

On the Undecidability of the Magnus, Word, Isomorphism, and Markov Property Problems for Finitely Presented Groups

Jessica Shi

Advisor: Professor Adam Levine

Junior Seminar, Fall 2016

Contents

1	Introduction	2
1.1	Context	2
1.2	Outline	3
2	Preliminaries and definitions	4
2.1	Computational complexity definitions	4
2.2	Combinatorial group theoretic definitions	5
3	\mathbb{Z}^2-machines	7
3.1	Construction of \mathbb{Z}^2 -machines	7
3.2	Construction of modified Turing machines	9
3.3	Halting problem for \mathbb{Z}^2 -machines	10
4	HNN extension and Britton's lemma	10
4.1	HNN extension	10
4.2	Britton's lemma	13
5	Magnus problem	13
6	Word problem	17
7	Isomorphism problem	18
8	Markov property	19
9	Acknowledgments	21

1. Introduction

1.1. Context

One of the fundamental problems of knot theory is determining whether any two given knots are equivalent. In particular, if a knot K_1 is equivalent (ambiently isotopic) to a knot K_2 , then $\mathbb{R}^3 \setminus K_1$ is homeomorphic to $\mathbb{R}^3 \setminus K_2$, and as such, the fundamental groups $\pi_1(\mathbb{R}^3 \setminus K_1)$ and $\pi_1(\mathbb{R}^3 \setminus K_2)$ are isomorphic. The fundamental group $\pi_1(\mathbb{R}^3 \setminus K_1)$ is known as the *knot group*, and is one of many invariants on knots.

It can be shown that the unknot has group \mathbb{Z} , whereas the trefoil has group $\langle x, y \mid x^2 = y^3 \rangle$, hence not all knots are equivalent to the unknot. Moreover, all knot groups can be finitely presented, by a construction known as the *Wirtinger presentation*. In this manner, many questions about knots are in fact related to more general questions about finitely generated groups, and Dehn [9] was the first to formulate the question of knot equivalence in terms of finitely presented groups (note that Tietze [30] was the first to formulate the isomorphism problem on finitely presented groups in general, without reference to knots). In particular, Dehn raised the *word problem*, which asks if there is an algorithm to determine whether $\omega = 1$ for any $\omega \in \mathcal{G}$, where \mathcal{G} is a finitely presented group. In terms of knot theory, this is equivalent to finding an algorithm to determine if a knot K is trivial; this is because K is trivial if and only if the knot group is abelian (since it would be isomorphic to \mathbb{Z}), and determining if a finitely presented group is abelian simply involves checking if the words $a_i a_j a_i^{-1} a_j^{-1}$ equal 1, for all generators a_1, \dots, a_n . Dehn [9, 10] solved the word problem on the fundamental groups of closed orientable surfaces and for the trefoil knot group, but not for knot groups in general.

Later, Artin [3] solved the word problem for braid groups, and Waldhausen [32] solved the word problem for all knot groups. Waldhausen's general solution in fact refers to a topological version of the word problem: given the fundamental group $\pi_1(S)$, the word problem on $\pi_1(S)$ is equivalent to determining if a given closed curve on S is contractible.

The focus of this paper, however, is the *undecidability* of the word problem in general, on all finitely presented groups (and related results). As such, we switch gears and give a short introduction on the concept of an “algorithm”. There are three intuitive notions of what is “computable,” all developed independently of each other. These are *general recursive functions*, introduced and refined by Gödel [11], *λ -computable functions*, introduced by Church [7], and *Turing machines*, introduced by Turing [31]. All three of these were proven to coincide, and the Church-Turing thesis [7] hypothesizes that all three coincide with the concept of functions that can be computed in the real-world, assuming unlimited computation resources. While there is debate about whether the Church-Turing thesis is a hypothesis or a definition, generally an “algorithm” is taken to be one that can be performed by a Turing machine.

In brief, a Turing machine consists of a finite automaton, which memorizes “states” that the machine is in, and an infinite tape with a read/write mechanism. The machine can write to and move along the tape according to the state it is in, and can change states according to what it reads from the tape. Given an input, a Turing machine may never stop computing, or *halt*, but if it does, it does so on either an *accept* state or a *reject* state. In this manner, a Turing machine can answer yes/no questions. The notion of *decidability*, then, is whether a yes/no question can be answered by a Turing machine. If so, the question is said to be *decidable*.

Based on this formal definition of computability, Church [8] hypothesized that the word problem

is undecidable (he also hypothesized that the knot problem is undecidable). Following this, a series of discoveries eventually proved this hypothesis correct.

First, Post [23] and Markov [16] independently proved that the word problem on semigroups is undecidable. Then, Boone [4] defined the *quasi-Magnus problem* and prove it undecidable; the quasi-Magnus problem asks, if given a finite presentation $\mathcal{G} = \langle A \mid R \rangle$, a subset $S \subseteq A$, and a word $\omega \in \mathcal{G}$, whether ω can be formulated as a word on S with positive exponents. The corresponding *Magnus problem*, which asks if ω is in the subgroup generated by S , is closely related to the word problem (this problem was first formulated by Magnus [14], who also solved it on one-relator groups).

A few years later, Boone [5] proved that the word problem is undecidable. Independently, Novikov [21] proved that the word problem is undecidable as well. However, their arguments were primarily combinatorial, and have been superseded by group theoretic results. Namely, Higman, Neumann, and Neumann [13] introduced the *HNN extension*, which Britton [6] used to replace the combinatorial arguments in Boone’s original proof. This is the argument we use to prove the undecidability of the word problem here, and specifically, we use a version of Aanderaa and Cohen’s [1] formulation that also proves the undecidability of the Magnus problem in the process.

Now, the isomorphism problem can be directly proven undecidable from the undecidability of the isomorphism problem [22], but it was first proven undecidable as a result of stronger results independently by Adian [2] and Rabin [24]. Specifically, Adian and Rabin proved that any question about finitely presented groups regarding a property that satisfies the *Markov property* is undecidable. The Markov property was first introduced by Markov [15, 16, 17] in reference to the same decidability problem on semigroups, and includes isomorphic, abelian, finite, trivial, torsion-free, free, and simple.

This concludes the material we will introduce in this paper. Before we delve into the paper, we note one important consequence of these undecidable problems, that is outside the scope of this paper. From a topological perspective, Markov [18] used the results of Adian [2] and Rabin [24] to prove the undecidability of the homeomorphism problem. Specifically, any finitely presented group can be represented as the fundamental group of a closed n -manifold for any $n \geq 4$, and if we let $M(\mathcal{G})$ represent the 4-manifold with fundamental group \mathcal{G} , Markov showed that \mathcal{G}_1 is isomorphic to \mathcal{G}_2 if and only if $M(\mathcal{G}_1)$ is homeomorphic to $M(\mathcal{G}_2)$. Thus, the isomorphism problem reduces to the homeomorphism problem, showing the homeomorphism problem undecidable.

1.2. Outline

In Section 2, we introduce some preliminaries and definitions related to computational complexity and combinatorial group theory. In Section 3, we discuss a modification to the Turing machine that will aid the proof of the undecidability of the Magnus problem. In Section 4, we introduce the HNN extension and Britton’s lemma, which we use in Section 5 to prove the undecidability of the Magnus problem. In Sections 6 and 7, we use the results of Section 5 to prove the undecidability of the word and isomorphism problems respectively. In Section 8, we extend the undecidability of the word problem to prove the undecidability of any problem on finitely presented groups about a property that satisfies the Markov property.

2. Preliminaries and definitions

2.1. Computational complexity definitions

Informally, a *Turing machine* (TM) consists of an infinite tape and a read/write head that moves along the tape according to a finite automaton. The machine initially takes an input string on the otherwise empty tape; it reads the tape by moving the head to the left or right, and it stores information by writing on the tape. The finite automaton includes two distinct states, an accept state and a reject state, and the machine *accepts* or *rejects* the input upon entering the accept state or the reject state respectively. Note that the machine may never enter an accept or reject state, in which it runs forever and does not halt. Formally, we define a Turing machine as follows.

Definition 2.1. A *Turing machine* is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, where Q, Σ, Γ are finite sets and

1. Q is a set of states,
2. Σ is the input alphabet,
3. Γ is the tape alphabet (note that $\Sigma \subseteq \Gamma$),
4. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function,
5. $q_0 \in Q$ is the start state,
6. $q_{\text{accept}} \in Q$ is the accept state, and
7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{accept}} \neq q_{\text{reject}}$.

Note that for $(q, \omega, x) \in \delta(Q \times \Gamma)$, q denotes the next state of the finite automaton, ω denotes writing ω on the tape under the current position of the head, and x denotes moving the head to the left or to the right. For $(q, \omega) \in Q \times \Gamma$, q denotes the current state of the finite automaton and ω denotes the character on the tape under the current position of the head. Moreover, note that the blank symbol, denoted \sqcup , is an element of the tape alphabet Γ , but not the input alphabet Σ .

For the purposes of this paper, we will often not concern ourselves with using the formal definition of the Turing machine, and will instead use the commonly accepted abstractions. Moreover, many of the steps we take in the proofs of this paper will be computable (albeit tedious), so we will often not explicitly detail how a Turing machine can carry out those steps. The formal definitions will primarily be relevant in Section 3, where we construct a modified version of a Turing machine.

Note that the current state, tape contents, and head location encapsulate a “step” of a Turing machine, and with each use of the transition function, these items may change. We denote a setting of these items to be a *configuration* of the Turing machine, and in particular, we can represent any configuration as the word uqv , for state $q \in Q$ and strings $u, v \in \Gamma^* = \Gamma^+$. Here, the tape contains the string uv (and contains only blank symbols \sqcup before the first character of u and following the last character of v ¹), and the head location is at the first symbol of v . We call this representation a *complete state* of the Turing machine.

As an aside, for any string $u \in \Gamma^*$, note that u may be the empty string, which we denote as ε . Moreover, we let u^R denote the reverse of u ; that is to say, for $u = u_1 \dots u_n$ where $u_i \in \Gamma$, we let $u^R = u_n \dots u_1$.

¹Here, we take a Turing machine to have a two-way infinite tape, which is equivalent to a Turing machine with a one-way infinite tape.

Now, for a given Turing machine M , the *language of M* , denoted $L(M)$, is the set of all strings that M accepts.

Definition 2.2. A language L is *decidable* if there exists a Turing machine M such that M halts on all inputs and $L(M) = L$. We say that M *decides* L ².

One of the most famous undecidable problems, first introduced by Turing [31], is the *halting problem*, which asks, “For a Turing machine M and input ω , does M halt on input ω ?” More formally, the following language is undecidable:

$$\text{HALT} = \{ \langle M, \omega \rangle \mid M \text{ is a TM and } M \text{ halts on input } \omega \},$$

where the angular brackets $\langle \cdot \rangle$ denote a suitable encoding of M, ω ³.

A common technique for proving that a problem is undecidable is *reducibility*, which involves converting an undecidable problem into the given problem, such that a decider for the given problem could be used to give a decider for the undecidable problem. The proofs in this paper will formulate everything in terms of proofs by contradiction, which should be clear even without an understanding of reducibility. However, for the sake of formality, we have the following definitions.

Definition 2.3. A function $f : \Sigma^* \rightarrow \Sigma^*$ is a *computable function* if there exists a Turing machine M such that for all inputs ω , M halts with $f(\omega)$ on its tape.

Definition 2.4. A language A is *mapping reducible*, or *reducible*, to a language B , denoted $A \leq_m B$, if there exists a computable function f such that for every ω , $\omega \in A$ if and only if $f(\omega) \in B$. f is the *reduction* from A to B .

Proposition 2.1. For languages A, B , if $A \leq_m B$ and A is undecidable, then B is undecidable.

In Sections 5, 6, and 7, we show that the halting problem reduces to the Magnus problem, which reduces to the word problem, which reduces to the isomorphism problem, thus showing that all three problems are undecidable. We forego a proof of the undecidability of the halting problem for the purposes of this paper, and instead reference Turing’s [31] diagonalization proof.

2.2. Combinatorial group theoretic definitions

We now introduces some group theoretic terms and notations. A *free group* is a group with generators that satisfy no relations other than those that are given by the group axioms. A free group can be described by an arbitrary set of elements, say $S = \{a_1, a_2, \dots\}$, which represent the generators of the group. For a free group on S , we define a *word* to be a finite string of generators or their inverses, i.e. $a_{i_1}^{\epsilon_1} a_{i_2}^{\epsilon_2} \dots a_{i_k}^{\epsilon_k}$ where $\epsilon_k \in \{-1, 1\}$.

²Note that there is a distinction between *decidable* and *recognizable*. A language L is *recognizable* if there exists a Turing machine M such that $L(M) = L$ (so M is not required to halt on all inputs). A *decidable* language is also known as a *recursive* language, and a *recognizable* language is also known as a *recursively enumerable* language. Many sources regarding the word problem claim to show that the word problem is *unsolvable*, but this language is not quite precise because sometimes “solvable” is equated with “recognizable,” whereas these sources actually prove that the word problem is *undecidable* and the word problem is in actuality recognizable [28, 29]. For the purposes of this paper, we use the term “decidable” to avoid confusion.

³Note that the halting problem is undecidable, but recognizable, hence the importance of the distinction between the two terms. The complement of the halting problem is an example of a problem that is both undecidable and unrecognizable [27].

For any word ω in a free group on S , we can *reduce* ω by canceling all subwords of the form $a_i a_i^{-1}$ or $a_i^{-1} a_i$. We can repeatedly reduce ω until there are no such subwords left, and the word we receive, say ω_0 , is called the *reduced form* of ω . Note that there is exactly one reduced form for any given ω .

A *relation* R among elements a_1, \dots, a_n of a group \mathcal{G} is a word r in the free group on $\{a_1, \dots, a_n\}$ that evaluates to 1 in \mathcal{G} . Given a free group \mathcal{F} on $S = \{a_1, a_2, \dots\}$ and a subset $R = \{r_1, r_2, \dots\}$ of \mathcal{F} , the group generated by S with relations r_1, r_2, \dots is the quotient group $\mathcal{G} = \mathcal{F}/\mathcal{R}$, where \mathcal{R} is the normal subgroup of \mathcal{F} generated by R . The sets S and R are called a *group presentation* of \mathcal{G} , denoted by $\langle a_1, a_2, \dots \mid r_1, r_2, \dots \rangle$. If there exist S, R such that S and R are finite sets, then \mathcal{G} is said to be *finitely presented*.

Also, for any group \mathcal{G} and subset $S \subseteq \mathcal{G}$, we let $\langle S \rangle_{\mathcal{G}}$ denote the subgroup of \mathcal{G} generated by S . Note that we may drop the subscript if the group \mathcal{G} is clear.

For any two presentations $\mathcal{G} = \langle S_G \mid R_G \rangle$ and $\mathcal{H} = \langle S_H \mid R_H \rangle$, we define the *free product* to be $\mathcal{G} \cup \mathcal{H} = \mathcal{G} * \mathcal{H} = \langle S_G \cup S_H \mid R_G \cup R_H \rangle$. Given the homomorphisms $\phi_{\mathcal{G}} : \mathcal{F} \rightarrow \mathcal{G}$ and $\phi_{\mathcal{H}} : \mathcal{F} \rightarrow \mathcal{H}$ from some group \mathcal{F} , we define the *amalgamated product* or *free product with amalgamation* to be $\mathcal{G} *_F \mathcal{H} = (\mathcal{G} * \mathcal{H})/\mathcal{N}$, where \mathcal{N} is the normal subgroup of $\mathcal{G} * \mathcal{H}$ generated by elements of the form $\phi_{\mathcal{G}}(f)\phi_{\mathcal{H}}(f)^{-1}$ for all $f \in \mathcal{F}$. In more colloquial terms, the amalgamated product $\mathcal{G} *_F \mathcal{H}$ is given by the free product $\mathcal{G} * \mathcal{H}$ with the added relations $\phi_{\mathcal{G}}(f)\phi_{\mathcal{H}}(f)^{-1} = 1$ for all $f \in \mathcal{F}$.

We can now define the Magnus, word, and isomorphism problems.

Magnus Problem. [14]. Given a finitely presented group \mathcal{G} , a subset $\{a_1, \dots, a_m\}$ of the generators of \mathcal{G} , and a word $\omega \in \mathcal{G}$, is ω in the subgroup generated by $\{a_1, \dots, a_m\}$? More formally, the Magnus problem is given by the following language:

$$\text{MAGNUS} = \{ \langle \langle A \mid R \rangle, A', \omega \rangle \mid A' \subseteq A, \omega \in \langle A \mid R \rangle, \text{ and } \omega \text{ is in the subgroup generated by } A' \}.$$

Word Problem. [9]. Given a fixed finite presentation of a group \mathcal{G} and a word $\omega \in \mathcal{G}$, does $\omega = 1$ in \mathcal{G} ? More formally, the word problem is given by the following language:

$$\text{WORD} = \{ \langle \langle A \mid R \rangle, \omega \rangle \mid \omega \in \langle A \mid R \rangle \text{ and } \omega = 1 \}.$$

Isomorphism Problem. [9]. Given two finite presentations $\langle a_1, \dots, a_n \mid r_1, \dots, r_m \rangle$ and $\langle b_1, \dots, b_p \mid s_1, \dots, s_q \rangle$, do they represent isomorphic groups? More formally, the isomorphism problem is given by the following language:

$$\text{ISO} = \{ \langle \langle A \mid R \rangle, \langle B \mid S \rangle \rangle \mid R, S \text{ are relations on } A, B \text{ respectively and } \langle A \mid R \rangle \cong \langle B \mid S \rangle \}.$$

We also define the Markov property, and formulate the decidability question regarding the Markov property.

Definition 2.5. [15, 16, 17]. Let P be any property of finitely presented groups that is preserved under isomorphism. P is a *Markov property* if

1. there is a finitely presented group with property P and
2. there is a finitely presented group that cannot be embedded into any finitely presented group with property P .

Markov Property Problem. Given a fixed finite presentation of a group \mathcal{G} and a property P that satisfies the Markov property, does \mathcal{G} have property P ? More formally, the problem is given by the following language:

$$\text{MARKOV}_P = \{ \langle A \mid R \rangle \mid \langle A \mid R \rangle \text{ has the property } P \text{ where } P \text{ is a Markov property} \}.$$

3. \mathbb{Z}^2 -machines

Here, we introduce a modification to the standard Turing machine and to the halting problem, which will facilitate the reduction in Section 5. Specifically, we define the \mathbb{Z}^2 -machine, which represents each complete state as a pair of integers that allows the transition function to be represented by arithmetic computations. \mathbb{Z}^2 -machines were first introduced by Minsky [19] as Γ -machines and later refined by Aanderaa and Cohen [1] as modular machines. Aanderaa and Cohen used the modular machine to provide a proof for the undecidability of the word and isomorphism problems, and the proof we give here in Sections 6 and 7 are based on their proof. This form of the \mathbb{Z}^2 -machine was constructed by Stillwell [29].

3.1. Construction of \mathbb{Z}^2 -machines

First, for a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, consider any complete state uqv , where $q \in Q$ and $u, v \in \Gamma^+$. Then, (uq, v^R) is a *complete state pair*. We can interpret this pair as an element of \mathbb{N}^2 by considering the symbols $Q \cup \Gamma$ as digits in base $b = |Q| + |\Gamma|$, where we associate $\sqcup \in \Gamma$ with the digit 0.

Note that since \sqcup is associated with 0, the infinite blank spaces to the left and right of u and v respectively on the tape are represented in the complete state pair. This results in precisely one complete state pair for any configuration of M (whereas based on our definition of “complete state” in Section 2.1, there may be more than one complete state for any configuration of M), since u and v^R uniquely represent the entire infinite tape.

Now, note that the transition function can be given by quintuples $(q_i, \omega_j, q_{i'}, \omega_{j'}, x)$ where $(q_i, \omega_j) \in Q \times \Gamma$ and $(q_{i'}, \omega_{j'}, x) \in Q \times \Gamma \times \{L, R\}$. This quintuple acts on complete states of the form $uq_i\omega_jv$. In particular,

$$\begin{aligned} (q_i, \omega_j, q_{i'}, \omega_{j'}, L) &: u\omega_kq_i\omega_jv \rightarrow uq_{i'}\omega_k\omega_{j'}v, \\ (q_i, \omega_j, q_{i'}, \omega_{j'}, R) &: uq_i\omega_j\omega_kv \rightarrow u\omega_{j'}q_{i'}\omega_kv, \end{aligned} \quad (1)$$

where $\omega_k \in \Gamma$ and $u \in \Gamma^+, v \in \Gamma^*$. In terms of complete state pairs, the quintuple $(q_i, \omega_j, q_{i'}, \omega_{j'}, x)$ acts on pairs of the form $(uq_i, v\omega_j)$. Thus, we have

$$\begin{aligned} (q_i, \omega_j, q_{i'}, \omega_{j'}, L) &: (u\omega_kq_i, v\omega_j) \rightarrow (uq_{i'}, v\omega_{j'}\omega_k), \\ (q_i, \omega_j, q_{i'}, \omega_{j'}, R) &: (uq_i, v\omega_k\omega_j) \rightarrow (u\omega_{j'}q_{i'}, v\omega_k). \end{aligned} \quad (2)$$

Note that v in Equation 2 is equal to v^R in Equation 1; we make this change of notation for simplicity. Since complete state pairs are elements of \mathbb{N}^2 , when considered in base b , we can write these transformations arithmetically, as

$$\begin{aligned} (q_i, \omega_j, q_{i'}, \omega_{j'}, L) &: (b^2 \cdot u + \omega_kq_i, b \cdot v + \omega_j) \rightarrow (b \cdot u + q_{i'}, b^2 \cdot v + \omega_{j'}\omega_k), \\ (q_i, \omega_j, q_{i'}, \omega_{j'}, R) &: (b \cdot u + q_i, b^2 \cdot v + \omega_k\omega_j) \rightarrow (b^2 \cdot u + \omega_{j'}q_{i'}, b \cdot v + \omega_k). \end{aligned} \quad (3)$$

Since the transition function can be fully expressed in terms of these quintuples, we can instead write a transition function for complete state pairs in terms of transformations of the forms

$$(b^2 \cdot U + A_\ell, b \cdot V + B_\ell) \mapsto (b \cdot U + C_\ell, b^2 \cdot V + D_\ell), \quad (4)$$

$$(b \cdot U + A_r, b^2 \cdot V + B_r) \mapsto (b^2 \cdot U + C_r, b \cdot V + D_r). \quad (5)$$

We call all transformations in the form of Equation 4 ℓ -transformations, and we call all transformations in the form of Equation 5 r -transformations. Note that A_ℓ, B_r, C_r, D_ℓ represent 2-digit base b numbers, and A_r, B_ℓ, C_ℓ, D_r represent 1-digit base b numbers. Also, when we refer to transformations of these forms, we refer to transformations that apply for all $U, V \in \mathbb{N}$, where $A_\ell, B_\ell, C_\ell, D_\ell, A_r, B_r, C_r, D_r$ are constants.

Note that the ℓ - and r -transformations form our \mathbb{Z}^2 -machine, say Z . Z takes as input (q_0, ω) where $\omega \in \Sigma^*$ (and if $\omega = \varepsilon$, we take the input of Z to be $(q_0, 0)$). Z accepts an input if, after successively applying ℓ - and r -transformations, there are no more transformations to be applied, and the last complete state pair in the sequence is of the form (uq_{accept}, v) . Z rejects an input if, after successively applying ℓ - and r -transformations, there are no more transformations to be applied, and the last complete state pair in the sequence is of the form (uq_{reject}, v) . Z halts on an input if after successively applying ℓ - and r -transformations, there are no more transformations to be applied. Note that this is a purposefully broader definition of “halting” than on Turing machines. The construction thus far focuses on modifying a Turing machine to be a \mathbb{Z}^2 -machine, but note that we can also consider a \mathbb{Z}^2 -machine as its own construct. We give a generalized definition as follows:

Definition 3.1. A \mathbb{Z}^2 -machine is a 7-tuple, $(Q, \Sigma, \Gamma, \delta', q_0, q_{\text{accept}}, q_{\text{reject}})$, where Q, Σ, Γ are finite sets and

1. Q is a set of states,
2. Σ is the input alphabet,
3. Γ is the tape alphabet (note that $\Sigma \subseteq \Gamma$),
4. $\delta' : \mathbb{N}^2 \rightarrow \mathbb{N}^2$ is a partial function, where the transformations in δ' are ℓ - or r -transformations,
5. $q_0 \in Q$ is the start state,
6. $q_{\text{accept}} \in Q$ is the accept state, and
7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{accept}} \neq q_{\text{reject}}$.

Note that as in the construction, Z begins with some input $\omega \in \Sigma^*$, and uses δ' successively on $(q_0, \omega) \in \mathbb{N}^2$ (when considered as base b numbers) until no more transformations can be applied. Z may run forever, or Z may halt, in which Z may accept the input, reject the input, or neither.

Now, given a Turing machine M , denote the \mathbb{Z}^2 -machine constructed by the above procedure (that is, transforming the transition function of M into the corresponding ℓ - or r -transformations) by $Z(M)$. We now claim that M and $Z(M)$ are equivalent machines on the domain of inputs Σ^* .

Lemma 3.1. *Given a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, M and $Z(M)$ are equivalent on the domain of inputs Σ^* . That is to say, M accepts if and only if $Z(M)$ accepts, M rejects if and only if $Z(M)$ rejects, and the contents on the tape of M upon halting are equivalent to the contents of the complete state pair of $Z(M)$ upon halting.*

Proof. This is clear because we have a bijection between configurations and complete state pairs, and a bijection between the transition function and the ℓ - and r -transformations. The former follows from the fact that a complete state pair completely encodes the tape of the Turing machine, and the latter follows from construction.

Moreover, for each configuration of M , exactly one transition can act upon that configuration, and by construction, for each corresponding complete state pair, exactly the corresponding ℓ - or

r -transformation can act upon that pair (this is clear because if not, we can work backwards from Equation 3 to Equation 1 to obtain two transitions that act upon the same configuration, which is a contradiction). Thus, we have a transition applies to a configuration in M if and only if the corresponding ℓ - or r -transformation applies to the corresponding complete state pair in $Z(M)$.

Thus, examining the configurations of M and the complete state pairs of $Z(M)$ resulting from any input $\omega \in \Sigma^*$, we see that M and $Z(M)$ produce the same output. \square

3.2. Construction of modified Turing machines

Now, Stillwell [29] uses the *universal Turing machine* to define a version of the halting problem on \mathbb{Z}^2 -machines. However, universal Turing machines are a more powerful tool than necessary to achieve the required reduction, so instead we introduce a modification to Turing machines in general. Stillwell [28] introduces a similar modification, but we utilize a version that is easier to formalize. This modification is not unusual, and indeed is commonly used when constructing a *tableau* representation of a Turing machine [27]. Simply put, we modify any given Turing machine so that upon reaching an accepting or rejecting state, it clears the contents of the tape and then enters an arbitrary halting state; we do this by ensuring that the Turing machine never writes a blank space \sqcup , and then uses \sqcup to determine the used sections of the tape. We give a formal definition as follows.

Definition 3.2. Given a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, we define a *modified Turing machine* $M(M)$ to be the Turing machine $(Q \cup \{q_\ell, q_h, q'_h\}, \Sigma, \Gamma \cup \{\#\}, \delta', q_0, q_h, q'_h)$ where $q_\ell, q_h, q'_h \notin Q$, $\# \notin \Gamma$, and δ' is defined as follows:

1. For every $(q, \omega) \in Q \times \Gamma$ such that $\delta(q, \omega) = (q', \sqcup, x)$ for some $q' \in Q, x \in \{\text{L}, \text{R}\}$, let $\delta'(q, \omega) = (q', \#, x)$.
2. For every $(q, \omega) \in Q \times \Gamma$ such that $\delta(q, \omega) \neq (q', \sqcup, x)$ for all $q' \in Q, x \in \{\text{L}, \text{R}\}$, let $\delta'(q, \omega) = \delta(q, \omega)$.
3. For every $q \in Q$, let $\delta'(q, \#) = \delta'(q, \sqcup)$.
4. For every $\omega \in (\Gamma \cup \{\#\}) \setminus \{\sqcup\}$, let $\delta'(q_{\text{accept}}, \omega) = \delta'(q_{\text{reject}}, \omega) = (q_{\text{accept}}, \omega, \text{R})$.
5. Let $\delta'(q_{\text{accept}}, \sqcup) = \delta'(q_{\text{reject}}, \sqcup) = (q_\ell, \sqcup, \text{L})$.
6. For every $\omega \in (\Gamma \cup \{\#\}) \setminus \{\sqcup\}$, let $\delta'(q_\ell, \omega) = (q_\ell, \sqcup, \text{L})$.
7. Let $\delta'(q_\ell, \sqcup) = (q_h, \sqcup, \text{L})$.

This fully defines δ' , and note that by definition, if M halts on input $\omega \in \Sigma^*$, then $M(M)$ will halt on input ω with an empty tape and in state q_h (that is to say, $M(M)$ accepts input ω with an empty tape). We forego a formal proof of this, since it would be merely tedious and the result is clear from construction.

Lemma 3.2. Given a Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ and input $\omega \in \Sigma^*$, M halts on input ω if and only if $M(M)$ accepts input ω . Moreover, if $M(M)$ accepts input ω , then $M(M)$ accepts ω on an empty tape.

3.3. Halting problem for \mathbb{Z}^2 -machines

We can now formulate the halting problem for \mathbb{Z}^2 -machines, as follows.

Halting Problem. Given a \mathbb{Z}^2 -machine Z and an input ω , does Z halt on input ω in the pair $(0,0)$? More formally, the halting problem for \mathbb{Z}^2 -machines is given by the following language:

$$\text{HALT}_{\mathbb{Z}^2} = \{\langle Z, \omega \rangle \mid Z \text{ is a } \mathbb{Z}^2\text{-machine and } Z \text{ halts on input } \omega \text{ in pair } (0,0)\}.$$

We show that $\text{HALT}_{\mathbb{Z}^2}$ is undecidable by a reduction from the standard halting problem, HALT .

Lemma 3.3. *$\text{HALT}_{\mathbb{Z}^2}$ is undecidable.*

Proof. Assume $\text{HALT}_{\mathbb{Z}^2}$ is decidable by a TM H . We construct a TM H' to decide HALT , with input $\langle M, \omega \rangle$.

First, we have H' construct the modified Turing machine corresponding to M , $M(M)$, based on the steps in Section 3.2. Then, we have H' construct the \mathbb{Z}^2 -machine corresponding to $M(M)$, say $Z(M(M)) = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, based on the steps in Section 3.1. We have H' add to $Z(M(M))$ the ℓ -transformations

$$(b^2 \cdot U + \omega_k q_{\text{accept}}, b \cdot V) \rightarrow (b \cdot U, b^2 \cdot V + \omega_k)$$

for all $\omega_k \in \Gamma$, giving a new \mathbb{Z}^2 -machine, say Z .

Now, we have H' run H on $\langle Z, \omega \rangle$, and if H accepts, then H' accepts. Otherwise, H' rejects.

We now claim that H' accepts if and only if M halts on input ω . Clearly, if H' accepts, then this means that Z halts on ω in pair $(0,0)$, which means that $Z(M(M))$ accepts ω in pair $(q_{\text{accept}}, 0)$, which means that $M(M)$ accepts ω (by Lemma 3.1), which means that M halts on ω (by Lemma 3.2), as desired. If M halts on ω , then $M(M)$ accepts ω with an empty tape (by Lemma 3.2), which means that $Z(M(M))$ accepts ω in state $(q_{\text{accept}}, 0)$ (by Lemma 3.1), which means that Z halts on ω in pair $(0,0)$. Thus, H accepts $\langle Z, \omega \rangle$, so H' accepts, as desired.

However, HALT is undecidable, so we have reached a contradiction. Thus, $\text{HALT}_{\mathbb{Z}^2}$ is undecidable. \square

4. HNN extension and Britton's lemma

When Boone [5] and Novikov [21] first proved the undecidability of the word problem, they used combinatorial arguments, that are somewhat complicated and that we will not address in this paper. Instead, those arguments have been superseded by group theoretic results, mainly the HNN construction, which was first introduced by Higman, Neumann, and Neumann [13] (prior to the solution to the word problem). Britton [6] used these arguments to give a different proof of the word problem, introducing Britton's lemma in the process. We discuss the HNN construction in Section 4.1, and Britton's lemma in Section 4.2.

4.1. HNN extension

First, we define the *HNN extension*.

Definition 4.1. Let \mathcal{G} be a group, and let pairs of elements b_i, c_i define an isomorphism between the subgroups \mathcal{B} and \mathcal{C} generated by b_i and c_i respectively, such that $b_i \mapsto c_i$. Then, the group $\mathcal{G}_t^* = \mathcal{G} \cup \langle t \mid \{t^{-1}b_it = c_i\} \rangle$ is the *HNN extension* of \mathcal{G} with *stable letter* t .

Essentially, we have defined \mathcal{G}_t^* such that the isomorphism $b_i \mapsto c_i$ is induced by conjugation by the stable letter t . Moreover, Higman, Neumann, and Neumann [13] proved that \mathcal{G} embeds into \mathcal{G}_t^* . Britton [6] proved the aptly named Britton’s lemma using this result, and this in combination with Higman, Neumann, and Neumann’s work is equivalent to a unique *normal form* of each of the elements in \mathcal{G}_t^* . This normal form was first introduced by Schupp [26], and incidentally, it is simpler to prove the uniqueness of this normal form directly, and then derive Higman, Neumann, and Neumann’s and Britton’s results from that. As such, we discuss the normal form and give a proof of uniqueness, and later derive the necessary results.

Now, any given word in \mathcal{G}_t^* is of the form $g_0 t^{\varepsilon_1} g_1 t^{\varepsilon_2} g_2 \dots t^{\varepsilon_k} g_k$, where $g_i \in G$ and $\varepsilon_i = \pm 1$ (note that g_i may be 1). We would like to transform these words into a form in which for all $i \neq 0$, g_i is given by a coset representative of either \mathcal{B} or \mathcal{C} . For the sake of consistency, we take “coset” to mean “right coset.”

First, let ϕ denote the isomorphism between \mathcal{B} and \mathcal{C} , given by $b_i \mapsto c_i$. We fix a set of coset representatives of \mathcal{B} and of \mathcal{C} in \mathcal{G} , denoted by $\{g_{B,i}\}$ and $\{g_{C,i}\}$ respectively⁴. Let 1 be the representative of the coset \mathcal{B} and the representative of the coset \mathcal{C} .

For $i = k$ to 1, we perform the following operation:

If $\varepsilon_i = -1$, then consider the factorization of g_i into $b_i g_{B,i}$ for $b_i \in \mathcal{B}$ and some coset representative $g_{B,i}$. Note that

$$t^{\varepsilon_i} g_i = t^{-1} b_i g_{B,i} = t^{-1} t \phi(b_i) t^{-1} g_{B,i} = \phi(b_i) t^{-1} g_{B,i}.$$

So, replace $t^{\varepsilon_i} g_i$ with $\phi(b_i) t^{-1} g_{B,i}$, and combine $\phi(b_i)$ with g_{i-1} to receive a new g_{i-1} .

Similarly, if $\varepsilon_i = 1$, then consider the factorization of g_i into $c_i g_{C,i}$ for $c_i \in \mathcal{C}$ and some coset representative $g_{C,i}$. Note that

$$t^{\varepsilon_i} g_i = t c_i g_{C,i} = t t^{-1} \phi^{-1}(c_i) t g_{C,i} = \phi^{-1}(c_i) t g_{C,i}.$$

So, replace $t^{\varepsilon_i} g_i$ with $\phi^{-1}(c_i) t g_{C,i}$, and combine $\phi^{-1}(c_i)$ with g_{i-1} to receive a new g_{i-1} .

After this loop, we remove any words of the form $t \cdot 1 \cdot t^{-1}$ or of the form $t^{-1} \cdot 1 \cdot t$.

This gives us $g'_0 t^{\delta_1} g'_1 t^{\delta_2} g'_2 \dots t^{\delta_m} g'_m$, where $\delta_i = \pm 1$, $g_0 \in \mathcal{G}$, and g_i for $i \neq 0$ is a coset representative \mathcal{B} or \mathcal{C} when $\delta_i = -1$ or $\delta_i = 1$ respectively.

Now, note that each g'_i can be replaced by any equivalent word in \mathcal{G} ; in order for this normal form to be unique, we replace g'_i by its equivalence class, denoted $[g'_i]$, in \mathcal{G} . Thus, we have

$$[g'_0] t^{\delta_1} [g'_1] t^{\delta_2} [g'_2] \dots t^{\delta_m} [g'_m],$$

which we call the *normal form* of the element $g_0 t^{\varepsilon_1} g_1 t^{\varepsilon_2} g_2 \dots t^{\varepsilon_k} g_k \in \mathcal{G}_t^*$. Note that when we discuss normal forms, we omit the brackets $[\cdot]$ and assume that it is clear that when we reference an element $g'_i \in \mathcal{G}$, we mean the equivalence class of that element.

We now establish the uniqueness of this normal form.

Lemma 4.1. *Given a group \mathcal{G} , the normal form of an element of its HNN extension \mathcal{G}_t^* is unique.*

⁴Note that the normalization process is not necessarily computable by a Turing machine, and we will not need to normalize in any construction involving Turing machines.

Proof. We first construct a homomorphism $\rho : \mathcal{G}_t^* \rightarrow S(N)$, where N is the set of all normal forms in \mathcal{G}_t^* and where $S(N)$ is the set of all permutations $\pi : N \rightarrow N$. We now define ρ as follows.

For $g \in \mathcal{G}$, let $\rho(g)$ be given by

$$\rho(g)(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k) = g g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k,$$

and let $\rho(t)$ be given by

$$\rho(t)(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k) = \begin{cases} \phi^{-1}(g'_0) g'_1 t^{\delta_2} g'_2 \dots t^{\delta_k} g'_k, & \text{if } \delta_1 = -1 \text{ and } g'_0 \in \mathcal{C} \\ \phi^{-1}(c) t g''_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k, & \text{otherwise, where } g'_0 = c g''_0 \text{ for coset} \\ & \text{representative } g''_0 \in \mathcal{G} \text{ and } c \in \mathcal{C}. \end{cases}$$

First, note that $\rho(g)$ is clearly a permutation of N (since it merely involves left multiplying g) and gives a homomorphism, since for $g, g' \in \mathcal{G}$, we have $\rho(gg') = \rho(g)\rho(g')$ and $\rho(1) = 1$. Also, we claim that $\rho(t)$ does indeed give a permutation of N , since we can define an inverse

$$\rho(t^{-1})(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k) = \begin{cases} \phi(g'_0) g'_1 t^{\delta_2} g'_2 \dots t^{\delta_k} g'_k, & \text{if } \delta_1 = 1 \text{ and } g'_0 \in \mathcal{B} \\ \phi(b) t^{-1} g''_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k, & \text{otherwise, where } g'_0 = b g''_0 \text{ for coset} \\ & \text{representative } g''_0 \in \mathcal{G} \text{ and } b \in \mathcal{B}. \end{cases}$$

We now verify that $\rho(t)\rho(t^{-1}) = 1$. Consider any given normal form $g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k$. We split this verification into cases.

For the first case, let $\delta_1 = -1$ and $g'_0 \in \mathcal{C}$. Note that the first case of $\rho(t)$ applies, so we have

$$\rho(t)(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k) = \phi^{-1}(g'_0) g'_1 t^{\delta_2} g'_2 \dots t^{\delta_k} g'_k.$$

Now, note that since $g'_1 \notin \mathcal{B}$, we have $\phi^{-1}(g'_0) g'_1 \notin \mathcal{B}$, so the second case of $\rho(t^{-1})$ applies. Moreover, since $\phi^{-1}(g'_0) \in \mathcal{B}$, we have the coset representative of $\phi^{-1}(g'_0) g'_1$ is g'_1 . Thus,

$$\begin{aligned} \rho(t^{-1})(\phi^{-1}(g'_0) g'_1 t^{\delta_2} g'_2 \dots t^{\delta_k} g'_k) &= \phi(\phi^{-1}(g'_0)) t^{-1} g'_1 \dots t^{\delta_k} g'_k \\ &= g'_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k. \end{aligned}$$

For all other cases, the second case of $\rho(t)$ applies, so we have

$$\rho(t)(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k) = \phi^{-1}(c) t g''_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k,$$

where $g'_0 = c g''_0$ for coset representative $g''_0 \in \mathcal{G}$ and $c \in \mathcal{C}$. Trivially, $\phi^{-1}(c) \in \mathcal{B}$, so we have

$$\begin{aligned} \rho(t^{-1})(\phi^{-1}(c) t g''_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k) &= \phi(\phi^{-1}(c)) g''_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k \\ &= g'_0 t^{\delta_1} g'_1 \dots t^{\delta_k} g'_k, \end{aligned}$$

as desired.

Similarly, it is easy to check that $\rho(t^{-1})\rho(t) = 1$, so $\rho(t)$ does indeed give a permutation on N . To complete the proof that ρ defines a homomorphism, we need only show that for $c \in \mathcal{C}$, $\rho(c) = \rho(t^{-1})\rho(\phi^{-1}(c))\rho(t)$. This again can be checked with a great deal of casework, which we omit for the purposes of this paper.

We now claim that it is sufficient that for $g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k$ in normal form, $\rho(g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k)(1) = g'_0 t^{\delta_1} \dots t^{\delta_k} g'_k$. Assume for purposes of contradiction that two distinct normal forms $n_1, n_2 \in N$ correspond to the same $h \in \mathcal{G}_t^*$. Then, $\rho(h) = \rho(n_1) = \rho(n_2)$. However, $\rho(n_1)(1) = n_1$ and $\rho(n_2)(1) = n_2$ (where $n_1 \neq n_2$ when considered as elements of N), so clearly $\rho(n_1) \neq \rho(n_2)$, which is a contradiction. Thus, n_1 and n_2 must represent distinct elements in \mathcal{G}_t^* , as desired. \square

Corollary 4.2. *Higman, Neumann, Neumann [13].* $\mathcal{G} \hookrightarrow \mathcal{G}_t^*$.

Proof. This follows immediately, since every $[g] \in \mathcal{G}$ is identical with its normal form. \square

4.2. Britton's lemma

Britton's lemma follows directly from Lemma 4.1.

Corollary 4.3. *Britton's lemma [6].* Let $\omega = g_0 t^{\varepsilon_1} g_1 t^{\varepsilon_2} g_2 \dots t^{\varepsilon_k} g_k \in \mathcal{G}_t^*$, where $g_i \in \mathcal{G}$ and $\varepsilon_i = \pm 1$. If $\omega = 1$ in \mathcal{G}_t^* , then either

1. $k = 0$ and $g_0 = 1$, or
2. $k > 0$ and ω contains either a subword $t^{-1}bt$ where $b \in \mathcal{B}$ or a subword tct^{-1} where $c \in \mathcal{C}$.

Proof. Note that if $k = 0$, then trivially $g_0 = 1$. Now, we consider the case where $k > 0$. Note that by Lemma 4.1, the normal form of ω is unique, namely 1. As such, in normalizing ω , we must at some point cancel tt^{-1} or $t^{-1}t$. Thus, we must have either a subword tg_it^{-1} or $t^{-1}g_it$.

Now, if we have a subword tg_it^{-1} , when normalizing g_{i+1} , we insert some $c \in \mathcal{C}$ to receive tg_ict^{-1} . Necessarily, then, we must have $g_ic \in \mathcal{C}$, in order to move g_ic to the left when normalizing g'_i (to receive tt^{-1}). Thus, $g_i \in \mathcal{C}$, as desired.

Similarly, if we have a subword $t^{-1}g_it$, when normalizing g_{i+1} , we insert some $b \in \mathcal{B}$ to receive $t^{-1}gibt$. Necessarily, we must have $g_ib \in \mathcal{B}$, in order to move g_ib to the left when normalizing g'_i (to receive $t^{-1}t$). Thus, $g_i \in \mathcal{B}$, as desired. \square

5. Magnus problem

We now have all of the tools necessary to prove that the Magnus problem is undecidable. We use a methodology that is similar to one that Aanderaa and Cohen [1] uses to prove the undecidability of the word and isomorphism problems. However, their version is slightly more complex, and this version is simpler in terms of computation. The version here is reproduced from Stillwell's [28] treatment of the Magnus, word, and isomorphism problems.

We focus on a specific case of the Magnus problem, namely on free groups of rank 3, say $\mathcal{F} = \langle x, y, z \mid - \rangle$.

First, we define $p(i, j) = x^i z y^j$ for $(i, j) \in \mathbb{N}^2$. The idea here is to apply p to complete state pairs, sending complete state pairs to elements of \mathcal{F} . As such, we would like to define ℓ - and r -transformations as isomorphism ψ_ℓ and ψ_r respectively. Recall that an ℓ -transformation is given by $(b^2U + A_\ell, bV + B_\ell) \mapsto (bU + C_\ell, b^2V + D_\ell)$. Applying $p(\cdot)$, we have

$$x^{b^2U + A_\ell} z y^{bV + B_\ell} \mapsto x^{bU + C_\ell} z y^{b^2V + D_\ell}.$$

Thus, we see that we can define ψ_ℓ as

$$\psi_\ell : x^{b^2} \mapsto x^b, x^{A_\ell} z y^{B_\ell} \mapsto x^{C_\ell} z y^{D_\ell}, y^b \mapsto y^{b^2}.$$

Similarly, recall that an r -transformation is given by $(bU + A_r, b^2V + B_r) \mapsto (b^2U + C_r, bV + D_r)$. Applying $p(\cdot)$, we have

$$x^{bU+A_r}zy^{b^2V+B_r} \mapsto x^{b^2U+C_r}zy^{bV+D_r}.$$

Thus, we can define ψ_r as

$$\psi_r : x^b \mapsto x^{b^2}, x^{A_r}zy^{B_r} \mapsto x^{C_r}zy^{D_r}, y^{b^2} \mapsto y^b.$$

We now show that ψ_ℓ and ψ_r are isomorphisms. We do so using the following lemma:

Lemma 5.1. *Let $\mathcal{F} = \langle x, y, z \mid - \rangle$ and let $u_1 = x^m$, $u_2 = x^i zy^j$, and $u_3 = y^n$, for $m, n \neq 0$ and $i, j \in \mathbb{Z}$. Then, $\{u_1, u_2, u_3\}$ is a basis for the subgroup of \mathcal{F} that it generates. Moreover, if $x^{i_1} zy^{j_1} \in \langle u_1, u_2, u_3 \mid - \rangle$, then $i_1 = i + m_1 m$ and $j_1 = j + n_1 n$ for $m_1, n_1 \in \mathbb{Z}$.*

Proof. We first show that there are no nontrivial relations between u_1 , u_2 , and u_3 , hence showing that $\{u_1, u_2, u_3\}$ forms the basis of the free subgroup that it generates. Consider any nontrivial word ω on the subgroup $\langle u_1, u_2, u_3 \rangle_{\mathcal{F}}$ such that $\omega = 1$. Trivially, if ω is solely in terms of u_1 and u_3 , then since ω is nontrivial, $\omega \neq 1$. Thus, ω must contain some instance of u_2 , and in fact, ω must contain multiple instances of u_2 . Consider any two consecutive instances of u_2 (separated only by instances of u_1 and u_3).

If the exponents of the consecutive u_2 instance have the same sign, then trivially we have $\omega \neq 1$, since we have a z^n term that cannot be canceled. If they have different signs, then we have one of two cases: either we will have the subword $zy^n \alpha y^{-n} z^{-1}$ or the subword $z^{-1} x^{-m} \alpha x^m z$ for some reduced word α on u_1 and u_3 where $\alpha \neq 1$ (otherwise, we have $u_2 u_2^{-1}$ or $u_2^{-1} u_2$, which is trivial). In the case of $zy^n \alpha y^{-n} z^{-1}$, in order for the $y^n \alpha y^{-n}$ subword to disappear, we must have the exponent sum of y be 0, in which case the exponent sum of y in α must be 0. Since α is reduced, α must be a word on u_1 (and not u_3). In that case, though, the exponent sum of x cannot be 0, so $y^n \alpha y^{-n}$ cannot disappear. A similar argument applies to $z^{-1} x^{-m} \alpha x^m z$. Thus, no such ω exists.

Thus, we have no nontrivial relations between u_1 , u_2 , and u_3 , so the subgroup generated by $\{u_1, u_2, u_3\}$ is exactly $\langle u_1, u_2, u_3 \mid - \rangle$.

Moreover, for any $\omega = x^{i_1} zy^{j_1} \in \langle u_1, u_2, u_3 \mid - \rangle$ where $x^{i_1} zy^{j_1}$ is in reduced form, note that ω contains exactly one occurrence of u_2 , since at least one occurrence is needed to produce the z component, and more than one nontrivial occurrence will not cancel any z components by the previous argument. Trivially, we also note that no u_3 component can occur to the left of the u_2 occurrence, and no u_1 occurrence can occur to the right of the u_2 occurrence. Thus, $\omega = u_1^{m_1} u_2 u_3^{n_1}$ for $m_1, n_1 \in \mathbb{Z}$, so we have $x^{i_1} zy^{j_1} = x^{i+m_1 m} zy^{j+n_1 n}$. Since both sides are in reduced form, we must have $i_1 = i + m_1 m$ and $j_1 = j + n_1 n$, as desired. \square

Thus, by Lemma 5.1, we have the subgroups generated by $\{x^{b^2}, p(A_\ell, B_\ell), y^b\}$ and $\{x^b, p(C_\ell, D_\ell), y^{b^2}\}$ are free with rank 3, so since ψ_ℓ defines a mapping of the bases of these subgroups, it is an isomorphism. Similarly, ψ_r is an isomorphism, as desired.

Now, consider any \mathbb{Z}^2 -machine Z . Let $\{\ell_1, \dots, \ell_n\}$ and $\{r_1, \dots, r_m\}$ denote all of the ℓ - and r -transformations respectively of Z . Let ψ_{ℓ_i} and ψ_{r_i} denote the corresponding isomorphisms for ℓ_i and r_i respectively, and let $T = \{t_{\ell_1}, \dots, t_{\ell_n}, t_{r_1}, \dots, t_{r_m}\}$. We perform a series of HNN extensions with stable letters T on \mathcal{F} , where each ψ_{ℓ_i} is induced by conjugation with stable letter t_{ℓ_i} and each ψ_{r_i} is induced by conjugation with stable letter t_{r_i} . We denote the result of these HNN extensions $\mathcal{F}_T^*(Z)$.

Note that clearly, given a transformation from one complete pair to the next in Z , we have the corresponding ψ isomorphism that translates $p(\cdot)$ of that complete pair to the next in \mathcal{F} , which gives us a corresponding conjugation in $\mathcal{F}_T^*(Z)$. We must, however, show essentially the converse, namely that every conjugation of elements in $\mathcal{F}_T^*(Z)$ that could be translated into complete state pairs and interpreted as a transformation in Z does indeed represent a valid transformation. To do this, we prove a series of lemmas.

Lemma 5.2. *If (X, Y) is a complete state pair, then at most one of ψ_{ℓ_i} or ψ_{r_i} applies to $p(X, Y)$.*

Proof. First, if ψ_{ℓ_i} applies to $p(X, Y)$, then necessarily $p(X, Y) \in \langle x^{b^2}, p(A_{\ell_i}, B_{\ell_i}), y^b \rangle$. It is easy to check that necessarily, $X = A_{\ell_i} + m_{\ell_i}b^2$ and $Y = B_{\ell_i} + n_{\ell_i}b$ for $m_{\ell_i}, n_{\ell_i} \in \mathbb{Z}$ (alternatively, we have checked this formally in Lemma 5.1).

Recall from Section 3.1 that A_{ℓ_i} is a 2-digit number in base b , and B_{ℓ_i} is a 1-digit number in base b . Thus, considering X and Y in base b , these two equations fully determine the values of A_{ℓ_i} and B_{ℓ_i} . Moreover, the values of A_{ℓ_i} and B_{ℓ_i} fully determine the corresponding ℓ -transformation, since at most one ℓ -transformation can apply to any given complete state pair in Z . As such, they fully determine ψ_{ℓ_i} , so at most one ψ_{ℓ_i} can apply to $p(X, Y)$.

Similarly, we have at most one ψ_{r_i} can apply to $p(X, Y)$.

Now, it is clear that ψ_{ℓ_i} and ψ_{r_j} cannot both apply to $p(X, Y)$ for some i, j , since if so, then $X = A_{\ell_i} + m_{\ell_i}b^2 = A_{r_j} + m_{r_j}b$ and $Y = B_{\ell_i} + n_{\ell_i}b = B_{r_j} + n_{r_j}b^2$ for $m_{\ell_i}, m_{r_j}, n_{\ell_i}, n_{r_j} \in \mathbb{Z}$. Then, the last digit of A_{ℓ_i} would equal the last digit of A_{r_j} , and the last digit of B_{ℓ_i} would equal the last digit of B_{r_j} . By inspection, this would cause the ℓ - and r -transformations to have overlapping domain for some complete state pair, so the partial function of Z , $\delta' : \mathbb{N}^2 \rightarrow \mathbb{N}^2$, is not well-defined, which is a contradiction.

As such, at most one of ψ_{ℓ_i} or ψ_{r_i} applies to $p(X, Y)$. □

Now, Lemma 5.2 shows that “fake” computations aren’t permissible in \mathcal{F} , and as a result, most “fake” computations aren’t permissible in $\mathcal{F}_T^*(Z)$. However, when translating from \mathcal{F} to $\mathcal{F}_T^*(Z)$, the isomorphisms become represented by conjugations by T ; this forms an equality between consecutive computation steps (rather than defining consecutive computation steps by a function), which adds a bidirectionality. Specifically, using some ψ isomorphism, $p(X, Y)$ maps to $p(X', Y')$ but not vice versa (unless permissible by another isomorphism), whereas using conjugation by some t , $p(X, Y)$ equals some conjugate of $p(X', Y')$ and vice versa.

In order to prohibit this bidirectionality, we use an argument given by Post [23] in proving the word problem on semigroups. We define the *halting subgroup*, denoted $\mathcal{F}_0(Z)$, to be

$$\mathcal{F}_0(Z) = \{\{p(U, V) \mid (U, V) \text{ is a complete state pair which } Z \text{ converts to } (0, 0)\}\}^5$$

Lemma 5.3. *$\mathcal{F}_0(Z)$ is closed under all $\psi_{\ell_i}^{\pm 1}$ and $\psi_{r_i}^{\pm 1}$.*

Proof. Consider any $p(U, V) \in \mathcal{F}_0(Z)$. By Lemma 5.2, at most one of ψ_{ℓ_i} or ψ_{r_i} applies to $p(U, V)$. Without loss of generality, let some ψ apply to $p(U, V)$, so $\psi(p(U, V)) = p(U', V')$. Then, (U', V') must be the complete state pair that results from applying the corresponding ℓ - or r -transformation to (U, V) , so since Z converts (U, V) to $(0, 0)$, Z must also convert (U', V') to $(0, 0)$, as desired. Thus, $p(U', V') \in \mathcal{F}_0(Z)$.

⁵Note that $\mathcal{F}_0(Z) \subseteq \mathcal{F} \subseteq \mathcal{F}_T^*(Z)$.

Now, we must consider $\psi_{\ell_i}^{-1}$ and $\psi_{r_i}^{-1}$. Note that if for some ψ , $\psi^{-1}(p(U, V)) = p(U', V')$, then $\psi(p(U', V')) = p(U, V)$, so by the same argument as above, (U, V) must be the complete state pair that results from applying the corresponding ℓ - or r -transformation to (U', V') . Since Z converts (U, V) to $(0, 0)$, Z must also convert (U', V') to $(0, 0)$, as desired. Thus, $p(U', V') \in \mathcal{F}_0(Z)$. \square

We now complete the proof that no interaction of T with $\mathcal{F}_0(Z)$ will produce $p(X, Y) \notin \mathcal{F}_0(Z)$ for complete state pair (X, Y) ⁶.

Lemma 5.4. *Let $\langle \mathcal{F}_0(Z), T \rangle$ denote the subgroup of $\mathcal{F}_T^*(Z)$ generated by $\mathcal{F}_0(Z) \cup T$. Then, $\mathcal{F} \cap \langle \mathcal{F}_0(Z), T \rangle = \mathcal{F}_0(Z)$.*

Proof. Consider a word $\omega \in \langle \mathcal{F}_0(Z), T \rangle$ such that $\omega = f$, where f is expressed in terms of x, y , and z (so $f \in \mathcal{F}$). Then, $\omega f^{-1} = 1$, and note that $\omega f^{-1} \in \mathcal{F}_T^*(Z)$. Then, by Britton's lemma (Corollary 4.3), since ωf^{-1} trivially consists of more than one character, we must have ωf^{-1} contains either a subword $t_i^{-1} b_i t_i$ or $t_i c_i t_i^{-1}$ for some $b_i, c_i \in \mathcal{F}_T^*(Z)$, where $i \in T$. Since f is in terms of x, y , and z , the subword is necessarily contained in ω , so $b_i, c_i \in \mathcal{F}_0(Z)$.

Now, by Lemma 5.3, we have $t_i^{-1} b_i t_i = \psi_i(b_i) \in \mathcal{F}_0(Z)$ or $t_i c_i t_i^{-1} = \psi_i^{-1}(c_i) \in \mathcal{F}_0(Z)$. Thus, we can replace the subword by an element of $\mathcal{F}_0(Z)$, effectively removing the t elements. Repeating this process until all t elements are removed, we have $\omega \in \mathcal{F}_0(Z)$, as desired. \square

We now prove the undecidability of the Magnus problem, and in fact, a specific case of the Magnus problem that we use in Section 6.

Theorem 5.5. *The Magnus problem is undecidable.*

Proof. Assume MAGNUS is decidable by a TM M . We construct a TM M' to decide $\text{HALT}_{\mathbb{Z}^2}$, with input $\langle Z, \omega \rangle$, where $Z = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$.

First, we have M' construct the presentation for $\mathcal{F}_T^*(Z)$, as given earlier in this section and as based on the HNN extension in Section 4.1; denote this presentation by $\langle \mathcal{F}_T^*(Z) \rangle$. Then, we have M' run M on $\langle \langle \mathcal{F}_T^*(Z) \rangle, \{z\} \cup T, p(q_0, \omega) \rangle$, where $(q_0, \omega) \in \mathbb{N}^2$ when considered as numbers in base b . If M accepts, then M' accepts; otherwise, M' rejects.

We now claim that M' accepts if and only if Z halts on input ω in pair $(0, 0)$.

If Z halts on input ω in pair $(0, 0)$, then this means that Z converts complete state pair (q_0, ω) to $(0, 0)$. Then, we can convert $p(q_0, \omega)$ to $p(0, 0)$ using the corresponding isomorphisms ϕ_{ℓ_i} and ϕ_{r_i} , and as such, by a series of conjugations by T . As such, we have

$$\omega_t^{-1} p(q_0, \omega) \omega_t = p(0, 0) = z \text{ for some } \omega_t \in \langle T \rangle_{\mathcal{F}_T^*(Z)}.$$

Thus, we have $p(q_0, \omega) = \omega_t z \omega_t^{-1} \in \langle \{z\} \cup T \rangle_{\mathcal{F}_T^*(Z)}$. Thus, M accepts, so M' accepts, as desired.

Now, if M' accepts on input ω , then this means that $p(q_0, \omega) \in \langle \{z\} \cup T \rangle_{\mathcal{F}_T^*(Z)}$. Note that $z = p(0, 0) \in \mathcal{F}_0(Z)$, so we must have $p(q_0, \omega) \in \langle \mathcal{F}_0(Z) \cup T \rangle_{\mathcal{F}_T^*(Z)}$. By Lemma 5.4, we have $p(q_0, \omega) \in \mathcal{F}_0(Z)$. Thus, by definition of $\mathcal{F}_0(Z)$, Z converts (q_0, ω) to $(0, 0)$. Thus, Z halts on input ω in pair $(0, 0)$, as desired.

Thus, M' decides $\text{HALT}_{\mathbb{Z}^2}$. However, $\text{HALT}_{\mathbb{Z}^2}$ is undecidable by Lemma 3.3, so we have a contradiction. Thus, MAGNUS is undecidable. \square

⁶Note that Higman [12] proved a generalized version of this result without the use of Britton's lemma, based on subgroups of HNN extensions. However, Britton's lemma provides a simpler proof, which we use here.

Corollary 5.6. *The Magnus problem on $\mathcal{F}_T^*(Z)$ and subgroup generated by $\{z\} \cup T$ is undecidable.*

Proof. Denote this case of the Magnus problem by

$$\text{MAGNUS}_{\mathcal{F}_T^*(Z)} = \{\langle U, V \rangle \mid U, V \in \mathbb{N}^2 \text{ and } p(U, V) \text{ is in the subgroup generated by } \{z\} \cup T\}.$$

In the proof of Theorem 5.5, we instead assume that $\text{MAGNUS}_{\mathcal{F}_T^*(Z)}$ is decidable, and run M on input $\langle q_0, \omega \rangle$, where q_0 and ω are taken as elements of \mathbb{N} when considered in base b . The rest of the proof applies as written. \square

6. Word problem

We now prove the word problem from the Magnus problem. In particular, we use a group that was first introduced by Boone [5] in his original proof of the word problem; we prove the undecidability of the word problem on the HNN extension of $\mathcal{F}_T^*(Z)$ with stable letter k given by the identity isomorphism

$$\gamma : z \mapsto z, T \mapsto T,$$

from the subgroup of $\mathcal{F}_T^*(Z)$ generated by $\{z, T\}$ to itself. Let us denote this group by $\mathcal{K}_T^*(Z)$, and note that more explicitly, we have

$$\mathcal{K}_T^*(Z) = \mathcal{F}_T^*(Z) \cup \langle k \mid k^{-1}zk = z \text{ and for all } t \in T, k^{-1}tk = t \rangle.$$

Theorem 6.1. *The word problem is undecidable.*

Proof. Assume WORD is decidable by a TM W . We construct a TM W' to decide $\text{MAGNUS}_{\mathcal{F}_T^*(Z)}$, with input $\langle U, V \rangle$.

First, we have W' construct the presentation for $\mathcal{K}_T^*(Z)$, as given earlier in this section and as based on the HNN extension in Section 4.1; denote this presentation by $\langle \mathcal{K}_T^*(Z) \rangle$. Then, we have W' run W on $\langle \langle \mathcal{K}_T^*(Z) \rangle, kp(U, V)k^{-1}p(U, V)^{-1} \rangle$. If W accepts, then W' accepts, and otherwise, W' rejects.

We now claim that W' accepts if and only if $p(U, V) \in \langle z, T \rangle_{\mathcal{F}_T^*(Z)}$.

If $p(U, V) \in \langle z, T \rangle$, then since k commutes with z and T , we have $kp(U, V)k^{-1}p(U, V)^{-1} = kk^{-1}p(U, V)p(U, V)^{-1} = 1$. Thus, W accepts, so W' accepts, as desired.

If W' accepts, then this means that $kp(U, V)k^{-1}p(U, V)^{-1} = 1$. By Britton's lemma, applied to the extension involving stable letter k , we have $p(U, V) \in \langle z, T \rangle$.

Thus, W' decides $\text{MAGNUS}_{\mathcal{F}_T^*(Z)}$. However, $\text{MAGNUS}_{\mathcal{F}_T^*(Z)}$ is undecidable by Corollary 5.6, which is a contradiction. Thus, WORD is undecidable. \square

Corollary 6.2. *The word problem on $\mathcal{K}_T^*(Z)$ is undecidable.*

Proof. Denote this case of the word problem by

$$\text{WORD}_{\mathcal{K}_T^*(Z)} = \{\langle \omega \rangle \mid \omega \in \mathcal{K}_T^*(Z) \text{ and } \omega = 1\}.$$

In the proof of Theorem 6.1, we instead assume that $\text{WORD}_{\mathcal{K}_T^*(Z)}$ is decidable, and run W on input $\langle kp(U, V)k^{-1}p(U, V)^{-1} \rangle$. The rest of the proof applies as written. \square

7. Isomorphism problem

Now, using $\mathcal{K}_T^*(Z)$, we can prove the undecidability of the isomorphism problem. The isomorphism problem was first proven undecidable by Adian [2] and Rabin [24], but in fact, Adian and Rabin (independently) proved stronger results regarding the Markov property, which we discuss in Section 8. Here, we give a proof that follows directly from the word problem, and in fact, this proof works for any group with undecidable word problem such that every element $\neq 1$ in that group has infinite order. This idea was first suggested by Novikov [22].

Theorem 7.1. *The isomorphism problem is undecidable.*

Proof. We first show that every element $\neq 1$ in $\mathcal{K}_T^*(Z)$ has infinite order. First, note that for any group \mathcal{H} and HNN extension \mathcal{H}_t^* , elements in \mathcal{H} retain their order in \mathcal{H}_t^* . This is clear by Corollary 4.2, since we have an embedding $\mathcal{H} \hookrightarrow \mathcal{H}_t^*$. Moreover, t has infinite order in \mathcal{H}_t^* ; this is clear because for any $n \in \mathbb{N}$, the normal form of t^n is given by $1 \cdot t \cdot \dots \cdot t \cdot 1$, and by the contrapositive of Britton's lemma (Corollary 4.3), $t^n \neq 1$. Now, since $\mathcal{K}_T^*(Z)$ was derived from successive HNN extensions on the free group $\mathcal{F} = \langle x, y, z \mid - \rangle$, where clearly every element $\neq 1$ has infinite order, necessarily every element $\neq 1$ in $\mathcal{K}_T^*(Z)$ has infinite order.

Now, we assume that ISO is decidable by a TM I . We construct a TM I' to decide $\text{WORD}_{\mathcal{K}_T^*(Z)}$ with input $\langle \omega \rangle$.

First, we have I' construct a presentation of $\mathcal{K}_T^*(Z)$ as based on the HNN extension in Section 4.1. We rename the generators of $\mathcal{K}_T^*(Z)$ to be a_1, \dots, a_p , and then we have I' add to its presentation of $\mathcal{K}_T^*(Z)$ to construct

$$\mathcal{K}_{\omega, T}(Z) = \mathcal{K}_T^*(Z) \cup \langle k_1, \dots, k_p \mid \{k_i^{-1} \omega k_i = a_i \mid 1 \leq i \leq p\} \rangle.$$

Denote the presentation of $\mathcal{K}_{\omega, T}(Z)$ by $\langle \mathcal{K}_{\omega, T}(Z) \rangle$. Then, we have I' run I on $\langle \langle \mathcal{K}_{\omega, T}(Z) \rangle, \langle x_1, \dots, x_p \mid - \rangle \rangle$. If I accepts, then I' accepts, and otherwise, I' rejects.

We now claim that I' accepts if and only if $\omega = 1$ in $\mathcal{K}_T^*(Z)$.

If $\omega = 1$ in $\mathcal{K}_T^*(Z)$, then for all $1 \leq i \leq p$, we have $a_i = 1$ in $\mathcal{K}_{\omega, T}(Z)$. Thus, we are left with $\mathcal{K}_{\omega, T}(Z) = \langle k_1, \dots, k_p \mid - \rangle$, which is trivially isomorphic to $\langle x_1, \dots, x_p \mid - \rangle$. Thus, I accepts, so I' accepts.

To show the other direction, we in fact show the contrapositive, namely that if $\omega \neq 1$ in $\mathcal{K}_T^*(Z)$, then I' rejects. Since $\omega \neq 1$ in $\mathcal{K}_T^*(Z)$, ω must be of infinite order. As such, for each a_i , we have an isomorphism $\zeta : \omega \mapsto a_i$, so $\mathcal{K}_{\omega, T}(Z)$ can be constructed from $\mathcal{K}_T^*(Z)$ through a series of HNN extensions. Then, since by Corollary 4.2 we have an embedding $\mathcal{K}_T^*(Z) \hookrightarrow \mathcal{K}_{\omega, T}(Z)$, and since by Corollary 6.2 the word problem on $\mathcal{K}_T^*(Z)$ is undecidable, we must have that the word problem on $\mathcal{K}_{\omega, T}(Z)$ is undecidable (the reduction to do this is trivial). Now, we claim that $\mathcal{K}_{\omega, T}(Z)$ is not isomorphic to $\langle x_1, \dots, x_p \mid - \rangle$, since the word problem on $\langle x_1, \dots, x_p \mid - \rangle$ is decidable.

First, note that the word problem on free groups is decidable, because a TM needs only repeatedly cancel subwords of the form $x_i x_i^{-1}$. This results in a reduced form, and since on free groups the reduced form is unique, the TM needs only check if the reduced form is exactly 1. If so, the TM accepts, and if not, the TM rejects.

Now, note that if $\mathcal{K}_{\omega, T}(Z)$ were isomorphic to $\langle x_1, \dots, x_p \mid - \rangle$, then we could construct a TM to decide the word problem on $\mathcal{K}_{\omega, T}(Z)$ with input $\langle \omega \rangle$. The TM first transforms $\mathcal{K}_{\omega, T}(Z)$ to

$\langle x_1, \dots, x_p \mid - \rangle$ by *non-deterministically*⁷ applying *Tietze transformations*⁸ until $\mathcal{K}_{\omega, T}(Z)$ has no relations left. Since $\mathcal{K}_{\omega, T}(Z)$ is isomorphic to $\langle x_1, \dots, x_p \mid - \rangle$, the TM would eventually guess a correct sequence of operations that sends $\mathcal{K}_{\omega, T}(Z)$ to $\langle x_1, \dots, x_p \mid - \rangle$. While guessing these operations, we have the TM also modify ω based on modifications to $\mathcal{K}_{\omega, T}(Z)$, and once we have transformed $\mathcal{K}_{\omega, T}(Z)$ into a free group, the TM applies the previous algorithm on deciding the word problem for free groups to check if $\omega = 1$. If so, the TM accepts, and otherwise, the TM rejects. This is a contradiction, since the word problem on $\mathcal{K}_{\omega, T}(Z)$ is not decidable.

Thus, $\mathcal{K}_{\omega, T}(Z)$ is not isomorphic to $\langle x_1, \dots, x_p \mid - \rangle$, so we have I rejects. As such, I' rejects.

Thus, I' decides $\text{WORD}_{\mathcal{K}_T^*(Z)}$. However, by Corollary 6.2, $\text{WORD}_{\mathcal{K}_T^*(Z)}$ is undecidable, which is a contradiction. Thus, ISO is undecidable. \square

8. Markov property

The Markov property was first introduced by Markov [15, 16, 17] in reference to the same decidability problem on semigroups. Adian [2] and Rabin [24] independently proved that questions on finitely presented groups involving properties satisfying the Markov property are undecidable. This includes isomorphic (which we proved in Section 7), abelian, finite, trivial, torsion-free, free, and simple.

We begin by reiterating the definition of the Markov property.

Definition 8.1. Let P be any property of finitely presented groups that is preserved under isomorphism. P is a *Markov property* if

1. there is a finitely presented group with property P and
2. there is a finitely presented group that cannot be embedded into any finitely presented group with property P .

We use Rabin's [24] construction to prove the undecidability of properties satisfying the Markov property. To achieve the first step of this construction, we need the following lemma.

Lemma 8.1. *Every countable group \mathcal{H} can be embedded into a group \mathcal{G} generated by two elements of infinite order. If \mathcal{H} is finitely presented, then \mathcal{G} is finitely presented.*

Proof. Let \mathcal{H} consist of the elements h_1, h_2, \dots . Let \mathcal{F} be the free product of \mathcal{H} and $\langle a, b \rangle$, namely $\mathcal{H} \cup \langle a, b \rangle$. First, note that the set

$$\{a, b^{-1}ab, b^{-2}ab^2, \dots, b^{-n}ab^n, \dots\}$$

⁷*Non-deterministic* Turing machines are slightly outside the scope of this paper, but in short, they are exactly deterministic Turing machines except at any step in the computation, they can (finitely) branch and essentially “guess” a step to take. If any of these branches accept the input, then the entire machine accepts the input. It is simple to show that non-deterministic Turing machines have equivalent deterministic Turing machines, which we also omit for the purposes of this paper.

⁸*Tietze transformations* are outside the scope of this paper, but in short, there are four operations, namely adding/removing relations and adding/removing generators, such that given two isomorphic finitely presented groups, some finite sequence of those operations will transform one group to the other. They were first introduced by Tietze [30] when he formulated the isomorphism problem on finitely presented groups. Moreover, by having a non-deterministic Turing machine repeatedly guess Tietze transformations, it is clear that the isomorphism problem is recognizable.

freely generates a subgroup of $\langle a, b \rangle$ (this follows by the Nielsen-Schreier Theorem⁹). Moreover,

$$\{b, h_1 a^{-1} b a, h_2 a^{-2} b a^2, \dots, h_n a^{-n} b a^n, \dots\}$$

freely generates a subgroup of \mathcal{F} . This is clear because we can consider the projection of \mathcal{F} onto $\langle a, b \rangle$ given by $\pi : a \mapsto a, b \mapsto b, c_i \mapsto 1$ for all i . Since the images $\{b, a^{-1} b a, a^{-2} b a^2, \dots\}$ are free generators, we must have $\{b, h_1 a^{-1} b a, h_2 a^{-2} b a^2, \dots\}$ are free generators as well.

Thus, we have an isomorphism given by $v : a \mapsto b, b^{-i} a b \mapsto h_i a^{-i} b a$. Let \mathcal{G} be the HNN extension of \mathcal{F} with stable letter t given by v . More concretely, we have

$$\mathcal{G} = \mathcal{F} \cup \langle t \mid t^{-1} a t = b, t^{-1} b^{-i} a b^i t = h_i a^{-i} b a^i \text{ for all } i \rangle.$$

Thus, \mathcal{H} is embedded into \mathcal{G} by Corollary 4.2. Moreover, \mathcal{G} is generated by a and t , which is clear by inspection of the relations. Also by Corollary 4.2, we have a is of infinite order, and by the normal form from the HNN extension, it is clear that t is of infinite order (this is also shown in the proof of Theorem 7.1).

Finally, we claim that if \mathcal{H} is finitely presented, then we can remove all relations $t^{-1} b^{-i} a b^i t = h_i a^{-i} b a^i$ where h_i is not a generator in the presentation of \mathcal{H} ; this can be somewhat trivially shown, and is formally true by the Tietze transformations.¹⁰ Thus, we have \mathcal{G} is finitely presented. \square

Now, we prove the Adian-Rabin theorem, largely following Rabin's [24] proof.

Theorem 8.2. *Adian-Rabin theorem [2, 24]. Let P be a Markov property of finitely presented groups. The question of whether a finitely presented group has property P is undecidable.*

Proof. Let \mathcal{G}_+ denote a finitely presented group with property P , and let \mathcal{G}_- denote a finitely presented group that cannot be embedded into any finitely presented group with property P . Let $\mathcal{H} = \mathcal{K}_T^*(Z)$ (note that \mathcal{H} can be any finitely presented group with undecidable word problem, for the purposes of this proof).

For any $\omega \in \mathcal{H}$, we make the following constructions. First, by Lemma 8.1, we can embed $\mathcal{G}_- \cup \mathcal{H} \cup \langle x \rangle$ into a group generated by two elements of infinite order, say \mathcal{U} , generated by u_1 and u_2 . Then, let \mathcal{J} be the HNN extension of \mathcal{U} with stable letters y_1 and y_2 given by the isomorphisms $\psi_1 : u_1 \mapsto u_1^2$ and $\psi_2 : u_2 \mapsto u_2^2$ respectively. Thus, we have

$$\mathcal{J} = \mathcal{U} \cup \langle y_1, y_2 \mid y_1^{-1} u_1 y_1 = u_1^2, y_2^{-1} u_2 y_2 = u_2^2 \rangle.$$

Then, let \mathcal{K} be the HNN extension of \mathcal{J} with stable letter z given by the isomorphism $\chi : y_1 \mapsto y_1^2, y_2 \mapsto y_2^2$. This gives us

$$\mathcal{K} = \mathcal{J} \cup \langle z \mid z^{-1} y_1 z = y_1^2, z^{-1} y_2 z = y_2^2 \rangle.$$

Now, let

$$\mathcal{Q} = \langle r, s, t \mid s^{-1} r s = r^2, t^{-1} s t = s^2 \rangle.$$

⁹The Nielsen-Schreier Theorem states that every subgroup of a free group is free [20, 25]. The proof of this is outside the scope of this paper.

¹⁰See footnote 8.

Note that if we let $\mathcal{P} = \langle r, s \mid s^{-1}rs = r^2 \rangle$, \mathcal{P} is exactly the HNN extension of $\langle r \rangle$ with stable letter s given by the isomorphism $\rho : r \mapsto r^2$. Moreover, \mathcal{Q} is exactly the HNN extension of \mathcal{P} with stable letter t given by the isomorphism $\sigma : s \mapsto s^2$.

Finally, let

$$\mathcal{D}_\omega = \mathcal{K} \cup \mathcal{Q} \cup \langle - \mid r = z, t = [\omega, x] \rangle$$

(essentially, \mathcal{D}_ω is the free product $\mathcal{K} * \mathcal{Q}$ with the added relations $r = z$ and $t = [\omega, x]$), and let

$$\mathcal{E}_\omega = \mathcal{D}_\omega \cup \mathcal{G}_+.$$

This concludes the necessary constructions.

Assume MARKOV_P is decidable by a TM M . We construct a TM M' to decide $\text{WORD}_{\mathcal{K}_T^*(Z)}$, with input $\langle \omega \rangle$.

First, we have M' construct the presentation of \mathcal{E}_ω , as given previously; we denote this presentation by $\langle \mathcal{E}_\omega \rangle$. Then, we have M' run M on $\langle \langle \mathcal{E}_\omega \rangle \rangle$, and if M accepts, then M' accepts, and otherwise, M' rejects.

We now claim M' accepts if and only if $\omega = 1$ in \mathcal{H} .

Case: First, we show that if $\omega \neq 1$ in \mathcal{H} , then M' rejects.

We claim that $\{r, t\}$ generates a free subgroup of \mathcal{Q} . Let α be a nontrivial reduced word on r and t , such that $\alpha = 1$. By Britton's lemma (Corollary 4.3) applied to \mathcal{Q} , α must contain a subword $t^{-\varepsilon}\beta t^\varepsilon$ where $\varepsilon = \pm 1$, and $\beta \in \langle s \rangle_{\mathcal{P}}$ if $\varepsilon = 1$, and $\beta \in \langle s^2 \rangle_{\mathcal{P}}$ if $\varepsilon = -1$. Note that as such, we have $\beta = s^m$ for some $m \neq 0$. Also, since α is a freely reduced word on r and t , we must have $\beta = r^n$ for some $n \neq 0$ (if β contains t or t^{-1} , then α contains a shorter subword of the form $t^{-\varepsilon}\beta t^\varepsilon$, and we can repeat this process until β is of the desired form). Thus, we have $s^m = r^n \Rightarrow s^m r^{-n} = 1$. But, by Britton's lemma (Corollary 4.3) applied to \mathcal{P} , since we have a word that equals 1 without the specified subwords, we have a contradiction. Thus, r and t freely generate a subgroup of rank 2, $\langle r, t \rangle_{\mathcal{Q}}$.

Now, since $\omega \neq 1$, note that the commutator $[\omega, x] = \omega^{-1}x^{-1}\omega x$ has infinite order in \mathcal{U} . We claim that $[\omega, x]$ and z generate a free subgroup of rank 2 of \mathcal{K} ; the proof of this is exactly the same as the proof for r and t in the previous paragraph. Note that we trivially have embeddings $\langle r, t \rangle \hookrightarrow \mathcal{Q}$ and $\langle [\omega, x], z \rangle \hookrightarrow \mathcal{K}$ for $\mathcal{F} \cong \langle r, t \rangle \cong \langle [\omega, x], z \rangle$, so we have the free product with amalgamation

$$\mathcal{D} = \mathcal{K} *_F \mathcal{Q}.$$

Note that by construction, we have $\mathcal{D}_\omega = \mathcal{D}$. Note that as such, we have \mathcal{G}_- is embedded into \mathcal{D}_ω , and as such, into \mathcal{E}_ω . Thus, \mathcal{E}_ω cannot have the property P , so M rejects. Thus, M' rejects, as desired.

Case: Now, we show that if $\omega = 1$ in \mathcal{H} , then M' accepts.

Since $\omega = 1$, then by inspection, we have $\mathcal{D}_\omega = \{1\}$. Thus, $\mathcal{E}_\omega = \mathcal{G}_+$, so trivially, \mathcal{E}_ω has property P . Thus, M accepts, so M' accepts, as desired.

Thus, we have M' decides $\text{WORD}_{\mathcal{K}_T^*(Z)}$. However, this contradicts Corollary 6.2. Thus, MARKOV_P is undecidable. \square

9. Acknowledgments

I would like to thank Professor Adam Levine, for his invaluable help and guidance in this work and throughout the Knot Theory Junior Seminar. I would also like to thank all of my classmates in the seminar, for their insightful presentations and constant support.

This paper represents my own work in accordance with university regulations. /s/ Jessica Shi.

References

- [1] S. Aanderaa and D. E. Cohen. Modular machines, the word problem for finitely presented groups and Collins' theorem. In S. I. Adian, W. W. Boone, and G. Higman, editors, *WORD PROBLEMS II*, volume 95 of *Studies in Logic and the Foundations of Mathematics*, pages 1–16. Elsevier, 1980.
- [2] S. I. Adian. Unsolvability of some algorithmic problems in the theory of groups. *Tr. Mosk. Mat. Obs.*, 6:231–298, 1957.
- [3] E. Artin. Theory of braids. *Annals of Mathematics*, 48(1):101–126, 1947.
- [4] W. W. Boone. Certain simple, unsolvable problems of group theory i-iv. *Journal of Symbolic Logic*, 22(4):372–373, 1957.
- [5] W. W. Boone. The word problem. *Annals of Mathematics*, 70(2):207–265, 1959.
- [6] J. L. Britton. The word problem. *Annals of Mathematics*, 77(1):16–32, 1963.
- [7] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.
- [8] A. Church. *The Journal of Symbolic Logic*, 3(1):45–46, 1938.
- [9] M. Dehn. Über die Topologie des dreidimensionalen Raumes. *Math. Ann.*, 69:137–168, 1910.
- [10] M. Dehn. Die beiden Kleeblattschlingen. *Math. Ann.*, 75:402–413, 1914.
- [11] K. Gödel. On Undecidable Propositions of Formal Mathematical Systems. In B. Meltzer, editor, *Lecture Notes Taken by Kleene and Rosser at the Institute for Advanced Study. Reprinted in Davis, M. (Ed.) 1965. The Undecidable*. New York: Raven, 1934.
- [12] G. Higman. Subgroups of finitely presented groups. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 262(1311):455–475, 1961.
- [13] G. Higman, B. H. Neumann, and H. Neuman. Embedding theorems for groups. *Journal of the London Mathematical Society*, s1-24(4):247–254, 1949.
- [14] W. Magnus. Das Identitätsproblem für Gruppen mit einer definierenden Relation. *Math. Ann.*, 106:295–307, 1932.
- [15] A. A. Markov. On certain unsolvable problems concerning matrices. *Dokl. Akad. Sci. USSR*, 57:539–542, 1947.
- [16] A. A. Markov. On the impossibility of certain algorithms in the theory of associative systems. *Dokl. Akad. Sci. USSR*, 55:583–586, 1947.
- [17] A. A. Markov. On the impossibility of certain algorithms in the theory of associative systems II. *Dokl. Akad. Sci. USSR*, 58:353–356, 1947.
- [18] A. A. Markov. Insolubility of the problem of homeomorphy. *Proc. Internat. Congr. Math.*, pages 300–306, 1958.
- [19] M. L. Minsky. Recursive unsolvability of Post's problem of "Tag" and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961.
- [20] J. Nielsen. Om Regning med ikke-kommutative Faktorer og dens Anvendelse i Gruppeteorien. *Matematisk Tidsskrift. B*, pages 77–94, 1921.
- [21] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov.*, 4:3–143, 1955.
- [22] P. S. Novikov. Über einige algorithmische probleme der gruppentheorie. *Jber. Deutsch. Math. Verein.*, 61:88–92, 1958.
- [23] E. L. Post. Recursive unsolvability of a problem of Thue. *The Journal of Symbolic Logic*, 12(1):1–11, 1947.
- [24] M. O. Rabin. Recursive unsolvability of group theoretic problems. *Annals of Mathematics*, 67(1):172–194, 1958.
- [25] O. Schreier. Die untergruppen der freien gruppen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 5, pages 161–183. Springer, 1927.
- [26] P. E. Schupp. Some reflections on hnn extensions. In M. F. Newman, editor, *Proceedings of the Second International Conference on the Theory of Groups: Australian National University, August 13–24, 1973*, pages 611–632. Springer Berlin Heidelberg, Berlin, Heidelberg, 1974.
- [27] M. Sipser. *Introduction to the Theory of Computation*. Cengage Learning, Boston MA, 3rd edition, 2013.
- [28] J. Stillwell. The word problem and the isomorphism problem for groups. *Bulletin of the American Mathematical Society*, 6(1):33–56, 1982.
- [29] J. Stillwell. *Classical Topology and Combinatorial Group Theory*. Springer-Verlag, New York NY, 2nd edition, 1993.
- [30] H. Tietze. Über die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten. *Monatsh. f. Math. u. Phys.*, pages 1–118, 1908.
- [31] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.
- [32] F. Waldhausen. The word problem in fundamental groups of sufficiently large irreducible 3-manifolds. *Annals of Mathematics*, 88(2):272–280, 1968.