the edge connects $v_i$ to another vertex in $\mathcal{S}_n(1)$;
2. if $w_i x_j (w_m \sigma_n)^{-1} = 1$ in $G$ for some $\sigma_n$, the edge connects $v_i$ to a vertex in $\mathcal{S}_{n+1}(1)$ that is joined to $v_m$ in $\mathcal{S}_n(1)$, where $m < i$;
3. if none of the above conditions hold, a new vertex needs to be added in $\mathcal{B}_{n+1}(1)$.

Since any vertex in $\mathcal{B}_{n+1}(1)$ is connected to a vertex in $\mathcal{B}_n(1)$, this process will eventually construct all vertices within radius $n + 1$. Therefore, $\mathcal{B}_{n+1}(1)$ is constructible.

Proof of ($\Leftarrow$). Consider an arbitrary word $w \in (X^{\pm})^*$ of length $n$. Given the constructibility of $\text{Cayley}(G, X)$, we are able to construct a subgraph $\mathcal{B}_n(1)$ within $\text{Cayley}(G, X)$. Within the ball $\mathcal{B}_n(1)$, we can identify a path $p_w$ that corresponds to the word $w$. Let us focus on the endpoint of the path $p_w$, which we denote as $v_{\text{id}}$. If $v_{\text{id}}$ indeed coincides with the identity element 1 of the group $G$, then it follows that $w = 1$ in $G$. Conversely, if $v_{\text{id}}$ does not correspond to the identity element 1 in $G$, then $w \neq 1$ in $G$. □

# 4 Dehn's algorithms

In this section, we explore the algorithms provided by Dehn in [13] for the fundamental group of a closed orientable surface of genus $g \geq 2$.

We commence by revisiting the definition of the fundamental group of a closed orientable surface with genus $g \geq 0$. Drawing upon this, we motivate a presentation of these groups, demonstrating their adherence to specific small cancellation conditions. Through the lens of Van Kampen diagrams, we explore the extension of Dehn's algorithms to tackle the word problem beyond his original scope. Additionally, we highlight recent advancements in determining the decidability of the word and conjugacy problem in groups that meet certain small cancellation criteria.

## 4.1 The fundamental group

We provide an informal overview of the fundamental group, focusing on its intuitive essence rather than detailed formalisms. While the formal constructions are not necessary to grasp the underlying concepts, familiarity with the general idea of this group proves advantageous. For a comprehensive and rigorous treatment, readers are encouraged to consult [20, Chapter 1].

The fundamental group serves as a tool for understanding the presence of loops within a given topological space. It gauges the degree to which loops in the space can be continuously deformed or contracted to a single point.

To provide a more precise definition, let $X$ be a topological space with a specified base point $x_0 \in X$. The fundamental group $\pi_1(X, x_0)$ consists of equivalence classes of loops based at $x_0$. Two loops are considered equivalent if one can be continuously transformed into the other while keeping the base point fixed — a notion known as homotopy.

We illustrate some fundamental groups of common surfaces.

**Example 4.1.** Consider the circle $S^1$, where the fundamental group $\pi_1(S^1)$ encompasses loops originating from any point on the circle. The defining characteristic of a loop on $S^1$ is its 'winding' around the circle, which represents the number of times the loop encircles or revolves around the circle. It quantifies the rotational motion of the loop around the circle's perimeter. Loops that complete a full rotation around the circle, yet cannot be contracted to a point, are regarded as non-trivial. On the contrary, loops that do not wind around the circle can be continuously contracted to a point and are therefore considered trivial. Consequently, the fundamental group of $S^1$ is isomorphic to the group $\mathbb{Z}$. Each integer represents the number of rotations a loop makes around the circle, with positive values indicating anticlockwise rotation and negative values indicating clockwise rotation. In Figure 9(a), we observe a loop with a winding number of $-1$, while in Figure 9(b), a loop with a winding number of $-2$ is depicted.

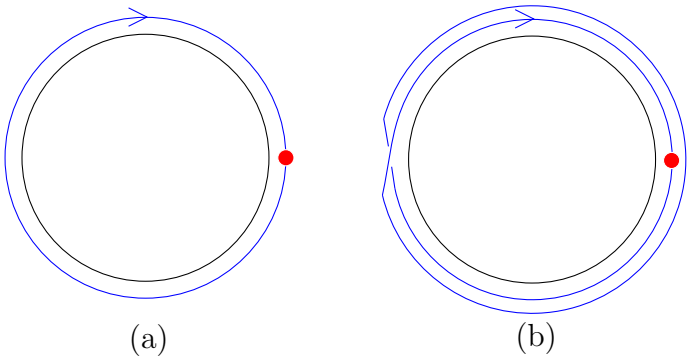



Figure 9: Two loops on a circle exhibiting distinct winding numbers.

**Example 4.2.** On the sphere $S^2$, every loop is contractible, signifying that any loop on the surface can be contracted to a single point without exiting the sphere, as depicted in Figure 10. Consequently, the fundamental group of the sphere is trivial: $\pi_1(S^2) = \{1\}$.

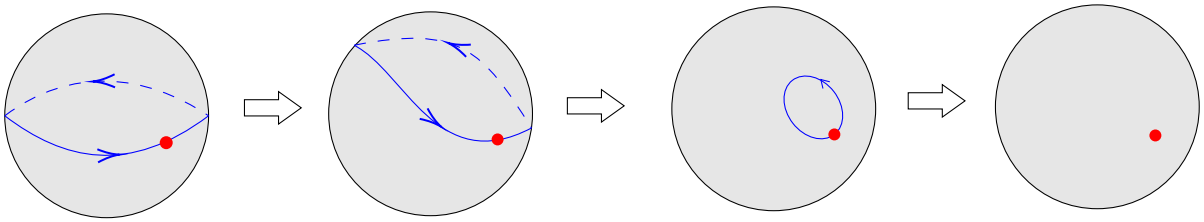


Figure 10: An illustration showing a loop on a sphere being contracted to a point.

**Example 4.3.** In contrast, the torus $\mathbb{T}^2$ has non-trivial loops encircling its aperture. Trivial loops on the torus are those that can be continuously contracted to a single point without departing from the surface, corresponding to the identity element in $\pi_1(\mathbb{T}^2)$. Non-trivial loops, on the other hand, traverse around the hole of the torus and cannot be contracted to a point within the torus. Each non-trivial loop denotes a distinct element in $\pi_1(\mathbb{T}^2)$. We depict the possible loops occurring on $\mathbb{T}^2$ in Figure 11.

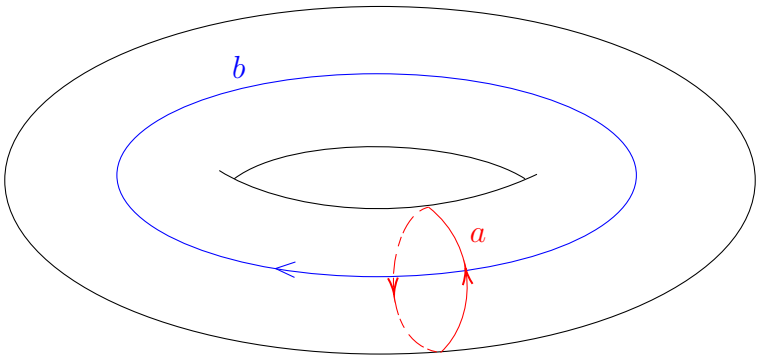



Figure 11: An illustration of a $\mathbb{T}^2$ with loops $a$ and $b$.

Remarkably, $\pi_1(\mathbb{T}^2) \cong \mathbb{Z} \times \mathbb{Z}$. This isomorphism stems from examining loops on the torus that encircle both its aperture and its body. These loops can be uniquely described by pairs of integers, where each integer signifies the winding number along a particular direction.

Geometrically, the identity problem concerning the fundamental group $\pi_1(X, x_0)$ revolves around identifying whether a closed loop, originating at $x_0$, can be contracted to a single

point. Conversely, the conjugacy problem for $\pi_1(X, x_0)$ involves establishing whether two closed loops are unbased homotopic. That is, whether one loop can be continuously deformed into the other while allowing for movement of their respective base point. This concept is depicted in Figure 12.

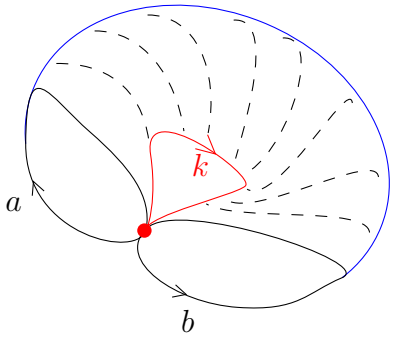



Figure 12: Homotopic loops $a$ and $b$.

Both loops $a$ and $b$ originate from the base point depicted by the dot (in red). We can continuously deform $a$ into $b$ while the base point moves along the path outlined by the loop $k$, resulting in the equation $a = kbk^{-1}$.

**Remark 4.4.** The unlabelled arc (blue) in Figure 12 facilitates the visualisation of the homotopy of the loops, and is not representative of a path.

We aim to justify why the presentation of fundamental groups for closed orientable surfaces with genus $g$ is typically represented as:

$$\langle a_1, b_1, \ldots, a_g, b_g \mid [a_1, b_1] \cdots [a_g, b_g] \rangle,$$

where the commutator $[a_i, b_i] = a_i b_i a_i^{-1} b_i^{-1}$. To exemplify this, we will examine the case of a torus and a closed orientable surface with genus 2. The logic applied in these cases can be generalised to closed orientable surfaces with genus $g \geq 2$.

Recall that closed orientable surfaces can be depicted using topological polygons. In the case of $\mathbb{T}^2$, this representation entails a square where each side is labelled by its generators and their inverses, as illustrated in Figure 13.

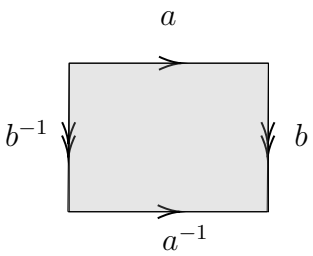



Figure 13: The torus represented by a topological rectangle.

We claim that this rectangle has only one topological vertex. To illustrate this, examine the vertex located at the starting point of $a$. As we traverse clockwise, this vertex corresponds successively to the end of $a^{-1}$, the beginning of $b^{-1}$, the end of $b$, the beginning of

$a^{-1}$, the end of $a$, and the beginning of $b$. This sequence returns to the endpoint of $b^{-1}$, precisely matching the starting vertex. We can designate this vertex as the base vertex for all the loops occurring on the torus.

In Figure 11, we depict the different types of loops that can manifest on $\mathbb{T}^2$. As previously mentioned, the torus has a single vertex, designated as the base point for our loops, which for our purposes we can situate at the intersection of the two depicted loops. Imagine traversing the surface of the torus along the paths indicated by these loops. What occurs is a return to the initial base point, as depicted in Figure 14.

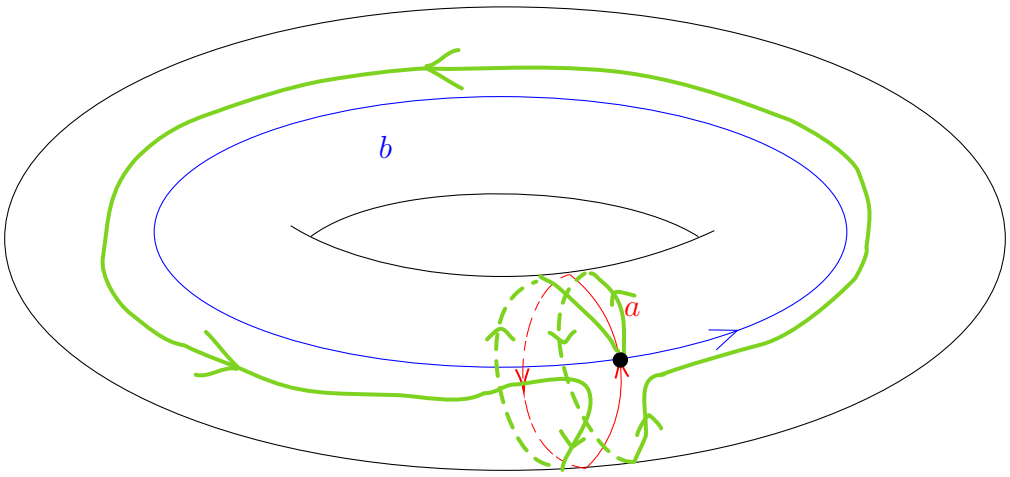



Figure 14: A traversal of the torus, shown in green, in accordance to the paths $a$ and $b$.

We can further demonstrate this phenomenon using the square depicted in Figure 15.

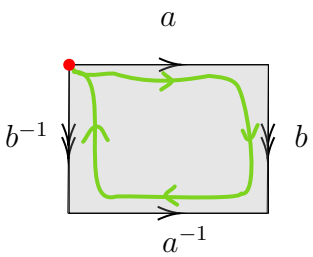



Figure 15: Traversal of the torus in its corresponding topological rectangle.

Algebraically, this process within the group $\pi_1(\mathbb{T}^2)$ can be described by the equation $aba^{-1}b^{-1} = 1$. Hence, a presentation for $\pi(\mathbb{T}^2)$ is given by $\langle a, b \mid [a, b] \rangle$.

We can do a similar process for a closed orientable surface of genus 2. We create such a surface by starting with a torus and performing a specific operation. This operation involves cutting a disc from the torus and then attaching a copy of this disk along the cut. This process effectively doubles the number of holes in the surface, resulting in a surface of genus 2. Figure 16 visually demonstrates this step-by-step construction.

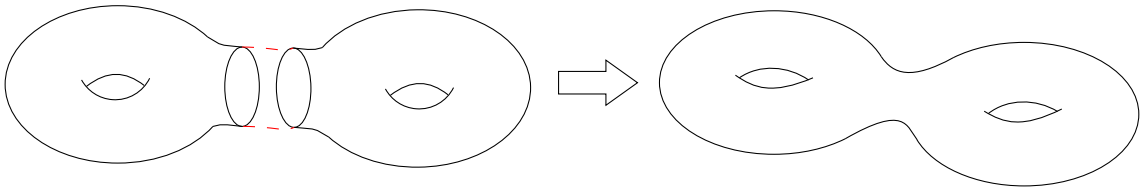


Figure 16: An illustration depicting the 'surgical' and 'gluing' process used to construct a closed orientable surface of genus 2.

To depict a closed orientable surface of genus 2 with a polygon, we follow the procedure outlined in Figure 17. By cutting a disc within the torus and designating one of the vertices of the square as its base point — illustrated in Figure 17(c) — we proceed to 'cut' along this loop. This cutting results in a pentagon. Upon gluing a copy of this surface to itself along the loop's cut, we attain the final polygon, which takes the form of an octagon.

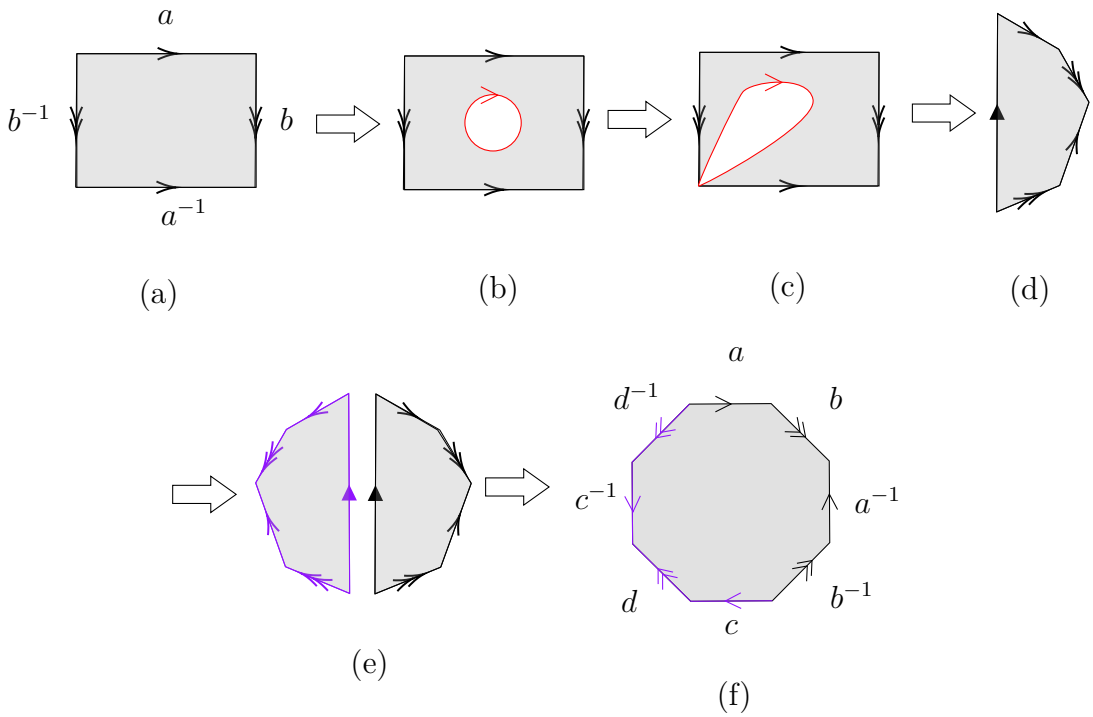



Figure 17: The 'surgery' and 'gluing' process in terms of topological polygons.

We assert that this octagon possesses only one topological vertex. To demonstrate this, consider the vertex located to the left of $a$. When traversing clockwise, this vertex corresponds to the end of $a^{-1}$, which in turn connects to the beginning of $b^{-1}$, subsequently linking to the end of $b$, and so forth. Eventually, this sequence loops back to the beginning of $a$.

Designate the sole vertex of this surface as the base point for all of its loops. By employing a similar procedure as we did with the torus, if we envision traversing this surface from the base point, we ultimately return to the initial position. We showcase the loops we traverse in Figure 18. Algebraically, this process equates to $aba^{-1}b^{-1}cdc^{-1}d^{-1} = [a,b][c,d] = 1$, leading to the conclusion that a presentation for the fundamental group of a closed orientable surface with genus 2 is given by $\langle a, b \mid [a,b][c,d] \rangle$.

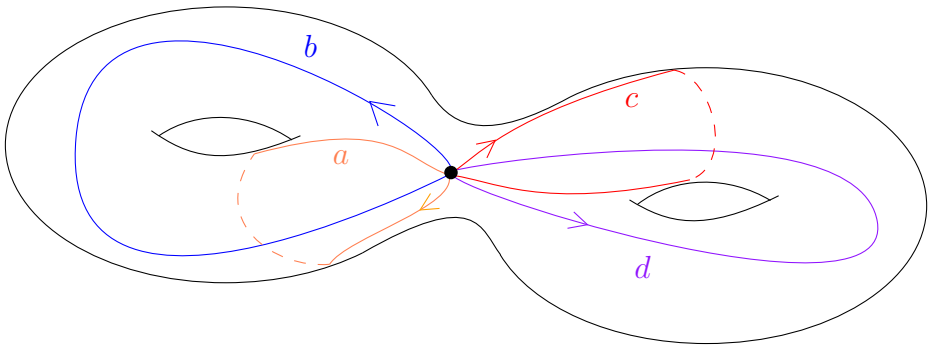



Figure 18: A closed orientable surface of genus 2 depicted alongside loops $a$, $b$, $c$, and $d$.

Similar reasoning can be applied to closed orientable surfaces of genus $g \geq 2$, revealing that a presentation for their fundamental group is given by

$$\langle a_1, b_1, \ldots, a_g, b_g \mid [a_1, b_1] \cdots [a_g, b_g] \rangle.$$

The rigorous derivation of these presentations typically relies on the Seifert-Van Kampen theorem [20, Theorem 1.20]. However, delving into this proof extends beyond the scope of this thesis.

## 4.2 Small cancellation hypotheses

The fundamental groups of closed orientable surface of genus $g \geq 1$ belong to a category of groups known to satisfy the *small cancellation conditions*.

In this section, we establish the defining traits of small cancellation groups. These groups conform to one or more of the following small cancellation hypotheses. However, before defining these conditions, we must first establish some additional technical definitions.

**Definition 4.5** ([39, Section 1.2.1])**.** Let $G$ be a group with presentation $\langle X \mid R \rangle$. Given a set of relations $R$ its **symmetrisation** $R_*$ is a set of all cyclic permutations of the relations $r \in R$ and their inverse.

**Example 4.6.** In a group represented by $\langle a, b, c \mid abc^2 \rangle$, the set of relators $R$ is defined as $\{abc^2\}$. Consequently, the symmetrised set is given by:

$$R_* = \{abc^2, bc^2a, c^2ab, c^{-2}b^{-1}a^{-1}, b^{-1}a^{-1}c^{-2}, a^{-1}c^{-2}b^{-1}\}.$$

**Definition 4.7** ([39, Section 1.2.1])**.** A word $b$ is called a **piece** (with respect to $R$) if there are not two distinct elements $r_1, r_2 \in R_*$ with the common beginning $b$, that is $r_1 = bc_1$ and $r_2 = bc_2$.

We can now outline the small cancellation conditions. Groups satisfying any of these criteria will be categorised as 'small cancellation groups'. The first condition is as follows.

**Definition 4.8** ([39, Definition 1.7])**.** Let $\lambda$ be a positive real number. A set of relations $R$ is said to satisfy the $C'(\lambda)$ condition if

$$|u| < \lambda\,|r|$$

for every $r \in R_*$ and its any beginning $u$ which is a piece with respect to $R$.

The term 'metric' is used to signify this criterion because it resembles a quantitative measure, similar to a metric, illustrating the correlation between the length of the initial segment of a relation and the overall length of the relation itself. Next, we present the following condition.

**Definition 4.9** ([39, Definition 1.8])**.** Let $p \in \mathbb{N} \setminus \{0\}$. A set of relations $R$ is said to satisfy the condition $C(p)$ if every element of $R_*$ is a product of at least $p$ pieces.

This condition is alternatively known as the 'non-metric' condition. The relationship between these two conditions is closely examined in the following proposition.

**Proposition 4.10** ([39, Page 14])**.** The condition $C'\left(\frac{1}{n}\right)$ implies condition $C(n+1)$ for $n \in \mathbb{N} \setminus \{0\}$.

*Proof.* Let $R$ be a set of relations satisfying the condition $C'\left(\frac{1}{n}\right)$ for some $n \in \mathbb{N} \setminus \{0\}$. Let $r \in R_*$. We need to show that $r$ can be expressed as a product of at least $n+1$ pieces. Without loss of generality, assume that $|r| \geq 1$ (if $|r| = 0$, then the proposition holds trivially). Consider $n+1$ consecutive partitions of $r$ into pieces of equal length. Let these partitions be denoted as $p_1, p_2, \ldots, p_{n+1}$. Since $p_1$ is the first piece of $r$, it is a piece by definition. Therefore, by the condition $C'\left(\frac{1}{n}\right)$, the length $|p_1| < \frac{1}{n}|r|$. Similarly, for each subsequent piece $p_i$ for $2 \leq i \leq n+1$, its length $|p_i| < \frac{1}{n}|r|$. We can express this relationship as:

$$|r| = |p_1| + \cdots + |p_{n+1}| < \frac{k}{n}|r|.$$

To ensure $\frac{k}{n} > 1$, we require $k > n$, which implies $k \geq n+1$. Consequently, every element of $R_*$ is composed of at least $n+1$ pieces. □

The final small cancellation condition is as follows.

**Definition 4.11** ([39, Definition 1.9])**.** Let $q \in \mathbb{N}$ such that $3 \leq h \leq q$. Suppose $r_1, \ldots, r_h \in R_*$ with no successive elements as an inverse pair. Then $R$ is said to satisfy **condition** $T(q)$ if at least one of the products

$$r_1 r_1, \ldots, r_{h-1} r_h, r_h r_1$$

is reduced.

Let us recall that a presentation for the fundamental groups of closed orientable surfaces of genus $g$ takes the form:

$$\langle a_1, b_1, \ldots, a_g, b_g \mid [a_1, b_1] \cdots [a_g, b_g] \rangle$$

where $[a_i, b_i] = a_i b_i a_i^{-1} b_i^{-1}$. It becomes apparent that the non-trivial pieces are the individual letters and the longest relator has a length of $4g$. Since $[a_1, b_1] \cdots [a_g, b_g]$ is the sole relator in $R$, it follows that $R$ satisfies the $C(4g)$ condition. Moreover, considering that single letters have a length of 1, $R$ also adheres to the condition $C\left(\frac{1}{4g-1}\right)$, as for any $g \geq 1$, we have $1 < \frac{4g}{4g-1}$.

## 4.3 Van Kampen diagrams

Dehn's geometric techniques relied on the use of regular tessellations of the hyperbolic plane. While we also adopt a geometric approach, it diverges from Dehn's methods.

In this section, we consider a group $G = \langle X \mid R \rangle$. We observe that an element $w \in F(X)$ is the identity in $G \cong F(X)/\langle\langle R \rangle\rangle$ if and only if $w \in \langle\langle R \rangle\rangle$. Recalling our discourse in Section 2.3, we note that $w \in \langle\langle R \rangle\rangle$ precisely when, within the free group $F(X)$, $w$ can be written as a product of conjugate elements of $R^{\pm 1}$, denoted as $w = c_1 \cdots c_n$, where $c_i = u_i r_i u_i^{-1}$, and $r_i, r_i^{-1} \in R$. By constructing such a sequence of $c_i$, we generate a diagram in the Euclidean plane that encapsulates all crucial information regarding $c_1 \cdots c_n$. These diagrams are referred to as *Van Kampen diagrams* [36, Chapter V, Section 1].[1]

Van Kampen diagrams offer a distinct approach compared to Cayley graphs by emphasising the graphical representation of group relations in the group's presentation. Whereas Cayley graphs serve to visualise the structure of a group in relation to selected generators.

To construct these diagrams we will use the notion of CW complexes. A CW complex is a fundamental structure in algebraic topology, used to study spaces by decomposing them into simple building blocks called cells. Introduced by J.H.C. Whitehead in the 1940s [61], CW complexes provide a flexible framework for understanding the topology of various spaces, including manifolds and more general topological spaces.

Before defining CW complexes, let us first introduce the concept of a cell.

**Definition 4.12.** An **$n$-cell**, denoted $e^n$, is a topological space that is homeomorphic to an $n$-dimensional open disk, represented by $D^n \setminus \partial D^n$, where $D^n$ refers to the unit disk or ball in $\mathbb{R}^n$ centred at the origin.

**Example 4.13**

When constructing Van Kampen diagrams, our focus is primarily on 0-cells, 1-cells, and 2-cells. Let us examine each of these components.

1. A 0-cell is a point.
2. A 1-cell is a line segment, which is homeomorphic to the open unit interval.
3. A 2-cell corresponds to a filled-in area enclosed by 1-cells, which is homeomorphic to the open unit disk.

Now, we can formally define a CW complex.

**Definition 4.14** ([20, Page 519])**.** A **CW complex** (or **cell complex**) is a topological space constructed by attaching cells of increasing dimension. Formally, it is defined as follows.

1. Start with a collection of 0-cells, denoted by $Y^0$.
2. Form the $n$-skeleton $Y^n$ from $Y^{n-1}$ by attaching $n$-cells $e^n_\alpha$ via maps

$$\phi_\alpha : S^{n-1} \to Y^{n-1}.$$

Geometrically, this means gluing the boundary of an $n$-dimensional disc to the previous skeleton $Y^{n-1}$ via the maps $\phi_\alpha$. In fact, $Y^n$ can be expressed as the quotient space $Y^{n-1} \sqcup_\alpha D^n_\alpha / \sim$, where the equivalence relation is defined as $x \sim \phi_\alpha(x)$ for

[1] The diagrams we are referring to were initially discovered by E. R. Van Kampen [28], although he did not utilise them in the same manner as we do. Subsequently, R. C. Lyndon and P. E. Schupp were among the first to employ these diagrams in the way that we are discussing.

$x \in \partial D^n_\alpha$. Consequently, the cell $e^n_\alpha$ is homeomorphic to the image of $D^n_\alpha \setminus \partial D^n_\alpha$ under the quotient map.

3. Define the CW complex $Y = \bigcup_n Y^n$ with the weak topology[2].

If $Y = Y^n$ for some $n \in \mathbb{N}$, then $Y$ is said to be finite-dimensional, and the smallest such $n$ is the dimension of $Y$, the maximum dimension of the cells of $Y$.

**Remark 4.15.** Condition (3) becomes redundant when $X$ is finite-dimensional. This is because, if $X$ is finite-dimensional, and $\mathcal{U}$ is an open subset of $X = X^n$ for some $n \in \mathbb{N}$, the definition of the quotient topology on $X^n$ implies that $\mathcal{U} \cap X^{n-1}$ is open in $X^{n-1}$. By the same reasoning, $\mathcal{U} \cap X^{n-2}$ is open in $X^{n-2}$, and this logic extends similarly for all the skeleta $Y^{n-i}$ for $0 \leq i \leq n$.

For each cell $e^n_\alpha$, its **characteristic map**, denoted as $\Phi_\alpha$, is defined as the composition $D^n_\alpha \hookrightarrow Y^{n-1} \sqcup_\alpha D^n_\alpha \to Y^n \hookrightarrow Y$. Here, $D^n_\alpha$ can be thought as the 'pre-image' of the cell $e^n_\alpha$ before being mapped into the CW complex. We show an illustration of this map in Figure 19.

This composition is continuous because it is formed by composing continuous maps. The inclusion map $X^n \hookrightarrow X$ is continuous due to condition (3) in Definition 4.14, which guarantees the continuity of the 'gluing' maps in a CW complex.

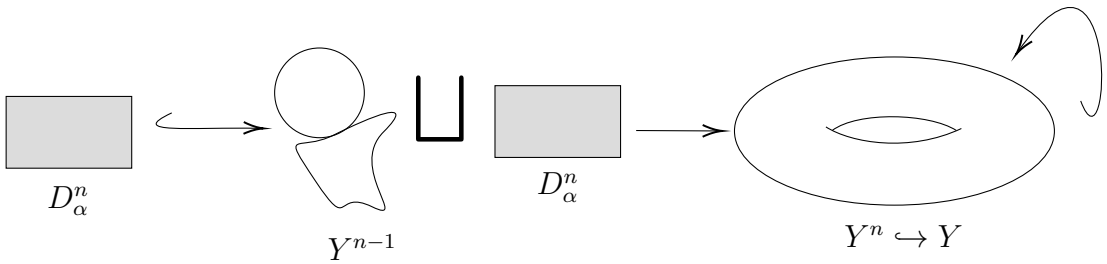



Figure 19: A cartoon depicting the function of the characteristic map.

**Example 4.16** ([20, Example 0.1])**.** A 1-dimensional cell complex $Y = Y^1$ is defined as a graph, where the vertices represent the 0-cells and the edges represent the 1-cells.

We define a Van Kampen diagram, which is an example of a 2-dimensional cell complex.

**Definition 4.17** ([36, Chapter V, Section 1][5, Part III, Chapter 2])**.** Consider a presentation $\langle X \mid R \rangle$ for the group $G$. Let $M \subset \mathbb{R}^2$ be a planar finite cell complex satisfying the following conditions:

1. The complex $M$ is both connected and simply connected.
2. A 0-cell (vertex) $O$ (designated as the base vertex) lies on the topological boundary of $M \subseteq \mathbb{R}^2$.
3. Each 1-cell (edge) of $M$ is oriented and labelled by a letter $x \in X$.
4. The boundary of each 2-cell $\mathcal{D}$, is labelled by a word in $\langle\langle R \rangle\rangle$. This word is obtained by reading the labels on the edges as they are traversed, starting from a vertex on

[2]In the **weak topology** a set $\mathcal{U} \subset Y$ is open (or closed) if and only if $\mathcal{U} \cap Y^n$ is open (or closed) in $Y^n$ for each $n$.

the boundary of $\mathcal{D}$, in either of the two possible directions. Each label on the 1-cell is assigned a $\pm 1$ exponent based on whether the traversal direction aligns with or opposes the orientation of the 1-cell. The choice of direction and starting point may lead to inversion and/or cyclic conjugation of the word.

5. The **boundary word** of $M$ is the word $w$ read on the boundary of $\mathbb{R}^2 \setminus M$ starting from the base vertex $O$.

We define $M$ as a **Van Kampen diagram** for the boundary word $w$ over the presentation $\langle X \mid R \rangle$.

**Remark 4.18.** We will abbreviate Van Kampen diagrams as simply 'diagrams'.

As a result, the 1-skeleton of $M$ constitutes a finite connected planar graph $\Gamma$ that is embedded in $\mathbb{R}^2$. The bounded regions of this graph precisely correspond to the 2-cells of $M$. The embedding of a graph $\Gamma$ in the plane imposes significant constraints on the structure of $M$, such as those related to the Euler characteristic. These constraints allow us to infer from local properties of $\Gamma$, reflecting combinatorial conditions in the set $R$, to properties of its boundary. These relationships are extensively explored in [36, Chapter V].

**Example 4.19** ([36, Chapter V, Theorem 1.1.])**.** In this example we provide a construction for the Van Kampen diagram for a sequence of elements $c_1, \ldots, c_n$ in the free group $F$. We can write each element $c_i$ of a free group as the unique product $u_i r_i u_i^{-1}$ where $u_i r_i u_i^{-1}$ is reduced without cancellation and $r$ is cyclically reduced.

To create the Van Kampen diagram of $c = uru^{-1}$, we begin with a 0-cell $v_1$ and a 1-cell originating from $v_1$ labelled $r$. If $u = 1$, we set the base point $O = v_1$, and $M$ is constructed accordingly. However, if $u \neq 1$, we select a 0-cell other than $O$ and connect it to $O$ using a 1-cell labelled $u$, along with a loop labelled $r$ at $v_1$.

In general, to construct the diagram of the product $c_1 \cdots c_n$, we assemble a diagram $M'$ by combining individual diagrams $M_1, \ldots, M_n$ corresponding to each factor $c_1, \ldots, c_n$. These individual diagrams are arranged sequentially around a shared base point $O$, as depicted in Figure 20, where we have chosen the orientation of the diagram to be clockwise.

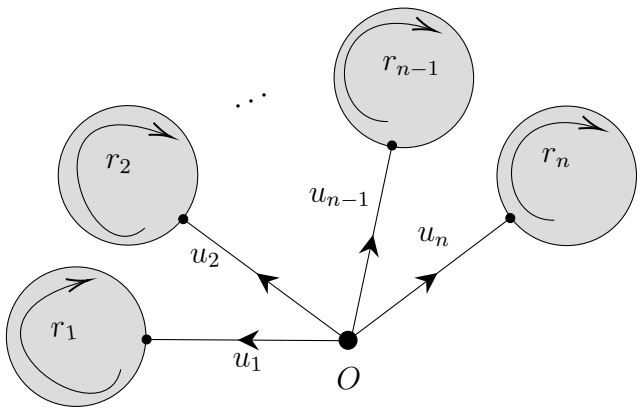



Figure 20: Balloon Van Kampen diagrams.

**Example 4.20** ([36, Chapter V, Theorem 1.1.])**.** In this example, we demonstrate the concept of cancellation within a Van Kampen diagram. We examine a scenario within the diagram resembling the one depicted in Figure 21.

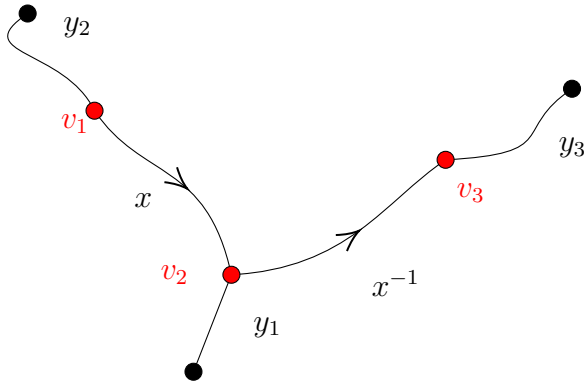



Figure 21: A scenario illustrating the occurrence of cancellation.

In this particular instance, we have presumed that $v_1$ is distinct from both $v_2$ and $v_3$. Given this assumption, we have the ability to 'fold' the 1-cell labelled $x$ onto the 1-cell labelled $x^{-1}$ without changing the diagram's structure, as illustrated in Figure 22.

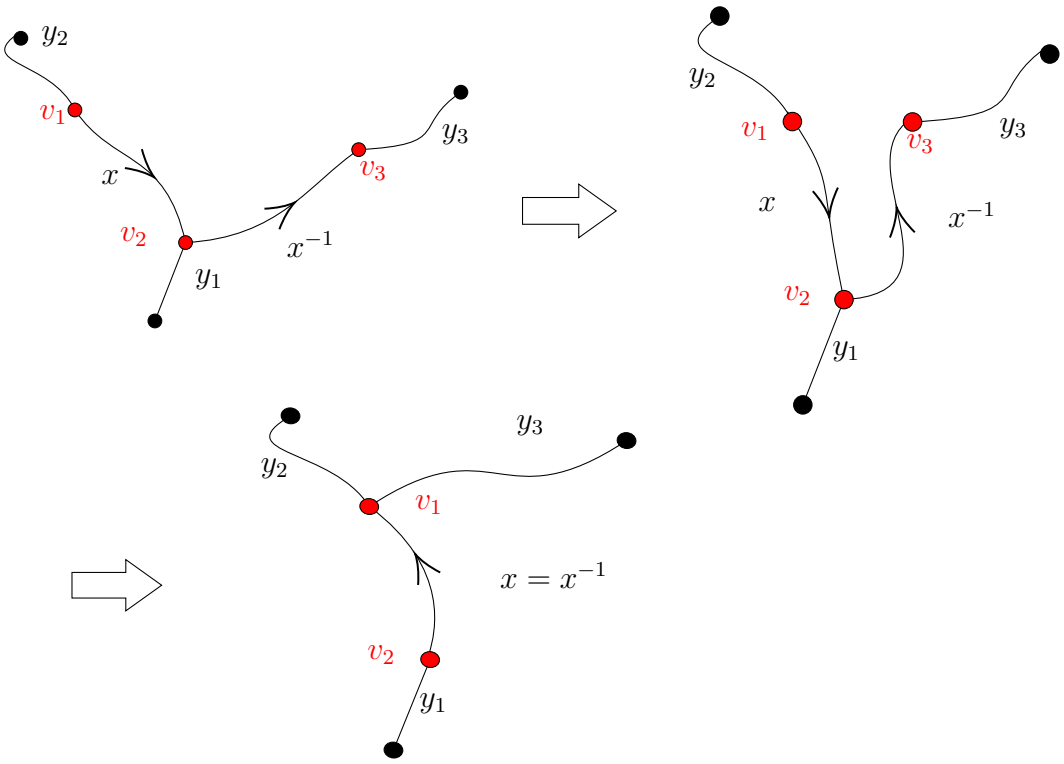



Figure 22: 'Folding' $x$ onto $x^{-1}$.

Following the 'fold', we assign a label to the new 1-cell based on the chosen orientation. For instance, adopting a clockwise direction permits us to designate the label as $x$; conversely, traversing this 1-cell in the opposite direction enables us to interpret the label as $x^{-1}$. Despite resulting in fewer 0-cells and 1-cells, the resulting diagram preserves all pertinent information regarding the corresponding word.

Nevertheless, there exists the possibility of encountering the scenario portrayed in Figure 23, where $v_1 = v_3$.

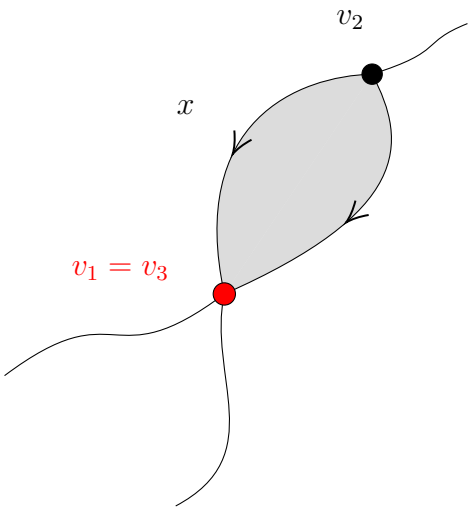



Figure 23: The case when $v_1 = v_3$.

Suppose the loop is denoted by $\delta$. Should such a circumstance occur, we generate a diagram by deleting $\delta - v_1$ and the corresponding 2-cell linked to this loop, as illustrated in Figure 24.

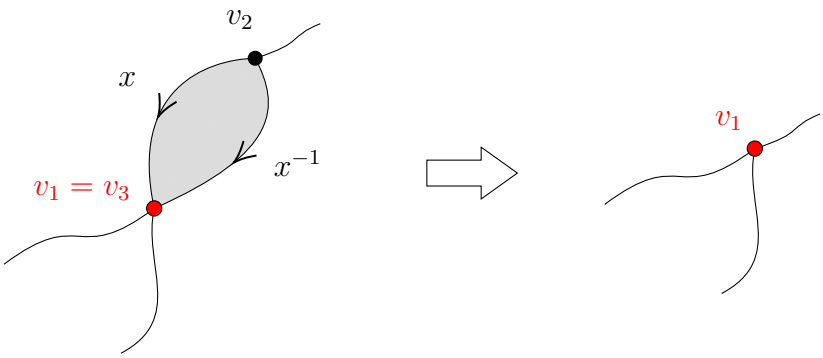



Figure 24: Deleting $\delta - v_1$ and the 2-cell of $\delta$.

The small cancellation conditions $C'(\lambda)$ and $C(p)$ are intrinsically linked to a geometric interpretation involving 1-cells, as noted in [39, Remark 1.2]. Specifically, a 1-cell contained entirely within a 2-cell $\mathcal{D}$ (meaning it is fully enclosed within $\mathcal{D}$ and does not intersect the diagram's boundary) satisfies the $C'(\lambda)$ condition if its length is smaller than $\lambda\,|\partial\mathcal{D}|$. Similarly, the $C(p)$ condition implies that the boundary of every 2-cell in the diagram consists of at least $p$ 1-cells.

## 4.4 Word problem

As outlined earlier in Section 4, Dehn's algorithm is applicable solely to the fundamental groups of closed orientable surfaces with genus $g \geq 2$. This restriction arises from the fact that the fundamental groups of surfaces with $g = 0$ (the sphere) or $g = 1$ (the torus) are too elementary for his algorithm to operate. As demonstrated in Section 4.1, $\pi_1(S^2) \cong \{1\}$, rendering the word problem trivial. Conversely, for $\pi_1(\mathbb{T}^2) \cong \mathbb{Z} \times \mathbb{Z}$, to determine the equality of two elements $(a_1, b_1)$ and $(a_2, b_2)$, a straightforward comparison of their components suffices; that is, we compare $a_1$ with $a_2$ and $b_1$ with $b_2$. This process is similar to solving the word problem in $\mathbb{Z} \cong F_1$, for which Theorem 2.37 assures us of decidability and provides an algorithm to do so.

Dehn observed that within the fundamental group of closed orientable surfaces of genus $g \geq 2$, a non-trivial word $w = 1$ in $G$ implied the presence of more than half of an element from $R$ [17, Page 642]. Building upon Dehn's insights, Martin Greendlinger expanded upon this concept, and proved that for group presentation satisfying the $C'(\lambda)$ condition for $\lambda \leq \frac{1}{6}$ this result also holds. This result is known as **Greendlinger's lemma** [39, Theorem 1.4].

We now present Dehn's algorithm for the word problem.

**Theorem 4.21** ([39, Page 19])
The word problem is decidable in fundamental groups of closed orientable surfaces of genus $g \geq 2$.

*Dehn's algorithm.* Without loss of generality, we may focus on cyclically reduced words, as any proper subword of a cyclic permutation of the word allows us to ascertain, through induction, whether it equals 1 in the group $G$ or not.

Consider a cyclically reduced word $w$ of length $n$ such that $w = 1$ in $G$. According to Greendlinger's lemma, if $w = 1$ in $G$, then the cyclic word $w$ contains a subword $v$ satisfying a relation $r = ab \in R$ and $|v| > \frac{1}{2}|r|$. We address two cases.

1. If $w$ lacks such subwords, it fails to satisfy Greendlinger's lemma, indicating $w \neq 1$.
2. If such a subword exists, we substitute it with the subword $u^{-1}$ within $w$, yielding a new word $w'$ where $w = w'$ in the group $G$ and $|w'| < |w|$ since $|u| < |v|$. Given the finite length of the initial $w$, we can iterate this process. Through induction, we can ascertain whether $w' = 1$ in the group $G$.

Hence, the word problem is decidable. □

The potency of Van Kampen diagrams in tackling the word problem lies in the fact that the boundary word of the diagram equals the identity element in the group. This important result is commonly referred to as the **Van Kampen Lemma** [28, Lemma 1].

**Example 4.22.** We have employed this result discreetly in our discussion concerning Figure 15 and Figure 17(f) to demonstrate the relations $[a, b] = 1$ and $[a, b][c, d] = 1$ respectively.

Now, we will explore an alternative example involving a more intricate Van Kampen diagram. Let $G$ be a group with relations $a^4 = 1$ and $ba^{-1}b^{-1} = c$. It is worth noting that $c^4 = b(a^{-1})^4b^{-1} = 1$. We provide a visual representation of these relations in the Van Kampen diagram of $G = \langle a, b \mid a^4, (ba^{-1}b^{-1})^4 \rangle$ as depicted in Figure 25.

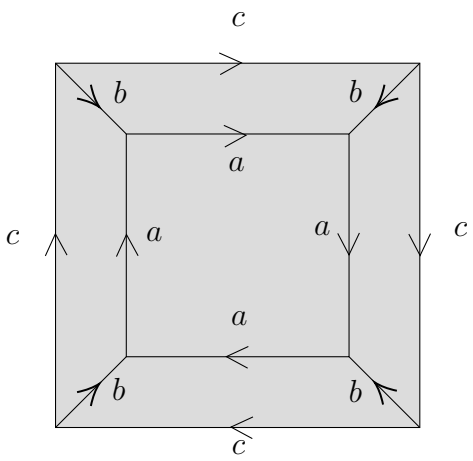



Figure 25: Diagrammatic view of the relation $c^4 = 1$.

Indeed, commencing from any 1-cell within the inner triangle and traversing it clockwise, we consistently return to our initial point. Likewise, the quadrilaterals affixed to the square establish the second provided relation $c^{-1}ba^{-1}b^{-1} = 1$. Moreover, by considering the outer boundary of the square, we deduce the relation $c^4 = 1$, employing a similar reasoning as previously outlined in the discussions regarding Figure 15 and Figure 17(f) concerning the count of 1-cells within this complex.

Building upon this insight, Lyndon's research, as documented in [35] and [36, Chapter V, Theorem 6.3], significantly contributed to advancing the small cancellation conditions under which the word problem becomes decidable. His findings demonstrated that the word problem is decidable for groups satisfying conditions $C(6), C(4)$ and $T(4)$, or $C(3)$ and $T(6)$.

He introduces a combinatorial notion of an 'area' of a Van Kampen diagram $M$ as its count of 1-cells (or the total number of 2-cells in the dual diagram[3]). Lyndon observes that the diagrams in his consideration 'grow towards the boundary', indicating that an increase in the diagram's size is mirrored in the boundary since there are more relations and generators which partition the boundary word of the diagram [36, Chapter V, Section 6].

Lyndon proves that this area is bounded by a function dependent on the defining characteristics of the boundary 0-cells (i.e. 0-cells which lie on the boundary of the diagram), a result commonly known as the "Area theorem" ([36, Chapter V, Theorem 6.2]).

Given an element in the group $G \cong F/\langle\langle R \rangle\rangle$, we can associate it with a corresponding word in the quotient group, denoted as $w$. If $w \in \langle\langle R \rangle\rangle$, then we can construct a Van Kampen diagram with $w$ as its boundary word. Applying the Area theorem, we conclude that $w$ can be represented as a finite product of elements in the form $c_i = u_i r_i u_i^{-1}$, where $|u_i| \leq C$ and $C$ is a constant determined by the theorem. Given that there exists only a finite number of such products to investigate, we can systematically examine whether any of them are equal to $w$. Thus, the word problem is decidable.

[3]Given a diagram $M$, the **dual diagram** $M^*$ [36, Page 245] is formed by selecting a vertex $v_i^*$ in each region $\mathcal{D}_i$ of $M$ and drawing edges between these vertices according to specific rules, resulting in a graph $\Gamma_{M^*}$ whose regions correspond to those of $M$ and contain interior vertices of $M$.

## 4.5 Conjugacy problem

Analogous reasoning to that discussed in the previous section is used to explain why Dehn's algorithm does not hold for the fundamental groups of a circle, sphere, and torus.

**Theorem 4.23** ([17, Page 642])
The conjugacy problem is decidable in fundamental groups of closed orientable surfaces of genus $g \geq 2$.

*Dehn's algorithm.* We begin by considering the following trivial cases for two words $w$ and $v$:

1. if both $w$ and $v$ are the identity, then they are certainly conjugate in the group; or
2. if one of $w$ or $v$ is the identity and the other is not, they are not conjugate in the group.

Suppose neither of these trivial cases occurs. Then, we transform the words $w$ and $v$ into cyclically reduced forms, denoted as $w'$ and $v'$ respectively. We take the relators for the group and represent their letters on a circle, similarly for the inverse of the relators. If more than half of the symbols of the cyclically written relator, or its inverse, appear in the same order in some cyclic permutation of $w'$, we select that permutation and replace the corresponding sequence of symbols with the inverse of the product of the remaining symbols of the relator. Continuing this process, we eventually obtain a cyclically reduced word $w''$ that does not contain half of the symbols of the cyclically written defining relator, or its inverse, in the same order.

We then generate all cyclic permutations of $w''$, denoted as $w_1, w_2, \ldots, w_m$. Similarly, we repeat this process for $v$, obtaining words $v_1, v_2, \ldots, v_n$.

We consider a finite number of pairs of words:

$$\{(w_i, v_j)\}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n, \tag{1}$$

$$\{(u_k^{-1} w_i u_k, v_j)\}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, N, \tag{2}$$

and

$$\{(u_k v_i u_k^{-1}, v_j)\}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n \quad k = 1, 2, \ldots, N. \tag{3}$$

We apply Dehn's algorithm for the word problem to determine whether two words in each of the pairs (1), (2), and (3) are equal in the group. If at least one pair consists of equal words, then $w$ and $v$ are conjugate. If none of the pairs consists of equal words, then $w$ and $v$ are not conjugate. □

Schupp's research outlined in [51] and [36, Chapter V, Theorem 7.6], expanded the scope of the conjugacy problem's decidability. His contributions extended the problem's decidability to groups satisfying small cancellation conditions: $C(6), C(4)$, and $T(4)$, or $C(3)$ and $T(6)$.

For non-trivial instances of the conjugacy problem, we turn to annular diagrams. Here, we inquire whether $w = uvu^{-1}$ in the group, which is equivalent to verifying $wuv^{-1}u^{-1} = 1$ in the group. Conceptually, we can visualise the diagram for this word as a rectangle with

sides labelled $w$, $u$, $v^{-1}$, and $u^{-1}$. Subsequently, we can transform this rectangle into an annulus by matching the sides labelled by $u$ and $u^{-1}$, as demonstrated in Figure 26.

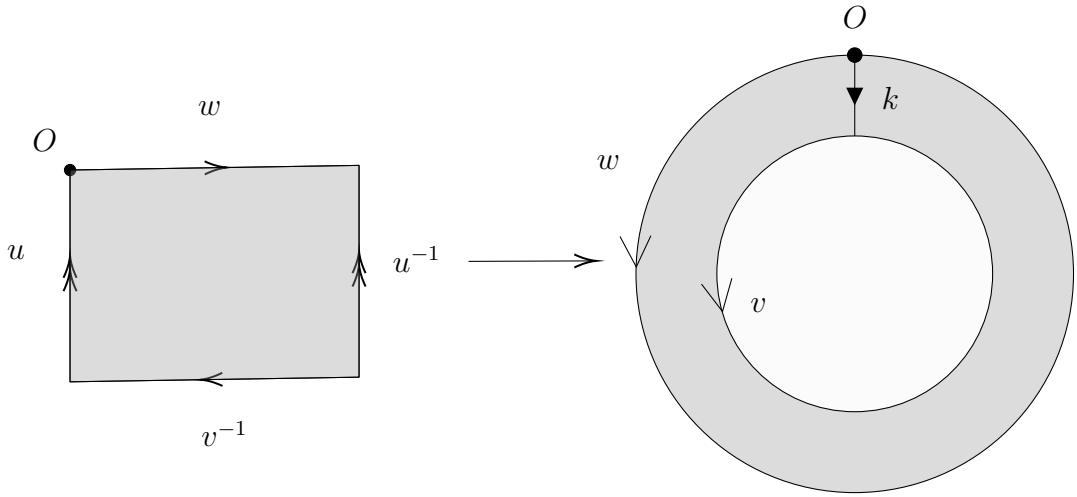



Figure 26: Transformation to an annular diagram.

In this case, we are unable to apply the Area theorem, as noted in [36, Chapter V, Section 7]. To illustrate, we begin by constructing a rectangle divided into $mn$ squares, which we then bend to form an annulus. By identifying along the side with $n$ subdivisions, we create a tessellation of the annulus. Here, the outer boundary length measures $2m$, while the number of regions equals $mn$ for any arbitrary $n$. Through careful selection of $n$, it is always possible to ensure that the number of regions surpasses the length of the outer boundary. Consequently, no function exists that can effectively bound the regions of the dual of an annular diagram, which itself constitutes an annular diagram.

However, Schupp observed that the boundaries of an annulus are connected by paths[4] formed by the boundary of the 2-cells within the annulus layer, as depicted in Figure 27. The length of these paths linking the outer and inner boundaries is at most $\frac{L}{2}$, where $L$ represents the length of the longest relator. Geometrically, the boundary labels correspond to group relators. Since the longest path made by 1-cells is at most $L$, we can traverse a path between the outer and inner boundaries and back again using a path of length at most $L$. Thus, to traverse the path once, we require precisely a path of length at most $\frac{L}{2}$.

[4]By 'path' we refer to the intuitive concept of traversal. However, we can alternatively interpret it using the mathematical definition, wherein there exist 0-cells on the boundaries of the annulus and within the gap of the annulus, which can act as the initial and terminal vertices connected by a 1-cell. Consequently, to construct a path between the boundaries of the annulus, we may opt for a singular path that starts and ends on the boundaries or multiple paths linked by common 0-cells within the gap of the annulus.

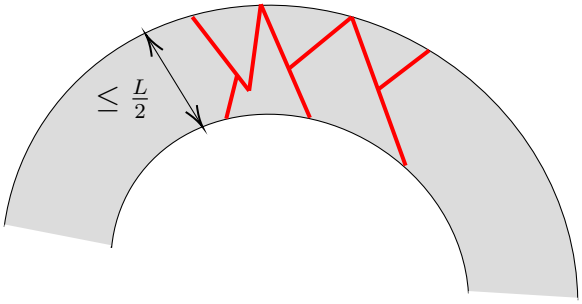



Figure 27: Paths linking the inner and outer boundary of the annulus.

With this concept in mind, we can consider multiple concentric annuli, each with an outermost boundary linked by paths to their innermost boundary. Given that the largest concentric annulus has its outermost boundary labelled as $w$ and the smallest annulus has its innermost boundary labelled as $v$, by connecting these boundaries via paths, we can find the conjugating element (up to cyclic conjugation) between $w$ and $v$. Schupp demonstrates this process terminates as he considers only a finite sequence of these diagrams, bounded by a finite quantity of regions, through his "Layer Theorem" [36, Chapter V, Theorem 7.4]. Furthermore, as discussed earlier, the width between each layer of the annuli is limited, allowing us to construct a finite number of paths from the boundary labelled by $w$ to the boundary labelled by $v$. Refer to Figure 28 for a visual representation of this procedure.

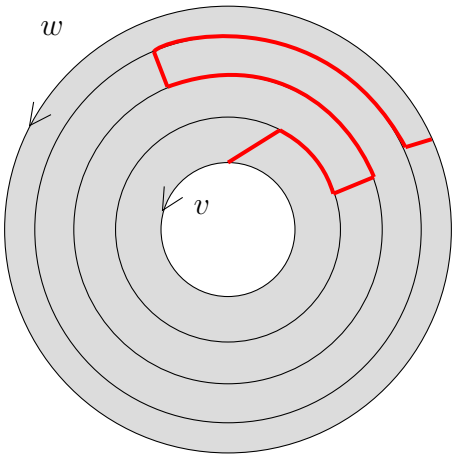



Figure 28: Visual representation of paths connecting the edges $w$ and $v$.

The necessity for multiple bounds to address the conjugacy problem, as opposed to the word problem which only requires a single bound, underscores the inherent complexity of the conjugacy problem.

# 5 The decidability of decision problems

In preceding sections of this thesis, we explored into specific instances of groups where both the word problem and the conjugacy problem are decidable. However, our current focus shifts towards addressing Questions 1.3 and 1.5. Regrettably, the answer to both is no; there exists no algorithm capable of decide them due to insufficient restrictions on the problem. To prove this, it is enough to identify a group for which either problem is undecidable. For further insights into the conjugacy problem, consult [15].

In this section, we will develop into the theory required to address the undecidability word problem. We will follow [50, Pages 418–460] to support our exploration.

## 5.1 Semigroups

To demonstrate the undecidability of the word problem, we construct a *semigroup* with specific characteristics that allow for the encoding of group words while preserving their operational structure. Using these characteristics, we simplify computations associated with the word problem in groups, effectively translating them into equivalent operations within the constructed semigroup.

In this section, we will establish the foundational theory required for the subsequent sections. We will follow the work of [18, Chapter 1].

The theory of semigroups bears significant resemblance to that of group theory, particularly in concepts such as the definition of a homomorphism, quotient semigroup, and presentation. While we will not delve extensively into formalism, as many ideas are already well-known, we will offer sufficient explanation to establish comprehension for the objective of this thesis.

**Definition 5.1.** A **semigroup** is a set $\gamma$ together with an associative binary operation on $\gamma$.

**Example 5.2.** Here are some examples of semigroups:

1. groups, and
2. every (associative) ring or algebra can be considered as a semigroup under multiplication.

The aim of this section is to outline the definition of the presentation of a semigroup, which will be used in subsequent discussions. Although the definition is intuitively similar to that of a group's presentation, we will formally construct it for clarity.

Recall Definition 2.24, where the presentation of a group was defined as an isomorphism involving the quotient of a free group. We will employ analogous concepts here. Before proceeding, it is necessary to define the quotient of a semigroup.

In defining the quotient group, we employ equivalence relations. In the context of a semigroup, we utilise a comparable concept, which we will now introduce.

**Definition 5.3** ([30, 3.2 Congruences])**.** A **congruence** on a semigroup $\gamma$ is an equivalence relation on the $\gamma$ stable under left and right multiplication. That is for every $x, y, z \in \gamma$ if $x \sim y$ then $xz \sim yz$ and $zx \sim zy$. We write $x \sim_C y$ to denote congruent elements.

Given a congruence $\sim_C$ on a semigroup $\gamma$, we denote by $[x]$ the congruence class of

an element; that is, if $x \in \gamma$, then $[x] = \{y \in \gamma : x \sim_C y\}$. With this notation, we can establish a semigroup structure on the set of all congruences, denoted by $\gamma/\sim_C$.

**Definition 5.4** ([30, Page 10])**.** The **quotient semigroup** of $\gamma$ by $\sim_C$ is the set $\gamma/\sim_C$ where the binary operation is given by $[x] \cdot [y] = [xy]$ for all $x, y \in \gamma$.

With this notion established, let us now define the concept of the free semigroup. Analogous to the free group, the free semigroup is a construct that represents the most general semigroup generated by a given set. Elements are formed purely through the concatenation of elements from the specified set of generators, without any imposed relations other than associativity.

**Definition 5.5.** We define the semigroup $\gamma_{F_X}$ as the set of all non-empty sequences $(x_1, \ldots, x_n)$ of elements $x_1, \ldots, x_n$ of $X$. The integer $n \geq 1$ is the **length** of $(x_1, \ldots, x_n)$. The operation on $\gamma_{F_X}$ is concatenation:

$$(x_1, \ldots, x_n)(y_1, \ldots, y_m) = (x_1, \ldots, x_n, y_1, \ldots, y_m).$$

**Proposition 5.6.** Every element of $\gamma_{F_X}$ can be written uniquely as a product of elements of $X$.

*Proof.* In $\gamma_{F_X}$ we can write $(x_1), \ldots, (x_n) = (x_1, \ldots, x_n)$ for each $x_i \in X$. □

By Proposition 5.6 we can write the elements of $\gamma_{F_X}$ as words $(x_1, \ldots, x_n) = x_1 \ldots x_n$ in the alphabet $X$. Henceforth, we refer to words with positive exponents as **positive words**.

Similar to free groups, free semigroups also demonstrate a universal property.

**Theorem 5.7** (Universal property of semigroups [18, Chapter I, Proposition 5.4])
Every mapping $\phi$ of $X$ into a semigroup $\gamma$ extends uniquely to a homomorphism of semigroups $\psi : \gamma_{F_X} \to \gamma$. Furthermore, the image of $\psi$ is the subsemigroup of $\gamma$ generated by $\phi(X)$.

*Proof.* We must prove the following.

*Existence.* Let $\phi : X \to \gamma$ be a mapping from $X$ to the semigroup $\gamma$. We define $\psi : \gamma_{F_X} \to \gamma$ as follows: for any word $w = x_1 x_2 \ldots x_n \in \gamma_{F_X}$, where $x_i \in X$, let

$$\psi(w) = \phi(x_1)\phi(x_2) \cdots \phi(x_n).$$

*Uniqueness.* Assume there exists another homomorphism $\psi'$ extending $\phi$. Then, for any word $w = x_1 x_2 \ldots x_n \in \gamma_{F_X}$, we have:

$$\begin{aligned} \psi'(w) &= \psi'(x_1 x_2 \cdots x_n) \\ &= \psi'(x_1)\psi'(x_2) \cdots \psi'(x_n) \\ &= \phi(x_1)\phi(x_2) \ldots \phi(x_n) \\ &= \psi(w). \end{aligned}$$

Thus, $\psi' = \psi$, and $\psi$ is unique.

We now prove the image of $\psi$ is the subsemigroup of $\gamma$ generated by $\phi(X)$. Let $\langle\phi(X)\rangle$ be the subsemigroup of $\gamma$ generated by $\phi(X)$. Since $\psi$ maps each word in $\gamma_{F_X}$ to a product of elements in $\phi(X)$, it follows that the image of $\psi$ contains all elements in $\langle\phi(X)\rangle$. Conversely, every element in $\langle\phi(X)\rangle$ can be expressed as a product of elements in $\phi(X)$, which corresponds to a word in $\gamma_{F_X}$. Therefore, the image of $\psi$ is $\langle\phi(X)\rangle$. □

By the universal property of semigroups, we derive a version similar to Corollary 2.21.

**Corollary 5.8** ([18, Chapter I, Corollary 5.5])
Every semigroup is a homomorphic image of a free semigroup.

*Proof.* Using Theorem 5.7, we observe that every semigroup generated by a set $X$ can be viewed as a homomorphic image of $\gamma_{F_X}$. This is because each semigroup can be generated by some subset $X \subseteq \gamma$, such as when $X = \gamma$. □

Free semigroups provide a way to construct semigroups by generators and relations.

**Definition 5.9.** Given a set $X$ we define a **relation** between elements of $X$ as an ordered pair of elements of $\gamma_{F_X}$.

In conventional notation, a relation $(u, v)$ where $u, v \in X$ is typically denoted as an equality $u = v$. It is crucial to differentiate this notation from equalities within $\gamma_{F_X}$, as established in Proposition 5.6. When examining a mapping $\phi : X \to \gamma$, the relation $u = v$ holds in $\gamma$ via $\phi$ if and only if $\psi(u) = \psi(v)$ holds in $\gamma$, where $\psi$ denotes the unique homomorphism $\psi : \gamma_{F_X} \to \gamma$ extending $\phi$.

Let $X$ represent a set and $R$ a set of relations among the elements of $X$, thereby making $R$ a binary relation on $\gamma_{F_X}$. Denoting the quotient semigroup $\gamma_{F_X}/\sim_C$ as $\langle X \mid R\rangle$, we observe that $\langle X \mid R\rangle$ is equipped with a mapping $\iota : X \to \langle X \mid R\rangle$, where $x \mapsto [x]$. The homomorphism $\gamma_{F_X} \to \gamma_{F_X}/\sim_C$ extends $\iota$, indicating that $\iota(X)$ generates $\langle X \mid R\rangle$. Consequently, every relation $u = v$ in $R$ holds in $\langle X \mid R\rangle$ via $\iota$. Note that $\iota$ may not be injective. Moreover, $\langle X \mid R\rangle$ is not the only semigroup generated by $\iota(X)$ where every relation from $R$ holds; however, it is the largest such semigroup. This distinction arises because $\langle X \mid R\rangle$ contains all the essential elements and relations required to meet the given conditions. Any larger semigroup would inevitably add extra elements or relations, which would contradict the constraints established by the relations in $R$.

**Definition 5.10.** A **presentation** of a semigroup $\gamma$ consists of a pair $\langle X \mid R\rangle$, where: $X$ is the set of generators and $R$ is the set of relations, such that

$$\gamma \cong \gamma_{F_X}/\sim_C .$$

By Corollary 5.8, we can guarantee that every semigroup has a presentation. We can choose $X$ as any subset of $\gamma$ that generates $\gamma$, and $R$ can represent any binary relation that generates the congruence induced by the mapping $\gamma_{F_X} \to \gamma$.

Now equipped with the necessary theoretical framework, we are ready to articulate the word problem within a semigroup.

**The semigroup word problem**. Given a semigroup $\gamma$, does there exist an algorithm capable of determining whether any pair of words $u$ and $v$ are equal in $\gamma$. If such an

algorithm exists, we classify the problem as decidable; otherwise, it is deemed undecidable. In Definition 5.28, we formalise this concept of decidability within a semigroup.

> **Remark 5.11.** In contrast to the word problem for groups, this does not equate to determining whether $uv^{-1} = 1$ in $\gamma$, where 1 symbolises the identity element. This distinction arises from the absence of requirements for semigroups to possess an identity element or inverse elements.

## 5.2 Turing machines

In this section, we introduce the concept of Turing machines, which will play a crucial role in formalising the notion of decidability for the problems at hand. Turing machines will serve as indispensable tools in demonstrating the undecidability of both the word and conjugacy problems.

A Turing machine is a mathematical model of computation, introduced by Alan Turing in 1936 [59, Computing machines]. Informally, we can visualise a Turing machine as a box with an infinitely long tape running through it. This tape is divided into discrete squares, extending infinitely to the left and right. Each square on the tape can hold a symbol from a finite alphabet, which includes characters such as $s_0, s_1, \ldots, s_M$. Additionally, the machine can be in any one of a finite number of states, denoted as $q_0, q_1, \ldots, q_N$.

At any given moment during its operation, the Turing machine finds itself in a particular state, say, $q_i$, as it 'scans' a specific square on the tape, which contains a single symbol, such as $s_j$ (with $s_0$ signifying a blank symbol). The machine's next action is determined by both its current state ($q_i$) and the symbol it is currently scanning ($s_j$), as well as its initial configuration. It can follow one of the following instructions:

1. replace the symbol $s_j$ with another symbol, say $s_k$, on the same square and continue scanning;
2. move one square to the right on the tape and scan the symbol in that square;
3. move one square to the left on the tape and scan the symbol in that square.

The Turing machine proceeds with these operations, and it can repeat them indefinitely or until it enters a special state, often denoted as $q_0$, signifying a 'halt' state, at which point the machine ceases its operation.

To initiate the Turing machine, it is provided with an initial tape configuration, which may contain non-blank symbols printed on it, with one symbol per square. It is also set to scan a specific square while being in an initial or 'starting state', such as $q_1$. The machine may either reach a halting state and stop, or it may continue its computation indefinitely.

Turing machines serve as a foundational concept in theoretical computer science, providing a rigorous framework for understanding algorithmic computation. They are capable of simulating any algorithm that can be computed by a digital computer, making them invaluable tools for exploring the limits and possibilities of computation.

We shall proceed to establish a precise mathematical definition of a Turing machine. While Turing machines are formally defined by 7-tuples, as outlined in [24, Page 319], for our purposes, we will adopt Emil Post's concise 4-tuple definition [47].

**Definition 5.12.** Choose, once and for all, two infinite lists of letters:

$$s_0, s_1, s_2, \ldots \quad \text{and} \quad q_0, q_1, q_2, \ldots$$

A **quadruple** is 4-tuple of one of the following types:

$$q_i s_j s_k q_l,$$
$$q_i s_j R q_l,$$
$$q_i s_j L q_l.$$

**Definition 5.13.** A **Turing machine** $T$ is a finite set of quadruples no two of which have the same first two letters. The **alphabet** of $T$ is the set $\{s_0, s_1, \ldots, s_M\}$ of all $s$-letters occurring in its quadruples.

These quadruples serve as the instructions guiding the actions of the Turing machine. As outlined in Definition 5.12, there are only three types, each defined as follows.

1. A **state transition quadruple** is denoted by $q_i s_j s_k q_l$. This instruction specifies that when the machine is in state $q_i$ and reads symbol $s_j$, it should erase $s_j$, print $s_k$, and transition to state $q_l$.
2. A **right movement quadruple** is expressed as $q_i s_j R q_l$. This instruction dictates that when the machine is in state $q_i$ and reads symbol $s_j$, it should move one square to the right and transition to state $q_l$.
3. A **left movement quadruple** is represented by $q_i s_j L q_l$. This instruction indicates that when the machine is in state $q_i$ and reads symbol $s_j$, it should move one square to the left and transition to state $q_l$.

These quadruples collectively provide the foundation for a Turing machine's operation, specifying how it interacts with symbols on the tape, changes states, and navigates the tape head in both rightward and leftward directions. Together, they constitute the essential elements that enable a Turing machine to carry out computational tasks.

**Definition 5.14.** An **instantaneous description** $\alpha$ is a positive word of the form $\alpha = \sigma q_i \tau$ where $\sigma$ and $\tau$ are $s$-words and $\tau$ is not empty.

> **Remark 5.15.** The concept of a positive word remains relevant in this context, where positive words are those with positive exponents.

An instantaneous description can be conceptualised as a description of the Turing machine at a specific moment. For instance, in an instantaneous description denoted as $\alpha$, the sequence of letters on the tape corresponds precisely to the symbols $s_i$ within $\alpha$, with blank spaces elsewhere. Additionally, the machine is in state $q_i$ while reading the symbol in $\alpha$ that follows state $q_i$.

**Example 5.16.** The string $s_3 s_0 q_4 s_2 s_9$ is an instantaneous description which is to be interpreted as 'the symbols on the tape are $s_3 s_0 s_2 s_9$ (with the rest of the tape blank) and the machine is in state $q_4$ scanning $s_2$'.

**Definition 5.17.** Let $T$ be a Turing machine. An ordered pair $(\alpha, \beta)$ of instantaneous descriptions is a **basic move** of $T$, denoted by $\alpha \to \beta$, if there are (possibly empty) positive $s$-words $\sigma$ and $\sigma'$ such that one of the following conditions hold:

1. $\alpha = \sigma q_i s_j \sigma'$ and $\beta = \sigma q_l s_k \sigma'$, where $q_i s_j s_k q_l \in T$;
2. $\alpha = \sigma q_i s_j \sigma'$ and $\beta = \sigma s_j q_l s_k \sigma'$, where $q_i s_j R q_l \in T$;
3. $\alpha = \sigma q_i s_j$ and $\beta = \sigma s_j q_l s_0$, where $q_i s_j R q_l \in T$;
4. $\alpha = \sigma s_k q_i s_j \sigma'$ and $\beta = \sigma q_l s_k s_j \sigma'$, where $q_i s_j L q_l \in T$;
5. $\alpha = q_i s_j \sigma'$ and $\beta = q_l s_0 s_j \sigma'$, where $q_i s_j L q_l \in T$.

**Remark 5.18.** The condition specified in the definition of a Turing machine, which prohibits any two quadruples from having identical first two symbols, ensures clarity regarding the subsequent action of the machine. In other words, if $\alpha \to \beta$ and $\alpha \to \delta$, then $\beta = \delta$, eliminating any potential ambiguity.

A basic move can be intuitively understood as a fundamental operation that represents a transition between two instantaneous descriptions ($\alpha$ and $\beta$) of the machine's state during its computation. The definition of a basic move comprises five distinct cases, each reflecting specific conditions under which such transitions occur. Conditions 2 and 3 distinguish between the scenario where the machine moves to the right and encounters another letter to read, as opposed to the scenario where the machine scans the rightmost symbol of the tape and is directed to move further rightward. In the latter scenario, the string is extended by appending $s_0$ (i.e., adding a blank square to the tape). Similarly, the distinction between conditions 4 and 5 follows a similar pattern.

These basic moves serve as the foundational operations that underlie the Turing machine's computation and are instrumental in understanding its information processing capabilities.

**Definition 5.19.** An instantaneous description $\alpha$ is **terminal** if there is no instantaneous description $\beta$ with $\alpha \to \beta$.

**Definition 5.20.** If $w$ is a positive word on the alphabet of $T$ then we say $w$ is **admissible** in $T$ if there is a finite sequence of instantaneous descriptions $\alpha_1 = q_1 w, \alpha_2, \ldots, \alpha_t$, where $\alpha_i \to \alpha_{i+1}$ for all $i \leq t-1$ and $\alpha_t$ is terminal.

**Remark 5.21.** The literature we are referencing utilises the term $T$ **computes** $w$. However, we have opted for alternative terminology to mitigate potential ambiguity, since the use of 'computes' in this context may deviate from its colloquial meaning.

In particular, we will employ the term 'computation' to denote the process of executing a series of well-defined actions based on a set of instructions.

We can consider $w$ as initially inscribed on the tape, with a Turing machine $T$ in its initial state $q_1$ while scanning the first square. The operation of machine $T$ unfolds as a sequence of instantaneous descriptions, represented as $q_1 w \to \alpha_2 \to \alpha_3 \to \cdots$. This sequence will terminate only when $w$ is admissible in $T$; otherwise, the machine will continue running indefinitely.

### 5.2.1 Decidability

In Definition 1.2, we outlined the concept of decidability. Now, we aim to express this concept within the framework of Turing machines. To facilitate this transition, we introduce the concept of *enumerability*.

In simpler terms, enumerability refers to the ability to systematically list all possible solutions to a problem. For instance, when determining if a number is even or odd, enumerability entails being able to compile a list of all even and odd numbers. This list serves as a reference for decision-making: by comparing the number in question against each entry on the list, we can reach an answer.

However, if a problem lacks enumerability — perhaps due to an infinite set of potential solutions or a lack of discernible patterns — deciding becomes significantly more challenging.

Let us now proceed to formalise this concept in terms of a Turing machine.

**Definition 5.22.** Let $S^+$ be the set of all positive words on symbols $S = \{s_0, s_1, \ldots, s_M\}$. If $T$ is a Turing machine whose alphabet contains $S$, define the set

$$e(T) = \left\{w \in S^+ : w \text{ is admissible in } T\right\},$$

and say that $T$ **enumerates** $e(T)$. A subset $E$ of $S^+$ is **recursively enumerable** if there exists some Turing machine $T$ that enumerates $E$.

Each Turing machine defines a recursively enumerable subset $E = e(T) \subset A^+$, comprising all positive words on its alphabet $A$. To ascertain whether a word $w \in A^+$ also belongs to $E$, we input $q_1 w$ into $T$ and await its response; we do this by executing basic moves $q_1 w \to \alpha_2 \to \alpha_3 \to \cdots$. If $w \in E$, then $w$ is admissible in $T$, and consequently, $T$ will eventually halt. However, it remains impossible, *a priori*, to ascertain whether $T$ will halt for a specific $w$. This observation prompts the introduction of the next definition.

**Definition 5.23.** Let $S^+$ be the set of all positive words on $\{s_0, s_1, \ldots, s_M\}$. A subset $E$ of $S^+$ is **recursive** if both $E$ and its complement $S^+ \setminus E$ are recursively enumerable subsets.

If $E$ is recursive, there is never an 'infinite wait' to determine whether a positive word $w$ lies in $E$. Consider Turing machines $T$ and $T'$, where $e(T) = E$ and $e(T') = S^+ \setminus E$. For every $w \in S^+$, it is either admissible in $T$ or in $T'$. Thus, it can be decided in a finite length of time if a given word $w$ lies in $E$: input $w$ into each machine and let $T$ and $T'$ run simultaneously.

We now move forward to establish a connection between this concept and the notion of decidability. Without loss of generality, we can presume that the instructions outlined by the algorithm can be encoded as positive words using an alphabet. Let $E$ denote the set of all words resulting in a 'yes' answer, while its complement comprises all words leading to a 'no' answer. We assert that recursive sets are exactly those subsets for which an algorithm can provide a decision. This proposition is precisely the **Church-Turing thesis** [54, Page 155] which proposes that any problem that can be effectively solved by a human using an algorithm, can also be solved by a Turing machine.

> **Remark 5.24.** The Church-Turing thesis remains unproven in general due to the challenge of translating an intuitive concept into precise terms. Church substantiated this proposition in his framework of $\lambda$-calculus, as outlined in [9], while Turing similarly validated its applicability through the framework of Turing machines [59]. Significantly, these two frameworks have been proven to be completely equivalent [58].

Now, we can think of a recursively enumerable set as being 'semi-decidable' because of the way it allows us to recognise membership in the set. A set is recursively enumerable if

there exists a Turing machine that can list out its members. However, this Turing machine may not halt on inputs that are not part of the set. Therefore, while we can effectively recognise strings that belong to the set, we cannot always definitively determine if a string does not belong to the set. This notion of semi-decidability arises from the fact that we can semi-effectively decide membership in the set, but we may not be able to do so for all strings outside the set.

On the other hand, we can think of a recursive set as decidable due to the deterministic manner in which membership can be recognised. A set is deemed recursive if there exists a Turing machine that halts for every string it is provided, irrespective of whether those strings are members of the set or not. This implies that for any given string, it is computationally feasible to determine whether it belongs to the set or not. The algorithm or Turing machine that decides membership in the set will always halt and provide a clear answer, making the set decidable.

We introduce a lemma that demonstrates the closure properties of recursive sets with respect to union and intersection operations.

**Lemma 5.25.** Let $S^+$ be the set of all positive words on $\{s_0, s_1, \ldots, s_M\}$. If $E_1$ and $E_2$ are recursive subsets of $S^+$, then both $E_1 \cup E_2$ and $E_1 \cap E_2$ are also recursive subsets.

*Proof.* Given that $E_1$ and $E_2$ are recursive subsets, there exist Turing machines $T_1$ and $T_2$ that enumerate $E_1$ and $E_2$, respectively, as well as Turing machines $T_1'$ and $T_2'$ that enumerate $S^+ \setminus E_1$ and $S^+ \setminus E_2$, respectively.

*Union.* We construct a new Turing machine $T_\cup$ that simulates both $T_1$ and $T_2$ in parallel. When given input $w$, $T_\cup$ executes $T_1$ and $T_2$ simultaneously. If $w$ is admissible in either $T_1$ or $T_2$, then it is admissible in $T_\cup$. Similarly, we construct $T_\cup'$ to simulate $T_1'$ and $T_2'$ concurrently. If $w$ is admissible in either $T_1'$ or $T_2'$, then it is admissible in $T_\cup'$. Thus, both $E_1 \cup E_2$ and $(S^+ \setminus (E_1 \cup E_2))$ are recursively enumerable, implying that $E_1 \cup E_2$ is recursive.

*Intersection.* Likewise, we construct a new Turing machine $T_\cap$ that simulates both $T_1$ and $T_2$ simultaneously. On input $w$, $T_\cap$ runs $T_1$ and $T_2$ in parallel. If $w$ is admissible in both $T_1$ and $T_2$, then it is admissible in $T_\cap$. Additionally, we create $T_\cap'$ to simulate $T_1'$ and $T_2'$ concurrently. If $w$ is admissible in both $T_1'$ and $T_2'$, then it is admissible in $T_\cap'$. Thus, both $E_1 \cap E_2$ and $(S^+ \setminus (E_1 \cap E_2))$ are recursively enumerable, and hence, $E_1 \cap E_2$ is recursive. □

We now present a technical lemma crucial for subsequent proofs and instrumental in characterising terminal instantaneous descriptions.

**Lemma 5.26.** If $T$ is a Turing machine that enumerates a set $E$, then there exists a Turing machine $T^*$ that also enumerates $E$, and an instantaneous description $\alpha$ of $T^*$ is terminal if and only if $\alpha$ involves the state $q_0$.

*Proof.* We prove the following.

*Existence of $T^*$.* Given that $T$ enumerates the set $E$, we can create a Turing machine $T^*$ to mimic $T$ and replicate its actions. Every word admissible in $T$ is likewise admissible in $T^*$, ensuring identical behaviour between $T$ and $T^*$ and thereby resulting in the enumeration of the same set $E$.

*Characterisation of a terminal description.* Let $\alpha$ represent an instantaneous description of $T^*$. If $\alpha$ includes the state $q_0$, it is inherently a terminal description, as $q_0$ is defined as the halt state. Conversely, if $\alpha'$ is a terminal description of $T^*$, we can transition to a new terminal state by employing a vacuous basic move $\alpha' \to \alpha$, where $\alpha$ involves the state $q_0$. Thus, $\alpha$ becomes the new terminal instantaneous description. □

Expanding on Lemma 5.26, consider a Turing machine $T$. When feeding various admissible words from its alphabet into $T$, each computation follows a sequence of instantaneous description, each with terminal instantaneous descriptions as illustrated in Figure 29.

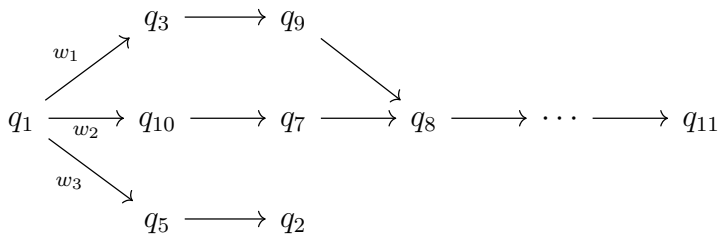



Figure 29: State transition diagram depicting a Turing machine with the initial state $q_1$ receiving input words $w_1$, $w_2$, and $w_3$ from its alphabet.

While the computation always initiates at state $q_1$, the sequence of instantaneous descriptions may vary depending on the input word. However, a crucial observation is that by employing a vacuous basic move, we can ensure that the terminal descriptions for each input of $T$ involve the state $q_0$, as depicted in Figure 30.

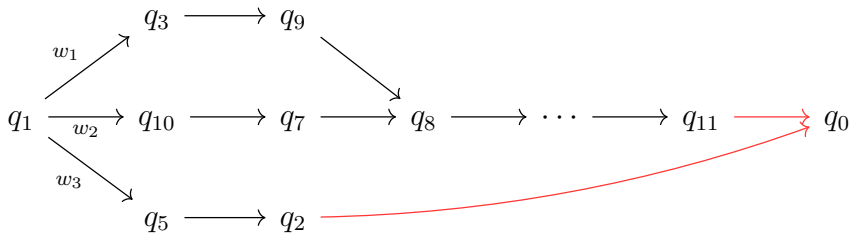



Figure 30: State transition diagram illustrating the vacuous basic move.

We conclude this section by presenting a theorem which illustrates an intriguing property of subsets of the natural numbers.

**Theorem 5.27** ([50, Theorem 12.1])
There exists a recursively enumerable subset of the natural numbers that is not recursive.

*Proof.* Firstly, we must create a recursively enumerable subset of the natural numbers. A Turing machine can be conceptualised as a finite subset of quadruples from the following set:

$$\{q_0, q_1, \ldots\} \times \{s_0, s_1, \ldots\} \times (\{s_0, s_1, \ldots\} \cup \{R, L\}) \times \{q_0, q_1, \ldots\},$$

which represents a countable set of tuples. Consequently, the number of Turing machines is countable. We assign a natural number to each of these letters in the following way:

$$R \mapsto 0; \quad L \mapsto 1;$$

$$q_0 \mapsto 2; \quad q_1 \mapsto 4; \quad \cdots \quad q_i \mapsto 2i+2;$$
$$s_0 \mapsto 3; \quad s_1 \mapsto 5; \quad \cdots \quad s_i \mapsto 2i+3$$

for $i \in \mathbb{N}$. Suppose $T$ is a Turing machine given by $m$ quadruples juxtapose them in some order to form a word $w(T)$ of length $4m$; we note that $T \neq T'$ implies $w(T) \neq w(T')$. We define the Gödel number

$$G(T) = \prod_{i=1}^{4m} p_i^{e_i},$$

where $p_i$ is the $i^{\text{th}}$ prime and $e_i$ is the natural number assigned to the $i^{\text{th}}$ letter in $w(T)$. By the uniqueness of prime factorisations, as dictated by the Fundamental Theorem of Arithmetic, we can assign a distinct Gödel number to each Turing machine. Consequently, all Turing machines can be enumerated as $T_0, T_1, \ldots, T_n$, with $T$ listed before $T'$ if $G(T) < G(T')$. For a fixed $i \in \mathbb{N}$, we let

$$q_1 s_1^{i+1} \to \alpha_{i+1,2} \to \alpha_{i+1,3} \to \cdots$$

be the computation of $T_i$ on input $q_1 s_1^{i+1}$.

We define the following set

$$E = \left\{ n \in \mathbb{N} : s_1^{n+1} \text{ is admissible in } T_n \right\}$$

therefore, $n \in E$ if and only if $n$ is admissible in the $n^{\text{th}}$ Turing machine, $T_n$. Consider Figure 31.

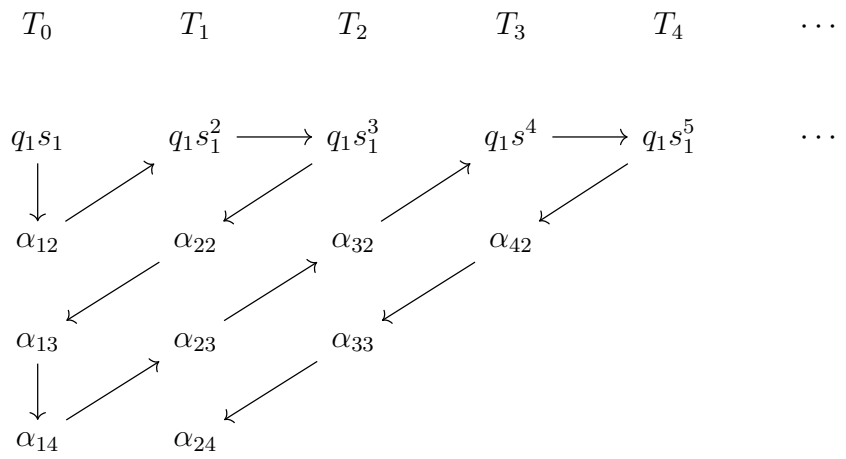



Figure 31: An array for enumerating Turing machines.

The $n^{\text{th}}$ column illustrates the sequence of basic operations carried out by the $n^{\text{th}}$ Turing machine $T_n$, starting from $q_1 s_1^{n+1}$. To enumerate $E$, we construct a Turing machine $T^*$ (as guaranteed by Lemma 5.26), which simultaneously executes $T_i$ on the input $q_1 s_1^{i+1}$ for all $i \in \mathbb{N}$, following the sequence of arrows in Figure 31. As $T^*$ encounters a terminal instantaneous description $\alpha_{ni}$ in column $n$, it verifies whether $n \in E$. Thus, $E$ constitutes a recursively enumerable subset of $\mathbb{N}$.

We now proceed to demonstrate that $E$ is not recursive. Our aim is to establish that its complement,

$$\mathbb{N} \setminus E = \{ n \in \mathbb{N} : n \notin E \} = \left\{ n \in \mathbb{N} : s_1^{n+1} \text{ is not admissible in } T_n \right\}$$

is not a recursively enumerable subset of $\mathbb{N}$. For the sake of contradiction, suppose there exists a Turing machine $T'$ that enumerates $\mathbb{N} \setminus E$. Then, there must be a Turing machine $T_m$ for some $m \in \mathbb{N}$ which enumerates $\mathbb{N} \setminus E$. We can consider two cases:

1. if $m \in \mathbb{N} \setminus E = e(T_m)$, then $s_1^{m+1}$ is admissible in $T_m$, implying $m \in E$;
2. if $m \notin \mathbb{N} \setminus E$, then $m \in E$, thus $s_1^{m+1}$ is admissible in $T_m$ by the definition of $E$. Consequently, $m \in e(T_m) = \mathbb{N} \setminus E$.

In both cases, a contradiction arises. Therefore, $\mathbb{N} \setminus E$ is not a recursively enumerable set, and thus $E$ is not recursive. □

## 5.3 The Markov-Post Theorem

In this section, we establish the connection between our developed ideas to demonstrate the undecidability of the word problem in a finitely presented semigroup. We commence by providing an informal definition of the decidability of the word problem within a semigroup. Given a semigroup $\gamma$ generated by a finite set $X$, and $X^+$ representing the set of all positive words on $X$, the semigroup $\gamma$ is said to have a decidable word problem if there exists an algorithm capable of determining, for any pair of words $w$ and $w'$ in $X^+$, whether $w = w'$ in $\gamma$. With this understanding, we can formally define undecidability within a semigroup.

**Definition 5.28.** Let $\gamma$ be a semigroup with generators $X = \{x_1, \ldots, x_n\}$, and let $X^+$ be the set of all positive words on $X$. The semigroup $\gamma$ has an **undecidable word problem** if there exists a word $w_0 \in \gamma$ such that the set $\{w \in X^+ : w = w_0 \text{ in } \gamma\}$ is not recursive.

Similarly, we define what it means for the word problem to be decidable in a group.

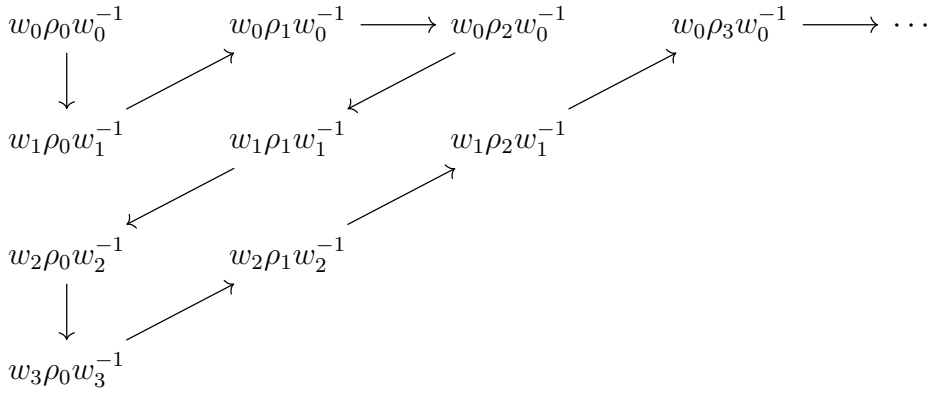
**Definition 5.29.** Let $G$ be a group with presentation $\langle x_1, \ldots, x_n \mid R\rangle$. Then $G$ has a **decidable word problem** if the set $\{w \in (X^{\pm})^* : w = 1 \text{ in } G\}$ is recursive.

As outlined in the preceding definition, demonstrating the decidability of the word problem in a group $G$ requires proving the recursiveness of a particular set, a task that can prove challenging due to its inherent complexity. With the introduction of the following theorem, we can streamline this definition and simplify the verification process. Specifically, we can reduce the task to demonstrating that a specific set is recursively enumerable, which offers a more manageable and efficient approach to establishing the decidability of the word problem in $G$.

**Theorem 5.30.** Let $G$ be a finitely presented group with presentation $\langle X \mid R\rangle$. The set $E = \{w \in (X^{\pm})^* : w = 1 \text{ in } G\}$ is recursively enumerable.

*Proof.* We can prove this statement by systematically enumerating the elements of $E$. To do so, we start by listing all words $w_0, w_1, \ldots$ in $(X^{\pm})^*$, organised first by length and then lexicographically. Similarly, we list all the words on $R$ as $\rho_0, \rho_1, \ldots$. Next, we construct

the following array:




where the rows correspond to words $w_i$ and the columns correspond to relators $\rho_j$. We then follow the arrows in the table to systematically enumerate the set $E$. We repeat this process until all elements of $E$ are enumerated. Since we have outlined a systematic procedure to enumerate $E$, it follows that $E$ is recursively enumerable. □

By Theorem 5.30, it is only necessary to verify if the set $\{w \in (X^{\pm})^* : w \neq 1 \text{ in } G\}$ is recursively enumerable. Therefore, we can redefine Definition 5.29 as follows.

**Definition 5.31.** A finitely presented group $G = \langle X \mid R\rangle$ has a **decidable word problem** if and only if the set $\{w \in (X^{\pm})^* : w \neq 1 \text{ in } G\}$ is recursively enumerable.

We will adopt the following notation: for any two (not necessarily reduced) words, $w$ and $w'$, formed from an alphabet $(X^{\pm})^*$, we denote

$$w \equiv w'$$

if they have the same spelling.

Consider a semigroup $\gamma = \langle X \mid \alpha_j = \beta_j, j \in J \subset \mathbb{N}\rangle$. For $w, w' \in X^+$, then $w = w'$ in $\gamma$ if and only if there exists a finite sequence

$$w \equiv w_1 \to w_2 \to \cdots \to w_t \equiv w',$$

where each $w_i \to w_{i+1}$ represents an elementary operation. These operations correspond to transitions from $w$ to $w'$ by replacing a subword satisfying one of the relations $\alpha_j = \beta_j$ with its equivalent subword. In the context of a Turing machine, an elementary operation is defined as either $w_i \equiv \sigma\alpha_j\tau$ and $w_{i+1} \equiv \sigma\beta_j\tau$ or $w_{i+1} \equiv \sigma\alpha_j\tau$ and $w_i \equiv \sigma\beta_j\tau$ for some $j$, where $\sigma$ and $\tau$ are positive words on $X$.

We will now describe Emil Post's construction of a finitely presented semigroup that serves as a representation of a Turing machine $T$, as detailed in [47]. To streamline our notation, we adopt the convention that the $s$-letters and $q$-letters utilised in the quadruples of $T$ are represented by $s_0, s_1, \ldots, s_M$ and $q_0, q_1, \ldots, q_N$, respectively. Additionally, we introduce two new letters, denoted as $q$ and $h$, to aid in this construction.

**Definition 5.32.** If $T$ is a Turing machine having stopping state $q_0$, then its **associated semigroup** $\gamma(T)$ has the presentation:

$$\gamma(T) = \langle q, h, s_0, s_1, \ldots, s_M, q_0, q_1, \ldots, q_N \mid R(T)\rangle$$

where the relations $R(T)$ are

$$q_i s_j = q_l s_k \quad \text{if} \quad q_i s_j s_k q_l \in T,$$

and for all $\beta = 0, 1, \ldots, M$:

$$\begin{aligned} q_i s_j s_\beta &= s_j q_i s_\beta \quad \text{if } q_i s_j R q_i \in T, \\ q_i s_j h &= s_j q_i s_0 h \quad \text{if } q_i s_j R q_i \in T, \\ s_\beta q_i s_j &= q_i s_\beta s_j \quad \text{if } q_i s_j L q_i \in T, \\ h q_i s_j &= h q_i s_0 s_j \quad \text{if } q_i s_j L q_i \in T, \end{aligned}$$

$$\begin{aligned} q_0 s_\beta &= q_0, \\ s_\beta q_0 h &= q_0 h, \\ h q_0 h &= q. \end{aligned}$$

The first five relations directly correspond to the basic moves of the Turing machine, the new letter $h$ enables us to distinguish basic move (2) from basic move (3) and to distinguish basic move (4) from basic move (5). We can interpret $h$ as marking the ends of the tape, which is relevant for certain expressions. The last three relations describe what happens when the machine reaches its halting state.

**Definition 5.33.** A word is **$h$-special** if it has the form $h\alpha h$, where $\alpha$ is an instantaneous description.

Since $T$ has halt state $q_0$, every $h\alpha h$ expression (with $\alpha$ being terminal) follows the format $h\sigma q_0 \tau h$, where $\sigma$ and $\tau$ are $s$-words, and $\tau$ is not empty. Consequently, the last three relations allows us to write $h\alpha h = q$ in $\gamma(T)$ whenever $\alpha$ is terminal.

The concept of an $h$-special word enhances our understanding of the final three relations outlined in Definition 5.32. When the Turing machine reaches its halting state $q_0$, there may still be symbols on the tape to the right of this point. The existence of the last three types of defining relations in $\gamma(T)$ is to manage any $h$-special word $h\alpha h$ that corresponds to a halted computation of $T$ by systematically removing symbols. Initially, all symbols to the right of $q_0$ are erased, leaving only those directly involved in the computation up to that point. Then, symbols to the left of $q_0 h$ are also erased, recognising that they may represent the initial setup before the computation began. This selective removal ensures that we focus solely on symbols indicating the result of the computation. Finally, the machine transitions to its final state $q$ without the $h$ symbols, effectively clearing the tape and ending the computation. After the computation is completed, all symbols on the tape are systematically erased, ensuring a clean tape for future operations. This process guarantees the extraction of the computed result and the orderly termination of the Turing machine's operation, simplifying the tape to include only relevant symbols.

We now present two technical lemmas essential for proving the main theorem in this section.

**Lemma 5.34.** Let $T$ be a Turing machine with stopping state $q_0$ and associated semigroup

$$\gamma(T) = \langle q, h, s_0, s_1, \ldots, s_M, q_0, q_1, \ldots, q_N \mid R(T) \rangle.$$

1. Let $w$ and $w'$ be words on $\{s_0, s_1, \ldots, s_M, q_0, q_1, \ldots, q_N\}$ with $w \not\equiv q$ and $w' \not\equiv q$. If $w \to w'$ is an elementary operation, then $w$ is $h$-special if and only if $w'$ is $h$-special.

2. If $w = h\alpha h$ is $h$-special, $w' \not\equiv q$, and $w \to w'$ is an elementary operation of one of the first five types, then $w' \equiv h\beta h$, where either $\alpha \to \beta$ or $\beta \to \alpha$ is a basic move of $T$.

*Proof.* We prove each statement separately.

1. Suppose $w$ is $h$-special, meaning $w = h\alpha h$ for some word $\alpha$. Since $w \to w'$ is an elementary operation, the only relation in $\gamma(T)$ that adds or removes $h$ is $hqh = q$. Thus, $w'$ must also have the form $h\beta h$ for some word $\beta$. Hence, $w'$ is $h$-special. Conversely, if $w'$ is $h$-special, the same argument applies, and $w$ must be $h$-special.
2. An elementary move in $\gamma(T)$ corresponds to a substitution using an equation in a defining relation. Such a relation in $\gamma(T)$ of one of the first five types corresponds to a quadruple of $T$, and a quadruple corresponds to a basic move. Therefore, either $\alpha \to \beta$ or $\beta \to \alpha$. □

**Lemma 5.35.** Let $T$ be a Turing machine with halt state $q_0$, let $S^+$ be the set of all positive words on the alphabet of $T$, and let $E = e(T)$. If $w \in S^+$ then

$$w \in E \iff hq_1wh = q \quad \text{in } \gamma(T).$$

*Proof.* We prove each direction in turn.

Proof of ($\Rightarrow$). If $w \in E$, then there exist a sequence of instantaneous descriptions $\alpha_1 = q_1w, \alpha_2, \ldots, \alpha_t$, where each transition $\alpha_i \to \alpha_{i+1}$ leads to a terminal state $\alpha_t$ involving the $q_0$ state. By applying the elementary operations of the first five types, we can demonstrate that $hq_1wh = h\alpha_t h$ in $\gamma(T)$. Since $\alpha_t = vq_0w$, where $v \in S^+$, repeated use of the relation $q_0s_\beta = q_0$ yields $h\alpha_t h = hvq_0h$ in $\gamma(T)$. Subsequent application of the relation $s_\beta q_0h = q_0h$ results in $hvq_0h = hq_0h$ in $\gamma(T)$. As $hq_0h = q$ is also a relation, we deduce the following chain of equalities:

$$hq_1wh = h\alpha_t h = hvq_0h = hq_0h = q$$

in $\gamma(T)$.

Proof of ($\Leftarrow$). The proof for sufficiency follows a different approach compared to the necessity proof, primarily due to the asymmetric nature of basic moves in the Turing machine's associated semigroup. While equality in $\gamma(T)$ is symmetric, $\alpha \to \beta$ being a basic move does not imply that $\beta \to \alpha$ is a basic move.

Given $hq_1wh = q$ in $\gamma(T)$, we can express this equality through a sequence of words $w_1, w_2, \ldots, w_t$ on $\{h, s_0, s_1, \ldots, s_M, q_0, q_1, \ldots, q_N\}$ involving elementary operations:

$$hq_1wh = w_1 \to w_2 \to \cdots \to w_t = hq_0h \to q.$$

Utilising Lemma 5.34(1.), we establish that each $w_i$ is $h$-special, denoted as $w_i \equiv h\alpha_i h$, where $\alpha_i$ represents an instantaneous description. According to Lemma 5.34(2.), for each $i$, either $\alpha_i \to \alpha_{i+1}$ or $\alpha_{i+1} \to \alpha_i$.

We must ensure that all arrows move rightward to validate the computation path of Turing machine $T$. These arrows represent transitions, ensuring a consistent progression through computation. This guarantees that the sequence $q_1w \to \alpha_2 \to \cdots \to \alpha_t$ forms a series of basic moves, with $\alpha_t$ being terminal (since it involves the halt state $q_0$). Thus, $w$ is admissible in $T$ and $w \in E$.

To demonstrate that all arrows progress to the right, we employ induction on $t \geq 2$. That is for all $i \leq t-1$ we have $\alpha_i \to \alpha_{i+1}$. Notably, when $\alpha_t$ is terminal, $\alpha_{t-1} \to \alpha_t$ always holds, precluding the occurrence of $\alpha_{t-1} \leftarrow \alpha_t$, thus initiating the induction at $t = 2$. Suppose $t > 2$ and an arrow points left. Tracing back from the last arrow $\alpha_{t-1} \to \alpha_t$ until encountering an arrow pointing left yields an index $i$ such that:

$$\alpha_{i-1} \leftarrow \alpha_i \to \alpha_{i+1}.$$

But there is never ambiguity about the next move of a Turing machine, thus $\alpha_{i-1} \equiv \alpha_{i+1}$, implying $w_{i-1} \equiv h\alpha_{i-1}h \equiv h\alpha_{i+1}h \equiv w_{i+1}$. Consequently, $w_i$ and $w_{i+1}$ can be eliminated, thereby reducing the number of steps $t$. The proof is thus completed by induction. □

We are now prepared to establish the undecidability of the word problem for finitely presented semigroups. Achieving this goal simply entails providing an example of a finitely presented semigroup with an undecidable word problem.

**Theorem 5.36** (Markov-Post theorem [50, Theorem 12.5])
We have the following.

1. There is a finitely presented semigroup
$$\gamma_{\text{swp}} = \langle q, h, s_0, s_1, \ldots, s_M, q_0, q_1, \ldots, q_N \mid R \rangle$$
with an undecidable word problem.
2. There is no algorithm which determines, for an arbitrary $h$-special word $h\alpha h$, whether $h\alpha h = q$ in $\gamma$.

*Proof.* We prove each statement separately.

1. Let $T$ denote a Turing machine with halt state $q_0$ and let $A = \{s_0, s_1, \ldots, s_M\}$ be its alphabet. Let $A^+$ represent all positive words derived from $A$, and define $E = e(T) \subset A^+$. To expand the alphabet, we introduce $\widetilde{A}^+$, encompassing all positive words from $A \cup \{q, h, q_0, q_1, \ldots, q_N\}$, where $q_0, q_1, \ldots, q_N$ denote the $q$-letters occurring within the quadruples of $T$. Our focus lies on elements of the semigroup, encoded within the Turing machine following its halted computation. These are precisely the words in the set:
$$\widetilde{E} = \left\{ \widetilde{w} \in \widetilde{A}^+ : \widetilde{w} = q \text{ in } \gamma(T) \right\}.$$
This set also encompasses elements of the semigroup that satisfy the specified condition, yet they might not have any meaningful interpretation within the context of a Turing machine. To examine these elements as words within the Turing machine, we define the mapping:
$$\begin{aligned} \phi : A^+ &\to \widetilde{A}^+ \\ w &\mapsto hq_1wh. \end{aligned}$$
We denote the image of $A^+$ and $E$ under $\phi$ as $\phi(A^+) = A_1$ and
$$\phi(E) = E_1 = \{hq_1wh : w \in E\}.$$

Now, $E_1$ represents precisely the set of admissible words for a Turing machine. We note that $E_1$ is a recursive subset of $A_1$ if and only if $E$ is a recursive subset of $A^+$. In this notation, Lemma 5.35 can be restated as:

$$E_1 = \widetilde{E} \cap A_1.$$

Assume $T$ is the Turing machine $T^*$ as defined in Theorem 5.27, so that $E$ and therefore $E_1$ are recursively enumerable but not recursive. If $\widetilde{E}$ were recursive, then by Lemma 5.25, it would imply $E_1$ is recursive, hence $E$ is also recursive, leading to a contradiction. Therefore, $\gamma_{\text{swp}} = \gamma(T^*)$ exhibits an undecidable word problem.

2. Let $F = \{h\alpha h : h\alpha h = q \text{ in } \gamma(T^*)\}$. If $F$ were a recursive subset of $\widetilde{A}^+$, then according to Lemma 5.25, $F \cap A_1$ would also be a recursive subset of $A_1$. However, we have established that $F \cap A_1 = E_1$, which we have shown is not a recursive subset of $A_1$. Therefore, $F$ cannot be recursive. □

**Example 5.37.** For an explicit example of a semigroup with an undecidable word problem refer to [60].

We introduce the following corollary, which allows us to reformulate the generators and relations of the Markov-Post semigroup $\gamma_{\text{swp}}$. This reformulation will be used in subsequent discussions.

**Corollary 5.38** ([50, Corollary 12.6])
We can restate Theorem 5.36 as follows.

1. There is a finitely presented semigroup

$$\gamma = \langle q, q_0, \ldots, q_N, s_0, \ldots, s_M \mid F_i q_{i_1} G_i = H_i q_{i_2} K_i, i \in I \subset \mathbb{N} \rangle,$$

with an undecidable word problem, where $F_i, G_i, H_i, K_i$ are (possibly empty) positive $s$-words and $q_{i_1}, q_{i_2} \in \{q, q_0, \ldots, q_N\}$.

2. There is no algorithm which determines, for arbitrary $q_{i_j}$ and positive $s$-words $W$ and $V$, whether $Wq_{i_j}V = q$ in $\gamma$.

*Proof.* We prove each statement.

1. Consider the generator $h$ of the semigroup specified in Theorem 5.36 as the final $s$-letter, and re-index these $s$-letters accordingly so that $h = s_M$. The rewritten relations in $R(T^*)$ now conform to the described format.

2. Let $A_2$ represent the set of all positive words on the rewritten generators of $\gamma$. We define the following sets:

$$\Lambda = \{Wq_{i_j}V : W, V \text{ are positive words on the rewritten } s\text{-letters and satisfy } Wq_{i_j}V = q \text{ in } \gamma\}$$

and

$$\widetilde{F} = \{s_M \alpha s_M : \alpha \equiv \sigma q_{i_j} \tau, \text{ where } \sigma \text{ and } \tau \text{ are positive words on } \{s_0, \ldots, s_{M-1}\} \text{ and } s_M \alpha s_M = q \text{ in } \gamma\}$$

Notice that $\widetilde{F}$ represents the set $F$ from Theorem 5.36 but rewritten in the new notation. Thus, $\Lambda \cap A_2 = \widetilde{F}$. According to Lemma 5.25, if both $\Lambda$ and $A_2$ are recursive, then $\widetilde{F}$ is also recursive. However, since $\widetilde{F}$ is known not to be recursive, it follows that $\Lambda$ cannot be recursive. □

## 5.4 The Novikov-Boone Britton Theorem

In this section, we present the proof of the undecidability of the word problem for a finitely presented group. It is crucial to recognise that the proof aims to reduce the equality of words in a group to the equality of words in a semigroup. Therefore, it is imperative to carefully track exponents, as whilst arbitrary words are meaningful in a group, only positive words hold significance in a semigroup.

We will employ the following notation: if $W \equiv s_{\beta_1}^{e_1} \cdots s_{\beta_m}^{e_m}$ represents a (not necessarily positive) $s$-word, then $W^{\#} \equiv s_{\beta_1}^{-e_1} \cdots s_{\beta_m}^{-e_m}$. We note that for any $s$-words $W$ and $V$, we have

$$(W^{\#})^{\#} \equiv W \quad \text{and} \quad (WV)^{\#} \equiv W^{\#}V^{\#}.$$

By Corollary 5.38, we have developed a method to construct a semigroup that faithfully emulates a Turing machine. Specifically, for any given Turing machine, there exists a corresponding semigroup $\gamma = \gamma(T)$ defined by the following presentation:

$$\gamma = \langle q, q_0, \ldots, q_N, s_0, \ldots, s_M \mid F_i q_{i_1} G_i = H_i q_{i_2} K_i, i \in I \subset \mathbb{N} \rangle$$

Here, $F_i$, $G_i$, $H_i$, and $K_i$ denote (potentially empty) positive $s$-words, while $q_{i_1}$ and $q_{i_2}$ belong to the set $\{q, q_0, \ldots, q_N\}$.

However, to advance this emulation into the realm of groups, we must incorporate the corresponding inverse element to fulfil the group axioms. Regrettably, a direct strategy of simply adding inverses to each element within the semigroup falls short in achieving its transformation into a group. Through the utilisation of the theory of HNN extensions,[5] we can devise relations that extend the semigroup into a group. Throughout this extension process, we introduce additional elements and impose relations to guarantee coherence with the desired group properties.

The group $\mathscr{B}(T)$ is presented with generators $q, q_0, \ldots, q_N, s_0, \ldots, s_M, r_i$ where $i \in I \subset \mathbb{N}$ and $x, t, k$, alongside the following relations:

$$\begin{aligned}
x s_\beta &= s_\beta x^2, \\
r_i s_\beta &= s_\beta x r_i x, \\
r_i^{-1} F_i^{\#} q_{i_1} G_i r_i &= H_i^{\#} q_{i_2} K_i, \\
t r_i &= r_i t, \\
t x &= x t, \\
k r_i &= r_i k, \\
k x &= x k, \\
k(q^{-1} t q) &= (q^{-1} t q) k,
\end{aligned}$$

[5] These are a construction that allows for the extension of a group by adding a stable letter and imposing relations that mimic the behaviour of elements in a chosen subgroup, thereby creating a larger group with specific interactions between its components [21].

where, $i \in I$ and $\beta = 0, \ldots, M$.

For each Turing machine $T$, we establish a corresponding group $\mathscr{B}(T)$. Our aim is to demonstrate that by selecting $T$ as the Turing machine $T^*$ described in Theorem 5.36, the word problem of the group $\mathscr{B}(T)$ is undecidable.

Before proceeding with the proof, it is essential to introduce some additional terminology. For any $s$-words $W$ and $V$, we introduce the notation

$$(Wq_jV)^* = W^{\#}q_jV,$$

where $q_j \in \{q, q_0, \ldots, q_N\}$.

**Definition 5.39.** A word $\Sigma$ as **special** if it can be expressed in the form $W^{\#}q_jV$, where $W$ and $V$ are positive $s$-words, and $q_j \in \{q, q_0, \ldots, q_N\}$.

Given a special word $\Sigma = W^{\#}q_jV$, where $W$ and $V$ are positive $s$-words and $q_j$ is chosen from the set $\{q, q_0, \ldots, q_N\}$, we have that $\Sigma^* \equiv \left(W^{\#}q_jV\right)^* \equiv Wq_jV$. This process effectively transforms the special word $\Sigma$ into a positive word $\Sigma^*$. As a positive word, $\Sigma^*$ represents an element within the semigroup $\gamma$, thereby establishing a direct link between special words $\Sigma$ and elements within the semigroup $\gamma$.

We shall now present a lemma essential for proving the undecidability of the word problem in finitely presented groups.

**Lemma 5.40** (Boone's lemma [50, Lemma 12.7])
Let $T$ be a Turing machine with halt state $q_0$ and associated semigroup $\gamma = \gamma(T)$ (rewritten as in Corollary 5.38). If $\Sigma$ is a special word, then

$$k\left(\Sigma^{-1}t\Sigma\right) = \left(\Sigma^{-1}t\Sigma\right)k \quad \text{in } \mathscr{B}(T)$$

if and only if $\Sigma^* = q$ in $\gamma(T)$.

We now present the theorem that showcases the undecidability of the word problem for finitely presented groups. We reiterate that demonstrating this undecidability simply requires finding an example of a finitely presented group with an undecidable word problem.

**Theorem 5.41** (Novikov-Boone-Britton theorem [50, Theorem 12.8])
There exists a finitely presented group with an undecidable word problem.

*Proof.* Let $T$ be the Turing machine $T^*$ introduced in Theorem 5.36. If there existed an algorithm capable of determining, for any arbitrary special word $\Sigma$, whether

$$\begin{aligned} k\left(\Sigma^{-1}t\Sigma\right) = \left(\Sigma^{-1}t\Sigma\right)k &\iff k\Sigma^{-1}t\Sigma\left(\Sigma^{-1}t\Sigma k\right)^{-1} &&= 1 \\ &\iff k\Sigma^{-1}t\Sigma k^{-1}\Sigma^{-1}t^{-1}\Sigma &&= 1 \end{aligned}$$

in $\mathscr{B}(T)$, then by Lemma 5.40, the same algorithm would determine whether $\Sigma^* = q$ in $\gamma(T^*)$. However, Corollary 5.38(2) affirms that no such algorithm exists for $\gamma(T^*)$. □

**Example 5.42.** A simpler example of a group with an undecidable word problem is provided in [10].

## 5.5 Boone's lemma

The proof of Boone's lemma (Lemma 5.40) is extensive and complex, which require ideas that have not been discussed in this thesis. Therefore, we opt to present the proof of sufficiency in full, and the remainder of the proof can be found in [50, Pages 433–447].

To establish the sufficiency of Boone's lemma, we rely on the following auxiliary lemma.

**Lemma 5.43.** We have the following.

1. If $V$ is a positive $s$-word, then
$$r_iV = VR \text{ in } \mathscr{B} \quad \text{and} \quad r_i^{-1}V = VR' \text{ in } \mathscr{B},$$
where $R$ and $R'$ are words on $\{r_i, x\}$ with $R$ a positive word.
2. If $U$ is a positive $s$-word, then
$$U^{\#}r_i = LU^{\#} \text{ in } \mathscr{B} \quad \text{and} \quad U^{\#}r_i^{-1} = L'U^{\#} \text{ in } \mathscr{B},$$
where $L$ and $L'$ are words on $\{r_i, x\}$.

*Proof.* We will demonstrate the proof for one of the equalities, as the arguments for the remaining equations follow a similar pattern.

We establish the equality $r_iV = VR$ in $\mathscr{B}$ through induction on $m \geq 0$, where the word $V \equiv s_{\beta_1} \dots s_{\beta_m}$. For the base case $m = 0$, where $V$ is the empty word, we have $r_i = R$, since $R$ is a word on $\{r_i, x\}$. Thus, we can choose $R = r_i$, satisfying the equality for $m = 0$. For $m > 0$, we express $V$ as $V's_{\beta_m}$. Utilising the inductive hypothesis, we can write $r_iV \equiv r_iV's_{\beta_m} = V'R's_{\beta_m}$, where $R'$ forms a positive word on $\{r_i, x\}$. Through the application of the relations $xs_\beta = s_\beta x^2$ and $r_is_\beta = s_\beta r_i x$, we deduce the existence of a positive word $R$ on $\{r_i, x\}$ such that $s_{\beta_m}R = R's_{\beta_m}$ holds in $\mathscr{B}$. Consequently, we can express $r_iV \equiv r_iV's_{\beta_m} = V'R's_{\beta_m} = V's_{\beta_m}R$. □

Recall our objective: to establish that for a special word $\Sigma$ such that $\Sigma^* = q$ in $\gamma(T)$, the equation $k\,(\Sigma^{-1}t\Sigma) = (\Sigma^{-1}t\Sigma)\,k$ holds in $\mathscr{B}(T)$.

*Proof of sufficiency in Boone's lemma.* If $\Sigma$ is a special word such that $\Sigma^* \equiv X_{q_j}Y = q$ in $\gamma$, then there exists a sequence of elementary operations:
$$\Sigma^* \equiv w_1 \to w_2 \to \cdots \to w_n \equiv q \quad \text{in } \gamma.$$
Since $w_v$ for some $v$ in an indexing set, are words in the semigroup $\gamma$, unless they are empty they must contain relations as described in Corollary 5.38 thus we can write the words $w_v$ and $w_{v+1}$ as $UF_{q_{i_1}}G_iV$ and $UH_{q_{i_2}}K_iV$ respectively with $U$ and $V$ being positive $s$-words. The only relation of this form in the group $\mathscr{B}$ is
$$r_i^{-1}F_i^{\#}q_{i_1}G_ir_i = H_i^{\#}q_{i_2}K_i$$
By Lemma 5.43, there exist equations in $\mathscr{B}$ such that:
$$U^{\#}(H_i^{\#}q_{i2}K_i)V = U^{\#}(r_i^{-1}F_i^{\#}q_{i1}G_ir_i)V = L'U^{\#}(F_i^{\#}q_{i1}G_i)VR',$$
where $L'$ and $R'$ are words on $\{r_i, x\}$. Similarly, there exist words $L''$ and $R''$ on $\{r_i, x\}$ such that:
$$\begin{aligned} U^{\#}(F_i^{\#}q_{i_1}G_i)V &= U^{\#}(r_iH_i^{\#}q_{i_2}K_ir_i^{-1})V \\ &= L''U^{\#}(H_i^{\#}q_{i_2}K_i)VR''. \end{aligned}$$

Since $w_v = w_{v+1}$ in $\gamma$ implies $w_v^* = w_{v+1}^*$ in $\mathscr{B}$, we can infer, for each $v$, that:

$$w_v^* = L_v w_{v+1}^* R_v \text{ in } \mathscr{B}$$

where $L_v$ and $R_v$ are words on some $r_i$ and $x$. Combining these relations, we obtain:

$$w_1^* = L w_n^* R \text{ in } \mathscr{B}.$$

However, as $w_1^* = (\Sigma^*)^* = \Sigma$ and $w_n^* = q^* = q$, it follows that:

$$\Sigma = LqR \text{ in } \mathscr{B}.$$

Since the generators $t$ and $k$ commute with $x$ and all the $r_i$, they also commute with $L$ and $R$. Hence,

$$\begin{aligned} k\Sigma^{-1}t\Sigma k^{-1}\Sigma^{-1}t^{-1}\Sigma &= k\left(R^{-1}q^{-1}L^{-1}\right)t(LqR)k^{-1}\left(R^{-1}q^{-1}L^{-1}\right)t^{-1}(LqR) \\ &= kR^{-1}q^{-1}tqk^{-1}q^{-1}t^{-1}qR \\ &= R^{-1}(kq^{-1}tqk^{-1}q^{-1}t^{-1}q)R \\ &= 1, \end{aligned}$$

because, it follows from the last relation of $\mathscr{B}$ that $kq^{-1}tqk^{-1}q^{-1}t^{-1}q = 1$. □

# 6 Concluding remarks

We extend our gratitude to the reader for accompanying us on this exploration of the decidability of the word and conjugacy problem within this thesis. As we conclude, it becomes evident that numerous avenues exist for further expansion. A natural progression from our current inquiries lies in the exploration of the third problem proposed by Dehn: the isomorphism problem [12, Das Isomorphieproblem]; which delves into the existence of algorithms capable of determining if two finitely presented groups are isomorphic. Despite being more challenging than the previous decision problems he posed, it has been established as undecidable. Based on Novikov's research, Sergei I. Adian (1957) proved the undecidability of isomorphism problem for any given finitely presented group. Moreover, he demonstrated the algorithmic undecidability of numerous Markov properties pertaining to finitely presented groups [1]. Subsequently, Michael O. Rabin (1958) extended these findings, proving the algorithmic undecidability of every Markov property within a finitely presented group [48]. These contributions collectively form what is now recognised as the Adian-Rabin theorem [45].

In the fundamental groups Dehn was considering, the isomorphism problem is geometrically intertwined with the homeomorphism problem. This is because homeomorphic spaces have isomorphic fundamental groups. However, the fact that two fundamental groups are isomorphic does not necessarily imply that the corresponding spaces are homeomorphic. For example, although the fundamental groups of a point and a sphere are isomorphic since they are both trivial, it is clear that a sphere is not homeomorphic to a point. Therefore, the two problems are not equivalent, but the homeomorphism problem can be utilised to tackle the isomorphism problem. For a more extensive exposition of all the decision problems proposed by Dehn, we suggest consulting [41].

In addition to the isomorphism and homeomorphism problems, the homotopy equivalence problem plays a crucial role in understanding fundamental groups. The concept of homotopy equivalence delves into whether two spaces can be continuously deformed into each other without 'tearing' or 'gluing', thus preserving their topological properties. If two spaces are homotopy equivalent, their fundamental groups are isomorphic. This insight allows us to infer similarities in the topology of spaces by examining their fundamental groups.

However, it is important to note that while two spaces may be homotopy equivalent, this relationship does not necessarily imply they are homeomorphic. For example, consider the unit circle and a single point in space. They are homotopy equivalent because one can be continuously deformed into the other, effectively contracting the circle to a point. However, they are not homeomorphic. This is because any mapping from a circle to a point must be a constant map, which is not bijective. In other words, we can deform a circle into a point, but we cannot deform a point into a circle. For further insight into the homotopy equivalence problem, please refer to [53].

The core concepts driving the formulation of the algorithms discussed in this thesis are simple: enumerate all elements and methodically compare each one with every other to establish equality or conjugacy. However, the proofs underpinning these algorithms are notably intricate. Further exploration could focus on examining the time complexity of these algorithms, alongside comparisons with other well-established methods [52].

We can further explore the geometric properties of Cayley graphs associated with small

cancellation groups, a field initially explored by Mikhail Gromov [19]. The theory of graphical small cancellation serves as a powerful tool for constructing groups with specific geometric and analytical characteristics, facilitating the examination of their geometric properties.

Moreover, these groups often exhibit traits reminiscent of hyperbolic geometry. Hyperbolic groups, similar to hyperbolic spaces, possess properties of negative curvature [3]. A notable aspect of hyperbolic groups is the fact that they satisfy small cancellation criteria [25], contributing to their complex geometric makeup. By analysing the interplay between small cancellation conditions and hyperbolicity, we glean valuable insights into the geometric essence of these groups and the structures of their associated Cayley graphs.

Dehn's contributions lay the groundwork for the construction of decision problems within group theory, specifically by questioning whether algorithms exist to determine certain properties within or for groups. In contrast, Hilbert and Ackermann's Entscheidungsproblem a similar foundation for decision problems across the entirety of mathematics. This approach allows for the exploration of additional decision problems within group theory, such as the 'subgroup membership problem', which asks whether an algorithm exists to decide if a given element from a group is also a member of one of its specified subgroups [33].

In conclusion, the extensive investigation into decision problems within group theory underscores the breadth and depth of this field of study. Through exploring questions about equations, computational complexity, and structural properties of groups, there are a myriad of fascinating insights into the nature of groups and their applications across diverse mathematical domains. While many decision problems in group theory remain challenging and unsolved, the ongoing pursuit of solutions continues to drive innovation and discovery. As we navigate the complexities of these problems, we deepen our understanding of group structures and their significance in mathematics and beyond. Ultimately, the exploration of decision problems in group theory not only enriches our theoretical understanding but also paves the way for practical advancements in cryptography, computational mathematics, and other related fields. By leveraging decision problems, we can devise innovative cryptographic protocols [42], while in computational mathematics [23], we can pioneer new algorithms to ascertain the structure of large groups, an example of this is the Todd-Coxeter algorithm, which is used to enumerate cosets [57].

# Appendix

# A Implementation of the algorithms

In our algorithms, we adopt a specific representation for words. A word is expressed in the form:

$$w = x_{i_1}^{\varepsilon_1} x_{i_2}^{\varepsilon_2} \cdots x_{i_k}^{\varepsilon_k}$$

where $x_{i_j} \in X^{\pm}$ and $\varepsilon_h \in \{-1, 1\}$. We represent this word using an array:

$$w = [[i_1, \varepsilon_1], [i_2, \varepsilon_2], \ldots, [i_k, \varepsilon_k]],$$

referred to as **pair notation**. We have opted for this representation in Python because using letters from the alphabet to represent the generators of the free group would impose restrictions on the maximum number of generators allowed in our group.

In this coding scheme, uppercase letters represent the inverse elements. For instance, the inverse of $x_i$ is denoted as $X_i$. When prompted to input words using this code, each letter must be separated by whitespace. For example, if we wish to input the word $w = x_1^{-1}x_2x_3^{-1}x_4$, it should be written as: `X1 x2 X3 x4`.

The algorithms presented in the following subsections are also available in [8] for convenient access.

## A.1 Word problem in free groups of finite rank

The author coded Algorithm 1 to decide if two given words are equal.

```
    # Define a function to convert a string to pair notation
def string_to_pair_notation(input_string):
    # Initialise an empty list to store pair notation
    pair_notation = []

    # Loop through each term in the input string (terms are
   separated by spaces)
    for term in input_string.split():
        # Extract the generator (first character of the term)
        generator = term[0]
        # Extract the index (remaining characters of the term
   converted to an integer)
        index = int(term[1:])

        # Check if the generator is lowercase
        if generator.islower():
            # If lowercase, append a pair with index and 1 to
   pair_notation
            pair_notation.append([index, 1])
        else:
            # If uppercase, append a pair with index and -1 to
   pair_notation
            pair_notation.append([index, -1])

    # Return the final pair notation list
```

```
    return pair_notation

# Define a function to reduce a word represented in pair notation
def reduce_word(pair_notation):
    # Initialise an empty list to store the reduced word in pair notation
    reduced_word = []

    # Loop through each pair in the pair notation
    for pair in pair_notation:
        # Check if the reduced word is empty or if the current pair's index is different from the last one
        if not reduced_word or reduced_word[-1][0] != pair[0]:
            # If so, append the current pair to the reduced word
            reduced_word.append(pair)
        else:
            # If the index is the same, check if the sum of the exponents is zero
            if reduced_word[-1][1] + pair[1] == 0:
                # If zero, cancel out and pop the last pair from the reduced word
                reduced_word.pop()
            else:
                # If not zero, append the current pair to the reduced word
                reduced_word.append(pair)

    # Return the final reduced word in pair notation
    return reduced_word

# Define a function to check if two words represented in pair notation are equal
def are_words_equal(word1, word2):
    # Convert both words to pair notation
    pair_notation_1 = string_to_pair_notation(word1)
    pair_notation_2 = string_to_pair_notation(word2)

    # Reduce both pair notations
    reduced_form_1 = reduce_word(pair_notation_1)
    reduced_form_2 = reduce_word(pair_notation_2)

    # Check if the reduced forms are equal
    return reduced_form_1 == reduced_form_2

# Example usage:
word1 = input("Enter the first word (e.g., 'x1 X3 X2 x3 x1 x1'): ")
word2 = input("Enter the second word: ")

result = are_words_equal(word1, word2)
```

```
print(f"The words '{word1}' and '{word2}' are {'equal' if result else 'not equal'}.")
```

Algorithm 1: Word problem in free groups of finite rank

## A.2 Conjugacy problem in free groups of finite rank

The author coded Algorithm 2 to decide if two given words are conjugate.

```
    # Define a function to convert a string to pair notation
def string_to_pair_notation(input_string):
    # Initialise an empty list to store pair notation
    pair_notation = []

    # Loop through each term in the input string (terms are separated by spaces)
    for term in input_string.split():
        # Extract the generator (first character of the term)
        generator = term[0]
        # Extract the index (remaining characters of the term converted to an integer)
        index = int(term[1:])

        # Check if the generator is lowercase
        if generator.islower():
            # If lowercase, append a pair with index and 1 to pair_notation
            pair_notation.append([index, 1])
        else:
            # If uppercase, append a pair with index and -1 to pair_notation
            pair_notation.append([index, -1])

    # Return the final pair notation list
    return pair_notation

# Define a function to reduce a word represented in pair notation
def reduce_word(pair_notation):
    # Initialise an empty list to store the reduced word in pair notation
    reduced_word = []

    # Loop through each pair in the pair notation
    for pair in pair_notation:
        # Check if the reduced word is empty or if the current pair's index is different from the last one
        if not reduced_word or reduced_word[-1][0] != pair[0]:
            # If so, append the current pair to the reduced word
            reduced_word.append(pair)
        else:
            # If the index is the same, check if the sum of the exponents is zero
```

```
            if reduced_word[-1][1] + pair[1] == 0:
                # If zero, cancel out and pop the last pair from the reduced word
                reduced_word.pop()
            else:
                # If not zero, append the current pair to the reduced word
                reduced_word.append(pair)

    # Return the final reduced word in pair notation
    return reduced_word

# Define a function to cyclically reduce a word represented in pair notation
def cyclically_reduce(pair_notation):
    # Continue removing pairs from the end of the pair notation until a different index is encountered
    while pair_notation:
        last_pair = pair_notation[-1]
        pair_notation.pop()
        if not pair_notation or pair_notation[-1][0] != last_pair[0]:
            break

    # Return the cyclically reduced pair notation
    return pair_notation

# Define a function to check if two words are conjugate
def are_words_conjugate(word1, word2):
    # Convert the input words to pair notation
    pair_notation_1 = string_to_pair_notation(word1)
    pair_notation_2 = string_to_pair_notation(word2)

    # Reduce the words to their canonical form in pair notation
    reduced_form_1 = reduce_word(pair_notation_1)
    reduced_form_2 = reduce_word(pair_notation_2)

    # Cyclically reduce the words in pair notation
    cyclically_reduced_1 = cyclically_reduce(reduced_form_1)
    cyclically_reduced_2 = cyclically_reduce(reduced_form_2)

    # Check if the cyclically reduced pair notations are equal
    return cyclically_reduced_1 == cyclically_reduced_2

# Usage:
word1 = input("Enter the first word (e.g., 'x1 X3 X2 x3 x1 x1'): ")
word2 = input("Enter the second word: ")

# Check if the words are conjugate and print the result
result = are_words_conjugate(word1, word2)
```

```
82 print(f"The words '{word1}' and '{word2}' are {'conjugate' if result
       else 'not conjugate'}.")
```

Algorithm 2: Conjugacy problem in free groups of finite rank

# References


[1] S. I. Adian. Unsolvability of some algorithmic problems in the theory of groups. *TTr. Mosk. Mat. Obs.*, 6:231–298, 1957.

[2] M. Artin. *Algebra*. Prentice Hall, 1991.

[3] Goulnara Arzhantseva, Christopher Cashen, Dominik Gruber, and David Hume. Negative curvature in graphical small cancellation groups. *Groups, Geometry, and Dynamics*, May 2019.

[4] William W. Boone. The word problem. *Annals of Mathematics*, 70(2):207–265, 1959.

[5] N. Brady, T. Riley, and H. Short. *The Geometry of the Word Problem for Finitely Generated Groups*. Advanced Courses in Mathematics - CRM Barcelona. Birkhäuser Basel, 2007.

[6] John L. Britton. The word problem. *Annals of Mathematics*, 77(1):16–32, 1963.

[7] Professor Cayley. Desiderata and suggestions: No. 2. the theory of groups: Graphical representation. *American Journal of Mathematics*, 1(2):174–176, 1878.

[8] Francesco Nishanil Chotuck. Algorithms for the word and conjugacy problems in free groups of finite rank. URL: https://github.com/FrankieNC/Algorithms-to-solve-the-word-and-conjugacy-problem-in-free-groups-of-finite-rank, 2024.

[9] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345, 1936.

[10] Donald J. Collins. A simple presentation of a group with unsolvable word problem. *Illinois Journal of Mathematics*, 30:230–234, 1986.

[11] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2024 edition, 2024.

[12] M. Dehn. Über unendliche diskontinuierliche gruppen. *Mathematische Annalen*, 71(1):116–144, 1911.

[13] M. Dehn. Transformation der kurven auf zweiseitigen flächen. *Mathematische Annalen*, 72(3):413–421, 1912.

[14] C. Druţu and M. Kapovich. *Geometric Group Theory*. Colloquium Publications. American Mathematical Society, 2018.

[15] A. A. Fridman. On the relation between the word problem and the conjugacy problem in finitely defined groups. *Journal of Symbolic Logic*, 34(3):507–508, 1969.

[16] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

[17] Martin Greendlinger. On dehn's algorithms for the conjugacy and word problems, with applications. *Communications on Pure and Applied Mathematics*, 13(4):641–677, 1960.

[18] Pierre Grillet. *Semigroups: An Introduction to the Structure Theory*. Taylor & Francis Inc, 2017.

[19] M. Gromov. Random walk in random groups. *Geometric and Functional Analysis*,

13(1):73–146, 2003.

[20] A. Hatcher. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002.

[21] Graham Higman, B. H. Neumann, and Hanna Neuman. Embedding theorems for groups. *Journal of the London Mathematical Society*, s1-24(4):247–254, 1949.

[22] D. Hilbert and W. Ackermann. *Grundzüge der theoretischen Logik*. Grundlehren der mathematischen Wissenschaften in Einzeidarstellungen. J. Springer, 1928.

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

[24] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA, 2006.

[25] S. V. Ivanov and P. E. Schupp. On the hyperbolicity of small cancellation groups and one-relator groups. *Transactions of the American Mathematical Society*, 350(5):1851–1894, 1998.

[26] D. L. Johnson. *Presentations of Groups*. London Mathematical Society Student Texts. Cambridge University Press, 2 edition, 1997.

[27] D.L. Johnson. *Topics in the Theory of Group Presentations*. Cambridge books online. Cambridge University Press, 1980.

[28] Egbert R. Van Kampen. On some lemmas in the theory of groups. *American Journal of Mathematics*, 55(1):268–273, 1933.

[29] Dexter C. Kozen. *Automata and Computability*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1997.

[30] G. Lallement. *Semigroups and Combinatorial Applications*. A Wiley-Interscience publication. Wiley, 1979.

[31] S. Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, 2005.

[32] C. Löh. *Geometric Group Theory: An Introduction*. Universitext. Springer International Publishing, Cham, 2017.

[33] Markus Lohrey. Subgroup Membership in GL(2,Z). In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51:1–51:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[34] Salvador Lucas. The origins of the halting problem. *Journal of Logical and Algebraic Methods in Programming*, 121, 2021.

[35] Roger C. Lyndon. On dehn's algorithm. *Mathematische Annalen*, 166(3):208–228, 1966.

[36] Roger C. Lyndon and Paul E. Schupp. *Combinatorial group theory*. Classics in mathematics. Springer, Berlin, 2001.

[37] 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.

[38] A.I. Maltsev. *Algorithms and Recursive Functions*. Wolters-Noordhoff series of monographs and textbooks on pure and applied mathematics. Wolters-Noordhoff Publishing Company, 1970.

[39] Vassily Olegovich Manturov, Denis Fedoseev, Seongjeong Kim, and Igor Nikonov. *Invariants and Pictures*. World Scientific, 2020.

[40] John Meier. *Groups, Graphs and Trees: An Introduction to the Geometry of Infinite Groups*. London Mathematical Society Student Texts. Cambridge University Press, 2008.

[41] Charles F. Miller. *On Group-Theoretic Decision Problems and Their Classification. (AM-68), Volume 68*. Princeton University Press, Princeton, 1972.

[42] A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based Cryptography*. Advanced Courses in Mathematics - CRM Barcelona. Birkhäuser Basel, 2008.

[43] B. H. Neumann. Some Remarks on Infinite Groups. *Journal of the London Mathematical Society*, s1-12(2):120–127, 04 1937.

[44] P.S. Novikov. *On the Algorithmic Unsolvability of the Word Problem in Group Theory*. American Mathematical Society translations. American Mathematical Society, 1958.

[45] Carl-Fredrik Nyberg-Brodda. The adian-rabin theorem — an english translation. *arXiv preprint arXiv:2208.08560*, 2022.

[46] Øystein Ore and American Mathematical Society. *Theory of graphs*. American Mathematical Society Providence, Phode Island, Providence, Phode Island, 1962.

[47] Emil L. Post. Recursive unsolvability of a problem of thue. *Journal of Symbolic Logic*, 12(1):1–11, 1947.

[48] Michael O. Rabin. Recursive unsolvability of group theoretic problems. *Annals of Mathematics*, 67(1):172–194, 1958.

[49] D. Robinson. *A Course in the Theory of Groups*. Graduate Texts in Mathematics. Springer New York, 1996.

[50] Joseph Jonah Rotman. *An introduction to the theory of groups*. Springer-Verlag, New York; Berlin; Heidelberg [etc.], 1999.

[51] Paul E. Schupp. On dehn's algorithm and the conjugacy problem. *Mathematische Annalen*, 178(2):119–130, 1968.

[52] Vladimir Shpilrain. Complexity of some algorithmic problems in groups: a survey. *arXiv preprint arXiv:2401.09218*, 2024.

[53] Mária Šimková. When are two spaces homotopy equivalent? *arXiv preprint arXiv:2105.10443*, 2021.

[54] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, Boston, MA, third edition, 2013.

[55] J. Stillwell and M. Dehn. *Papers on Group Theory and Topology*. Springer New York, 2012.

[56] Heinrich Tietze. Über die topologischen invarianten mehrdimensionaler mannigfaltigkeiten. *Monatshefte für Mathematik und Physik*, 19(1):1–118, 1908.

[57] J. A. Todd and H. S. M. Coxeter. A practical method for enumerating cosets of a finite abstract group. *Proceedings of the Edinburgh Mathematical Society*, 5(1):26–34, 1936.

[58] A. M. Turing. Computability and $\lambda$-definability. *The Journal of Symbolic Logic*, 2(4):153–163, 1937.

[59] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[60] A. M. Turing. The word problem in semi-groups with cancellation. *Annals of Mathematics*, 52(2):491–505, 1950.

[61] J. H. C. Whitehead. Combinatorial homotopy. I. *Bulletin of the American Mathematical Society*, 55(3.P1):213–245, 1949.