POLYNOMIAL-TIME COMPLEXITY FOR INSTANCES OF THE ENDOMORPHISM PROBLEM IN FREE GROUPS

LAURA CIOBANU

Abstract. We say the endomorphism problem is solvable for an element W in a free group F if it can be decided effectively whether, given U in F, there is an endomorphism ϕ of F sending W to U. This work analyzes an approach due to C. Edmunds and improved by C. Sims. Here we prove that the approach provides an efficient algorithm for solving the endomorphism problem when W is a two-generator word. We show that when W is a two-generator word this algorithm solves the problem in time polynomial in the length of U.

This result gives a polynomial-time algorithm for solving, in free groups, two-variable equations in which all the variables occur on one side of the equality and all the constants on the other side.

1. Introduction

The endomorphism problem is solvable for an element W in a free group F if it can be decided effectively whether, given U in F, there is an endomorphism ϕ of F sending W to U. For an arbitrary group G the corresponding problem is undecidable. For example, if G is free nilpotent of countable rank and nilpotence class at least 9, then the problem is undecidable [15]. In free groups the endomorphism problem is equivalent to solving an equation in which all the variables occur on one side of the equality and all the constants on the other side. Makanin showed that the endomorphism problem is always solvable in free groups by providing an algorithm for finding solutions to arbitrary equations [7]. Razborov has then generalized Makanin's result to systems of equations in free groups [14]. However, Makanin's use of very difficult group theoretical techniques makes this approach impractical. Additional evidence of the impracticability comes from the fact that the time complexity for Makanin's algorithm is not primitive recursive [10]. Therefore our goal has been to provide practical alternatives to Makanin's method.

The starting point for this work is an approach due to C. Edmunds [1],[2] that has been improved and made algorithmically formal by C. Sims [3]. While here we consider the approach for an infinitely generated free group

F, Edmunds has proved [6] that solving the endomorphism problem for a word W in F is equivalent to solving the problem in a finitely generated free group with the same number of generators as the word W. It has been shown that the Edmunds approach solves the endomorphism problem when W belongs to some restricted classes of words, such as quadratic words, examples of which are products of commutators. However, it has been unclear whether this approach solves the problem for any other type of words. That is, it has been unclear whether one can obtain an answer to this decision problem after performing only a finite number of the steps suggested by the Edmunds-Sims method.

Here we prove that the Edmunds-Sims approach provides an algorithm in the case when W is a two-generator word. For W a two-generator word, we modify the existing procedures and find that after a finite number of steps we are guaranteed to obtain an answer to our decision problem. Moreover, we show that this now established algorithm runs in time polynomial in the length of U, if W is a two-generator word. We intend to treat the case when W is a word on an arbitrary number of generators in a future paper, but in this case the complexity appears to be not polynomial.

The main idea of Edmunds' approach is to define a subset D_W for each W in F which forms a 'basis' for the set of endomorphic images of W in the sense of Theorem 1. In order to state the theorem we define the following. Given two cyclically reduced words V and V' in F we say that V' is a *proper image* of V if there exists an endomorphism ϕ with $\phi(V) = V'$ that sends every generator x in V to a non-identity element of F and no cancellations occur in $\phi(V)$. We say V' is a *proper cyclic image* of V if some cyclic permutation of V' is a proper image of V.

Theorem 1 (Edmunds [2])**.** *A cyclically reduced word U is an endomorphic image of W if and only if U is a proper cyclic image of some $W' \in D_W$.*

In [3] Sims constructs a set C_W similar to D_W in that it satisfies Theorem 1. The set C_W has the advantage that it is a subset of D_W in which most of the duplicate and less relevant words of D_W are eliminated. The exact description of the algorithm used to generate the set C_W can be found in Section 4. Checking for proper and cyclic images is a polynomial-time operation. Thus the solvability and efficiency of the endomorphism problem for W depends heavily on the tractability of the set C_W. For most words W the set C_W is infinite and has a very complicated structure.

In this paper we analyze the set C_W for two-generator words W. The fact that C_W is infinite suggests that the program does not terminate. However, we are able to find upper bounds on the number of words in C_W that need to be generated before obtaining the solution. One useful representation of the set C_W is as a tree with the nodes representing the words in the set C_W

and the edges representing the endomorphisms that connect the words, as we describe in Section 5. One can then associate to each node coordinates that represent the number of occurences of each generator in the word. In Sections 7, 8 and 9 we show that when W is a two-generated word, this tree can be partitioned into finitely many subtrees whose coordinates have a 'nice' behavior. We call these subtrees *almost perfect trees*. We concentrate most of the technical results that show this behavior in Section 8. The proofs rely on the analysis of automorphisms in the free group on two generators.

The key ingredient in computing the complexity of the endomorphism problem is the following theorem.

Theorem 2. *Let W be a two-generator word. The number of words in C_W that are generated before finding the last word of length at most n is quadratic in n, i.e. it is smaller than cn^2, where c is a constant depending on the length of W.*

By Theorem 1, in order to determine whether U is an endomorphic image of a word W it is sufficient to check whether there exists W' in C_W such that U is a proper cyclic image of W'. As the length of a proper image of a word is at least as long as the length of the word, we only need to perform the check for the words W' in C_W of length at most the length of U.

The above results show that one only needs to check a quadratic number of words when W is a two-generator word, and since all the other operations used in the implementation of the algorithm require polynomial time and amount of space, the endomorphism problem for two-generator words is polynomial in the length of U. We give more details on the complexity in Section 10.

2. Other results on the endomorphism problem in free groups

Efficient solutions to the endomorphism problem in free groups have been known for a while in a few cases. Let F and M be the free group and free monoid, respectively, on the generating set $A = \{a, b, \dots\}$ (more precisely, $A^{\pm}$ is the generating set for M). We think of the elements of M as words in $A^{\pm}$, and we think of the elements of F as equivalence classes of words under free equivalence. We denote by $[U]$ the equivalence class containing the word U.

If $W = a^k$ and U a cyclically reduced word in F, where k is a positive integer, then $[U]$ is an endomorphic image of $[W]$ if and only if U is of the form v^k, for some word v in M. If $W = [a, b]$ we have the following result by Wicks.

Proposition 1 (Wicks [18])**.** *Let U be a cyclically reduced word and let $W = [a, b]$. Then $[U]$ is an endomorphic image of $[W]$ if and only if some*

cyclic permutation of U is of the form $c\ d\ f\ c^{-1}d^{-1}f^{-1}$, for some $c, d, f \in M$.

In [17] P. Schupp shows that the endomorphism problem is solvable for a two-generator word W. Computationally, Schupp's approach is exponential in the length of U.

The endomorphism problem in free groups is equivalent to solving a certain type of equation. We have already mentioned the seminal work of Makanin and Razborov which provides algorithms for solving systems of arbitrary equations in the free group. As their algorithm is difficult and impractical, much work has been done to find different approaches to deal with equations in free groups. Particular attention has been devoted to quadratic equations, which are now well understood. Results in this direction have been obtained by Edmunds, Comerford [5], Lyndon [11], Mal'cev [12], Schupp [16], Grigorchuk [8], Kurchanov and many others.

3. Preliminaries

Let us fix a countably infinite set $X = \{a, b, c, \dots\}$ and let X^{-1} be a set of formal inverses for the elements of X, that is, $X^{-1} = \{a^{-1}, b^{-1}, c^{-1}, \dots\}$. Let $X^{\pm} = X \cup X^{-1}$. Elements of X will be called *generators* and elements of $X^{\pm}$ will be called *letters*. For $x \in X$ set $(x^{-1})^{-1} = x$.

A finite string of letters is called a *word*. We define the inverse of a word $U = x_1 \dots x_n$ to be $U^{-1} = x_n^{-1} \dots x_1^{-1}$. The length of U will be denoted by $|U|$. For a word W, a string of consecutive letters in W forms a *subword* of W. A word W is *freely reduced* if it contains no subword of the form xx^{-1} with x in $X \cup X^{-1}$ and W is *cyclically reduced* if all cyclic permutations of W are freely reduced. The empty word will be denoted by e. Let M be the free monoid with basis $X^{\pm}$ and let F be the free group generated by X. The elements of F are equivalence classes of words under free equivalence. We denote by $[U]$ the equivalence class containing the word U. (For simplicity, in the Introduction we have denoted the elements of F by U instead of $[U]$, and we have blurred the distinction between words, that is, elements of M, and elements of F.)

We will often consider *cyclic words* in the free group. A *cyclic word* is the equivalence class of a given cyclically reduced word under all cyclic permutations. Let U be a cyclic word. A generator x occurs in U if x or x^{-1} is in U. Let $(x)_U$ be the number of occurences of x and x^{-1} in U. More generally, for a word V we say that V occurs in U if there exists a word U' that is a cyclic permutation of U such that V or V^{-1} is a consecutive string of letters in U'. Let $(V)_U$ denote the number of occurences of the subwords V and V^{-1} in U. Let $\{V\}_U$ be the number of occurences of subword V in the cyclic word U. Thus $(V)_U = \{V\}_U + \{V^{-1}\}_U$.

We define a *syllable* of a cyclic word U with $|U| \geq 2$ to be any subword (or its inverse) of length 2 in U. Every letter in the word is thus a member of exactly two syllables. Let x be a letter. We define an *x-segment* to be any maximal subword in x only.

Example. *Let U be $aaba^{-1}a^{-1}a^{-1}b$. Then its syllables are aa, ab, ba^{-1}, $a^{-1}b, ba$ and its segments are aa, $a^{-1}a^{-1}a^{-1}$, and b. The word a^3 is not considered to be a segment of U.*

We shall often use monoid endomorphisms ψ of M. All such endomorphisms will be assumed to be compatible with the inverse map of M. That is, $\psi(U^{-1}) = \psi(U)^{-1}$ for all elements U of M. We will also use monoid automorphisms of M. These correspond to the permutations on the letters that are compatible with the inverse map.

Definition. Given two cyclically reduced words V and V' we say that V' is a *proper image* of V if:

(1) there exists an endomorphism $\phi : M \longmapsto M$ with $\phi(V) = V'$ and
(2) ϕ sends every generator x in V to a non-identity element.

Definition. Given two cyclically reduced words V and V' in M we say that V' is a *proper cyclic image* of V if some cyclic permutation of V' is a proper image of V.

Example. *The word $V = acacdac^{-1}a^{-1}da$ is a proper image of $U = aaba^{-1}b$ via the endomorphism that sends a to ac and b to da. On the other hand, one can easily check that word $U' = aaaba^{-1}bb$ is not a proper image of U.*

The proper (cyclic) image relation between words is a transitive relation.

The following definition is motivated by the fact that one can often find a simpler equivalent form for the words involved in the endomorphism problem.

Definition. A cyclic word U is *redundant* if there are distinct letters x and y such that $y \neq x^{-1}$, both x and y occur in U, and all occurrences of x, x^{-1}, y and y^{-1} in U are in subwords xy or $y^{-1}x^{-1}$. A word that is not redundant is *irredundant*.

Example. *The cyclic word $U = cabcac^{-1}b^{-1}ab$ is redundant, as the only occurences of b and c are in subwords of the form bc or $c^{-1}b^{-1}$.*

If we let $U' = abab^{-1}ab$, the cyclic word obtained from U by deleting c, then U and U' are endomorphic images of each other. Thus deciding whether some word V is an endomorphic image of U can be reduced to deciding whether V is an endomorphic image of U'. This is why when we

consider the endomorphism problem for a word W we can assume W is irredundant.

Let us fix a linear order on $X^{\pm}$, say $a < a^{-1} < b < b^{-1} < \dots$. We will compare words using the corresponding length-plus-lexicographic ordering on words, which is a well-ordering on M.

Definition. For a word W let $\overline{W}$ be the first word in the orbit of $\mathrm{Aut}(M)$ containing W and let $\widehat{W}$ be the first word in the set of words obtained from W by applying automorphisms of M and cyclic permutations.

Algorithms for determining whether a word V is a proper image of a word U, and for computing $\overline{W}$ and $\widehat{W}$ will be described in the Appendix. Although so far we have defined the free monoids and groups to be infinitely generated, in reality we work with words on a finite number of generators, which makes it possible to decide the size of the input. Thus in all procedures we make the assumption that the number of generators in the free group is bounded. Under this assumption the computation of $\overline{W}$ and $\widehat{W}$, as well as the proper (cyclic) image algorithms run in a time polynomial in the length of the words involved.

4. Description of Sims' Approach

For each word W in F, Edmunds [2] defines a subset D_W of F which forms, in the sense of Theorem 1, a 'basis' for the set of endomorphic images of W. In [3] Sims constructs a set C_W similar to D_W in that it satisfies Proposition 2.

Proposition 2 (Sims [3])**.** *Let W and U be words with U cyclically reduced. Then $[U]$ is an endomorphic image of $[W]$ if and only if U is a proper cyclic image of some element of C_W.*

The set C_W has the advantage that it is a subset of D_W in which much of the duplication of D_W is eliminated. For example, no element of C_W is a proper cyclic image of any other element, while this is allowed in D_W. Since deciding whether a word is a proper cyclic image of another word is a relatively straightforward procedure, the solvability of the endomorphism problem for W depends on the tractability of the set C_W.

Before constructing C_W, we define certain endomorphisms of M introduced by Edmunds that mimic free reduction.

Definition.

1) Let U be a cyclic word and let x and y be letters such that xy (or $y^{-1}x^{-1}$) is a subword of U. Let z be a letter not in U. If $x \neq y$ we

define the *replacement endomorphism* $\rho_{x,y,z} : M \mapsto M$ which acts as:

$$\rho_{x,y,z}(x) = xz, \quad \rho_{x,y,z}(y) = z^{-1}y, \quad \rho_{x,y,z}(t) = t,$$

where t is any letter in U such that $t \notin \{x, y, x^{-1}, y^{-1}\}$.

If xx (or $x^{-1}x^{-1}$) is a subword of U we define the *replacement endomorphism* $\rho_{x,x,z} : M \mapsto M$ which acts as:

$$\rho_{x,x,z}(x) = zxz^{-1}, \quad \rho_{x,x,z}(t) = t,$$

where t is any generator in U such that $t \notin \{x, x^{-1}\}$.

While each replacement endomorphism depends on a particular word, the choice of z is arbitrary, and in particular $\widehat{\rho_{x,y,z}(U)}$ is independent of this choice.

2) For any subset A of generators we define the *annihilation endomorphism* $\tau_A : M \mapsto M$ to be the endomorphism that sends all generators x in A to the identity and fixes the rest of the generators. The endomorphism $\tau_\emptyset$ is simply the identity map.

We will modify slightly the previous notation for the annihilation endomorphisms. We will denote $\tau_{\{x_1,x_2\}}$ by τ_{x_1,x_2}, and $\tau_\emptyset$ by τ_e, i.e. we enumerate the elements of the set rather than write down the set, unless the set is $\emptyset$.

We say $\rho_{x,y,z}$ is *admissible* for a cyclic word U if xy is a subword of U and z does not occur in U. We say τ_x is *admissible* for word U if x occurs in U. A sequence of ρ's and τ's such as $\phi_n, \phi_{n-1}, \ldots, \phi_1$ is *admissible* for U if each ϕ_i is admissible for $\phi_{i-1}\phi_{i-2}\ldots\phi_1(U)$, where $1 \leq i \leq n$.

We now present the approach suggested by C. Sims in [3]. We define the sets that are the key elements in the approach.

4.1. **The set L_U and K_V.** For a cyclic word U, let L_U be the set of irredundant words $\widehat{W}$ that result from all the possible chains of admissible transformations

$$U \xrightarrow{\rho_{x,y,z}} \xrightarrow[reduction]{cyclic} V \xrightarrow{\tau_A} \xrightarrow[reduction]{cyclic} W \xrightarrow{\widehat{\ }} \widehat{W} \tag{1}$$

applied to U, where x is not necessarily different from y and A is a set of generators that occur in V. The word V is the cyclic reduction of $\rho_{x,y,z}(U)$ and the word W is the cyclic reduction of $\tau_A(V)$. Notice that since the final operation is $\widehat{\ }$, any choice of generator z not in U is allowed in $\rho_{x,y,z}$.

We can also describe L_U in a slightly different way. For a word V, let K_V be the set of irredundant words M obtainable by choosing a subset A of the generators occuring in V, letting W be the cyclic reduction of $\tau_A(V)$,

and setting $M = \widehat{W}$. The set L_U is then equal to the union of the sets K_V, where $V = \rho_{x,y,z}(U)$ for any admissible $\rho_{x,y,z}$.

Let us assume W is irredundant. We will give the formal description of the algorithm used to generate the set C_W as found in [3], but since the details of the algorithm are rather technical, we first present the main idea:

• Let C be the empty set;

• Place the word W in C, and construct the set L_W by applying sequences of replacement and annihilation endomorphisms to W as in (1);

• Add each irredundant word U in L_W to the set C and construct the sets L_U starting with the word U of minimal length;

• Repeat previous step for each word in C while constantly removing words that are proper cyclic images of words already in C.

The set C_W is the set of elements added to C and never removed.

4.2. **Formalization of C_W.** Here we describe the algorithm used to obtain the set C_W in more detail.

The procedure starts by placing K_W in a set Q. Let C be the set of words V in Q such that V is not a proper cyclic image of some element of $Q - \{V\}$. Assign Q to C. As long as the set Q does not become empty, execute commands (1) & (2):

(1) Choose an element $V \in Q$ of minimal length and delete it from Q;

(2) For all U in L_V, check whether U is a proper cyclic image of some element of C. If it is, do not add it to C. Otherwise delete from C and Q all words that are proper cyclic images of U and add U to both C and Q.

As before, C_W is the set of elements added to C during the above procedure and never removed. The procedure may not terminate, but if C_W is finite, then the procedure terminates and returns C_W.

Note that if at any point in the procedure there are two words V_1 and V_2 such that each is a proper cyclic image of the other then there is no danger that they would both be removed from Q since $\widehat{V_1} = \widehat{V_2}$. The equality $\widehat{V_1} = \widehat{V_2}$ can be established by the following argument.

We have $|V_1| = |V_2|$ as a consequence of the fact that the proper image of some word V has length greater than or equal to the length of the V. This implies that every generator in V_1 is sent to a generator in V_2, and vice versa, and therefore the number of generators that appear in V_1 is equal to the number of generators that appear in V_2. Thus if ϕ is the endomorphism that sends V_1 to V_2, then ϕ produces a bijective correspondence (compatible with the inverse map) between the generators that appear in V_1 and the generators that appear in V_2. This gives $\widehat{V_1} = \widehat{V_2}$.

It is fairly easy to show that the set C_W is a subset of D_W. The proof of Proposition 2 is more involved and can be found in [4], Proposition 2.2. Notice, however, that one of the implications in the proposition is easy to establish. If U is a proper cyclic image of a word V in C_W, then clearly $[U]$ is an endomorphic image of $[W]$, since $[V]$ is obtained from $[W]$ by a series of endomorphisms.

5. Properties of the set C_W

5.1. **Description of the set C_W as a tree.** The set C can be viewed as a tree with nodes representing the words (from now on 'word' and 'node' will be used interchangeably while describing the tree) in the set C and the edges representing the maps that connect the words. The root of the tree is the word W and the grandchildren of every node U form the set L_U. Let T_C be the tree that corresponds to the set C, as in Figure 1.

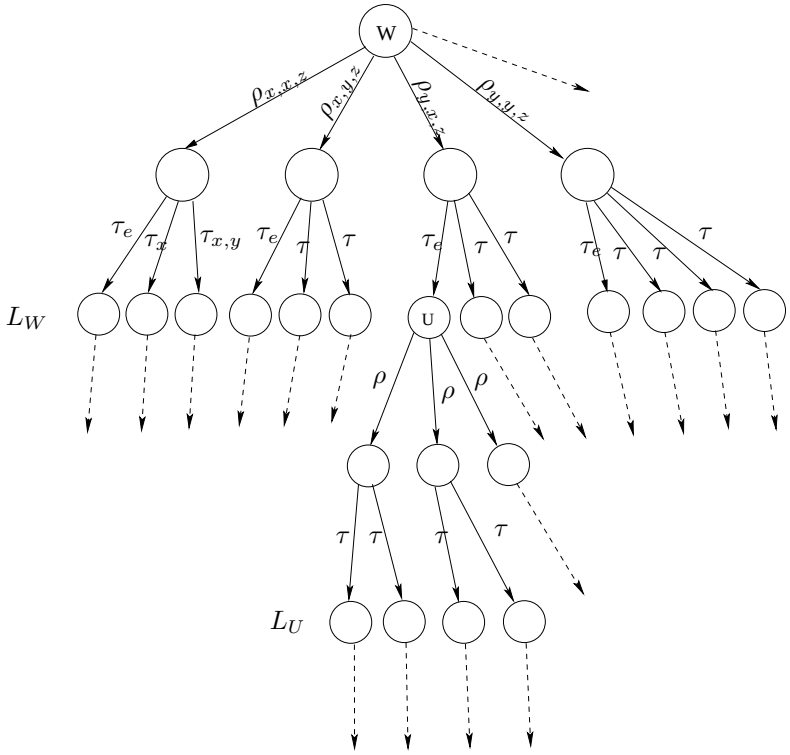

FIGURE 1. The tree T_C

The order in which the nodes of the tree are drawn corresponds to the order in which the words in C are generated by the algorithm. We start with node W and its children and grandchildren. Let U be the node with the shortest length among the children and grandchildren of W. We draw those children and grandchildren of U which are not proper images of words already in the tree. We continue to search for the shortest node in the tree for which the set L has not been computed yet, we compute the set L and draw the corresponding children and grandchildren in the tree.

While the tree picture can be very helpful intuitively, it is not entirely accurate because every time a word W' is found to be a proper cyclic image of a word in C the node W' and possibly other nodes in the subtree rooted at W' will be removed from T_C. Also, in practice we compute K_W before computing L_W, but since in the two-generator case K_W only contains one-generator words and W itself, we choose not to include K_W in the picture of T_C.

5.2. **Number of generators in C_W.** Results from Edmunds [2] can help us determine what words might belong to the set C_W. The set C_W is a subset of the set D_W defined by Edmunds in [2] and since all the results below are valid for D_W, they are also valid for C_W. For example, only one word of the type a^n, that is, a single word in one generator, needs to be added to C_W, as the next lemma shows.

Lemma 1 (Edmunds [2])**.** *Let x be a generator. A word W has x^n as an endomorphic image if and only if n is a multiple of* $\gcd(W)$*, where* $\gcd(W) = \gcd\{|\sigma_i(W)| : x_i$ *is generator in* $W\}$*, and $\sigma_i(W)$ is the number of occurences of x_i minus the number of occurences of x_i^{-1} in W.*

In [2], for any word W, Edmunds gives bounds on the number of generators that occur in the words in D_W. For example, if W is a two-generator word, then the set D_W does not contain words on four or more generators. This is a consequence of the Theorem 3.1 in [2]. Since the set C_W is a subset of D_W, the set C_W will not contain words on more than three generators, if W is a two-generator word.

6. The approach for a two-generator word W

In this section we provide a local picture of the set C_W in the case when W is a two-generator word. We analyze the endomorphisms that produce the set C_W and we show that at a local level some of them generate proper images of already existing words.

6.1. **Description of the set L_U, for a two-generator word U.** Let us recall that the set L_U is the set of irredundant words $\widehat{V'}$ that result from all the possible chains of transformations

$$U \xrightarrow{\rho_{x,y,z}} \xrightarrow[reduction]{cyclic} V \xrightarrow{\tau_A} \xrightarrow[reduction]{cyclic} V' \xrightarrow{\widehat{\ }} \widehat{V'} \tag{2}$$

applied to U, where x is not necessarily different from y and A is a set of generators that occur in V. From now on, every time we apply replacement or annihilation endomorphisms, we also perform cyclic reduction. Therefore we will not specifically mention the cyclic reduction in our exposition, and will call the composition of a replacement endomorphism and cyclic reduction simply a *replacement map* or just *replacement*. Similarly, the composition of an annihilation endomorphism and cyclic reduction will be called an *annihilation map*. We keep the same notation, $\rho_{x,y,z}$, for a replacement map, and τ_A for an annihilation map.

If U is a word in a and b, then each replacement $\rho_{x,y,z}$ can be followed by eight possible annihilation maps since $\rho_{x,y,z}(U)$ is a word in up to three generators. However, the only interesting annihilation maps are τ_e, τ_x and τ_y, as shown below.

$U \xrightarrow{\rho_{x,y,z}} V \xrightarrow{\tau_e} W_1 \xrightarrow{\widehat{\ }} \widehat{W_1}$
$U \xrightarrow{\rho_{x,y,z}} V \xrightarrow{\tau_x} W_2 \xrightarrow{\widehat{\ }} \widehat{W_2}$
$U \xrightarrow{\rho_{x,y,z}} V \xrightarrow{\tau_y} W_3 \xrightarrow{\widehat{\ }} \widehat{W_3}$

The other annihilation maps produce U, the identity element e or one-generator words.

Assuming $|U| \geq 2$, the word U must contain some of the syllables aa, ab, ab^{-1}, $a^{-1}b$, ba, bb (or their inverses). Thus in order to describe L_U we need to analyze the effect of each of six possible replacements on the word U. Let us note that $\rho_{x,y,z}(U) = \rho_{y^{-1},x^{-1},z}(U)$ and that $\rho_{x,y,z}(U)$ also is a proper image of $\rho_{x,y,z^{-1}}(U)$ for any letters $x, y \in \{a^{\pm 1}, b^{\pm 1}\}$ and $z = c^{\pm 1}$, so we only need to consider $\rho_{x,y,c}(U)$.

Lemma 2. *Let U be a two-generator word in a and b.*

i) *If U contains both syllables aa and bb, then $V_1 = \rho_{a,a,c}(U)$ and $V_2 = \rho_{b,b,c}(U)$ represent the same cyclic words up to monoid automorphisms.*
ii) *If U contains syllable aa but no bb, then $V_1 = \rho_{a,a,c}(U)$ is a proper cyclic image of U.*

Proof. (i) The word V_1 differs from U only in that in V_1 there is a letter c between every a - or a^{-1} - segment and every b - or b^{-1} - segment and there is a letter c^{-1} between every b - or b^{-1} - segment and a - or a^{-1} - segment. The word V_2 differs from U only in that in V_2 there is a letter c^{-1} between

every a - or a^{-1} - segment and every b - or b^{-1} - segment and there is a letter c between every b - or b^{-1} - segment and a - or a^{-1} - segment.

(ii) Let V_1 be as in part (i), $V_1 = \rho_{a,a,c}(U)$. The word V_1 differs from U only in that there is one letter c between every a - segment and every b or b^{-1} and there is a c^{-1} between every b or b^{-1} and every a - segment. Thus $V_1 = \phi(U)$, where $\phi(a) = a$ and $\phi(b) = cbc^{-1}$. □

Thus when applying replacements $\rho_{a,a,c}$ and $\rho_{b,b,c}$ to U we get at most one word that is not a proper cyclic image of any other word in L_U. In order to completely describe the set L_U we need to consider not only the replacements, but also the corresponding annihilation maps.

For example, applying the replacement $\rho_{a,a,c}$ followed by annihilation maps produces a three-generator word and two one-generator words, as shown below.

$U \xrightarrow{\rho_{a,a,c}} V \xrightarrow{\tau_e} V' \xrightarrow{\widehat{\ }} \widehat{V'}$ (three-generator word)

$U \xrightarrow{\rho_{a,a,c}} V \xrightarrow{\tau_a} b^p$ (one-generator word)

$U \xrightarrow{\rho_{a,a,c}} V \xrightarrow{\tau_b} a^p$ (one-generator word)

The word V differs from U only in that every a-segment or a^{-1}-segment is precedeed by c^{-1} and followed by c. Thus when we annihilate generator a all the c's will cancel out and we are left with a word in b, and when we annihilate b all the c's will cancel out and we are left with a word in a.

Similarly, the replacement $\rho_{b,b,c}$ followed by annihilation maps produces a three-generator word and two one-generator words. The one-generator words that belong to the set C_W are completely described by Lemma 1, so although the one-generator words produced by the replacements $\rho_{a,a,c}$ and $\rho_{b,b,c}$ followed by annihilation maps might be added to the set C, they will eventually be removed except for the final one. Among the words produced in Lemma 2 (i) only one of V_1 and V_2 will potentially be added to the set C.

Before we analyze the other replacements, we notice that every word that contains syllable ab will also contain syllable ba, and moreover, they have the same number of occurences. The proof of Lemma 3 is left to the reader.

Lemma 3. *In a two-generator cyclic word V, $(xy)_V = (yx)_V$, for all letters x and y with $x \neq y$, $x \neq y^{-1}$.*

Lemma 4. *Let U be a two-generator word in a and b. The word $\rho_{x,y,c}(U)$ is the same cyclic word as $\rho_{y,x,c}(U)$ up to monoid automorphisms, for all letters x and y in the set $\{a^{\pm 1}, b^{\pm 1}\}$ with $x \neq y$, $x \neq y^{-1}$.*

Proof. Let $V = \rho_{x,y,c}(U)$ and $V' = \rho_{y,x,c}(U)$. Our goal is to show that V and V' are the same cyclic word up to monoid automorphisms.

The effect of $\rho_{x,y,c}$ on U is to take syllable xy to xy, xx to $xcxc$, yy to $c^{-1}yc^{-1}y$, xy^{-1} to $xcy^{-1}c$, yx to $c^{-1}yxc$, $x^{-1}y$ to $c^{-1}x^{-1}c^{-1}y$. Now consider $\rho_{y,x,c^{-1}}$ instead of $\rho_{y,x,c}$. The effect of $\rho_{y,x,c^{-1}}$ on U is to take syllable xy to $cxyc^{-1}$, xx to $cxcx$, yy to $yc^{-1}yc^{-1}$, xy^{-1} to $xcy^{-1}c$, yx to yx, $x^{-1}y$ to $x^{-1}c^{-1}yc^{-1}$.

Thus each of the two replacements, $\rho_{x,y,c}$ and $\rho_{y,x,c^{-1}}$, inserts c, c^{-1} or the empty word between the two letters of a syllable. Since both maps insert the same element in every syllable of U as seen in the above paragraph and do not modify the word U in any other way, we conclude that V and $\rho_{y,x,c^{-1}}(U)$ are the same cyclic word. As $\rho_{y,x,c}(U)$ and $\rho_{y,x,c^{-1}}(U)$ are the same word, up to the monoid automorphism that sends c to c^{-1}, we get that V and V' are the same cyclic word up to monoid automorphism. □

Thus we only need to consider one of the replacements $\rho_{a,b,c}$ and $\rho_{b,a,c}$ when constructing L_U and the words that will be added to L_U are $\widehat{W_1}, \widehat{W_2}$ and $\widehat{W_3}$.

$U \xrightarrow{\rho_{a,b,c}} V \xrightarrow{\tau_e} W_1 \xrightarrow{\widehat{}} \widehat{W_1}$

$U \xrightarrow{\rho_{a,b,c}} V \xrightarrow{\tau_a} W_2 \xrightarrow{\widehat{}} \widehat{W_2}$

$U \xrightarrow{\rho_{a,b,c}} V \xrightarrow{\tau_b} W_3 \xrightarrow{\widehat{}} \widehat{W_3}$

Similarly, we only need to consider one of the replacements $\rho_{a,b^{-1},c}$ and $\rho_{a^{-1},b,c}$ when constructing L_U and the words that will be added to L_U are $\widehat{V_1}, \widehat{V_2}$ and $\widehat{V_3}$ as below.

$U \xrightarrow{\rho_{a,b^{-1},c}} V \xrightarrow{\tau_e} V_1 \xrightarrow{\widehat{}} \widehat{V_1}$

$U \xrightarrow{\rho_{a,b^{-1},c}} V \xrightarrow{\tau_a} V_2 \xrightarrow{\widehat{}} \widehat{V_2}$

$U \xrightarrow{\rho_{a,b^{-1},c}} V \xrightarrow{\tau_b} V_3 \xrightarrow{\widehat{}} \widehat{V_3}$

As a consequence of the previous lemma we get the following:

Proposition 3. *For each two-generator word U we need only apply at most three replacement endomorphisms (or maps) in order to obtain all the possible three-generator words in L_U.*

Proof. Recall that for a two-generator word U in a and b there are six possible replacements to consider: $\rho_{a,a,c}$, $\rho_{b,b,c}$, $\rho_{a,b,c}$, $\rho_{b,a,c}$, $\rho_{a,b^{-1},c}$, $\rho_{a^{-1},b,c}$.

By Lemmas 2 and 4 and we see that out of these six possible replacements, $\rho_{a,a,c}$ and $\rho_{b,b,c}$ generate the same word up to monoid automorphisms, $\rho_{a,b,c}$ and $\rho_{b,a,c}$ generate the same word up to monoid automorphisms, $\rho_{a,b^{-1},c}$ and $\rho_{a^{-1},b,c}$ generate the same word up to monoid automorphisms, so we can choose one endomorphism (or map) out of each above pair in order to obtain all possible three-generator words in L_U. □

6.2. **The set C'_W for a two-generator word W.** We continue in the spirit of the previous subsections, where we showed that at a local level some of the transformations in C_W generate proper images of already existing words. It turns out that applying the same replacements consecutively also becomes redundant and we can further reduce the number of such maps needed to generate C_W. We therefore consider only the relevant transformations and in order to eliminate the 'randomness' factor brought by the $\widehat{\ }$ operation, we replace it by equivalent transformations and we call the resulting set C'_W. Before going deeper into the behavior of the replacement maps and the definition of C'_W, we need to make the following important observation.

Observation. Let U be a three-generator word in the set C_W. Applying a replacement ρ to U produces a redundant word (follows from Theorem 3.1, [2]), which is the reason why there are no words on more than three generators in C_W. Also, when we apply τ_A to $\rho(U)$, for any set A of generators, it is easy to check that we obtain a proper image of some word in the set C. Therefore we will concentrate on the effect of the replacements on the two-generator words. We will explain later how the three-generator words fit into the big picture.

The following discussion becomes much clearer if we remove the $\widehat{\ }$ operation from consideration. The two-generator words in L_U are words in either a and c or b and c, so in order to have two-generator words on only a and b, we will send c to a and the other generator in the word to b. Let ϕ be the transformation described above, that is,

$$\phi : \begin{smallmatrix} c\to a \\ a\to b \end{smallmatrix} \quad \text{or} \quad \phi : \begin{smallmatrix} c\to a \\ b\to b \end{smallmatrix} \tag{3}$$

We keep all the other transformations, that is, the replacement and annihilation maps, the same. Thus for a word U on generators a and b construct the set L'_U as the set of irredundant words V'' that result from all the possible chains of transformations

$$U \xrightarrow{\rho_{x,y,z}} V \xrightarrow{\tau_t} V' \xrightarrow{\begin{smallmatrix} c\to a \\ a\to b \end{smallmatrix}} V'', \text{or} \tag{4}$$

$$U \xrightarrow{\rho_{x,y,z}} V \xrightarrow{\tau_t} V' \xrightarrow{\begin{smallmatrix} c\to a \\ b\to b \end{smallmatrix}} V''. \tag{5}$$

where $x, y \in \{a^{\pm 1}, b^{\pm 1}\}$, $z = c$ and $t = a, b$. Let C' be the set of words created in the same manner as the set C in section 2.3 but with the operation $\widehat{\ }$ replaced by the transformation ϕ in 3. The main difference between the sets C and C' lies in the way the sets K_V and K'_V are generated, as seen in Table 1.

The procedure to obtain the set C'_W follows the same rules as the one for obtaining C_W. It starts by placing K'_W in a set Q'. Let C' be the set of words V in Q' such that V is not a proper cyclic image of some element

K'_V	K_V
K'_V is the set of irredundant words P obtained by choosing a subset A of the generators in V, letting Z be $\tau_A(V)$, and setting $P = \phi(Z)$, where ϕ sends c to a, as in (3).	K_V is the set of irredundant words P obtained by choosing a subset A of the generators in V, letting Z be $\tau_A(V)$, and setting $P = \widehat{Z}$.
L'_U	L_U
L'_U is the union of the sets K'_V, where $V = \rho_{x,y,z}(U)$ for any admissible $\rho_{x,y,z}$.	L_U is the union of the sets K_V, where $V = \rho_{x,y,z}(U)$ for any admissible $\rho_{x,y,z}$.

TABLE 1. Comparison between C and C'

of $Q' - \{V\}$. (If in Q' there are two words, V_1 and V_2, such that each is a proper image of the other, we want to make sure we do not remove both. Therefore we will keep the one word that is first in its orbit in $\mathrm{Aut}(M)$.) Assign Q' to C'. As long as the set Q' does not become empty, execute commands (1) & (2):

(1) Choose an element $V \in Q'$ of minimal length and delete it from Q';

(2) For all U in L'_V, check whether U is a proper cyclic image of some element of C'. If it is, do not add it to C'. Otherwise delete from C' and Q' all words that are proper cyclic images of U and add U to both C' and Q'.

As before, C'_W is the set of elements added to C' during the above procedure and never removed.

We show in Theorem 3 that the set C' we obtained above is in a natural bijective correspondence with the set C. The bijection sends every word U in C to a word U' in C' such that $U = \widehat{U'}$.

Theorem 3. *Let W be a two-generator word, and let C and C' be the sets generated by the algorithms that produce the sets C_W and C'_W, respectively. There is a bijection f between C and C' such that for any word U in C, the word $f(U)$ has the property $\widehat{f(U)} = U$.*

Proof. We prove the theorem by explicitly constructing a bijection f from the set C to the set C'. Assume that $\widehat{W} = W$. It is easy to see that K_W and K'_W are in a bijective correspondence. If W is irredundant, then $\widehat{W} = W$ is in K_W, W is in K'_W and set $f(\widehat{W}) = W$. Since W has only two generators, the set K_W has at most four words: W, $\widehat{\tau_a(W)}$, $\widehat{\tau_b(W)}$ and the empty word. The same is true for K'_W, and we can set $f(\widehat{\tau_a(W)}) = \tau_a(W)$, $f(\widehat{\tau_b(W)}) = \tau_b(W)$, and $f(e) = e$.

To establish the bijection between the sets C and C' we need the following lemma, which shows that there exists a bijective correspondence between the sets L and L'.

Lemma 5. *Let U and U' be two-generator words in a and b such that $U = \widehat{U'}$. Then there exists a bijective map h between the sets L_U and $L'_{U'}$ such that $V = \widehat{h(V)}$ for all V in L_U.*

Proof. Let ψ be an automorphism that sends U to U'. Since we consider two-generator words in a and b there are eight such possible automorphisms, as far as actions on a and b. One can associate to each replacement endomorphism $\rho_{x,y,z}$ applied to U replacement endomorphism $\rho_{\psi(x),\psi(y),z}$ applied to U', where $x, y \in \{a, a^{-1}, b, b^{-1}\}$. Then $\widehat{\rho_{x,y,z}(U)} = \widehat{\rho_{\psi(x),\psi(y),z}(U')}$ and we let $h(\widehat{\rho_{x,y,z}(U)}) = \rho_{\psi(x),\psi(y),z}(U')$. For example, if $\psi(a) = a$ and $\psi(b) = b^{-1}$ then $\widehat{\rho_{a,b,c}(U)} = \widehat{\rho_{a,b^{-1},c}(U')}$. There is a similar correspondence between the annihilation endomorphisms applied in L_U and $L'_{U'}$. We associate to $\tau_t(\widehat{\rho_{x,y,z}(U)})$ in L_U the word $\tau_{\psi(t)}(\rho_{\psi(x),\psi(y),z}(U'))$ in $L_{U'}$, where $t \in \{a, b\}$. □

To conclude the proof of Theorem 3, we show that if a word U is the proper cyclic image of a word V then $f(U)$ is the proper cyclic image of $f(V)$. We use the fact that $\widehat{f(U)} = U$ and $\widehat{f(V)} = V$. The word $f(U)$ is a proper cyclic image of U since $\widehat{\ }$ interchanges the letters in $f(U)$, U is a proper cyclic image of V by assumption, and V is a proper cyclic of $f(V)$ because of the $\widehat{\ }$ operation. Thus $f(U)$ is a proper cyclic image of $f(V)$.

Therefore the restriction of the map f to the set C is a bijection between C and C'. □

We then want to keep track of all the possible two-generator words in L'_U, and the transformations that generate them. These transformations result from composing a replacement and an annihilation map, followed by a permutation on the generators. The permutation is given by ϕ as defined in (3).

We get the following four transformations.

(i) The first transformation comes from $\tau_a \circ \rho_{a,b,c} = {}^{a\to c}_{b\to c^{-1}b}$. We then apply $\phi = {}^{c\to a}_{b\to b}$, and we get $\theta_{a,b,a} : {}^{a\to a}_{b\to a^{-1}b}$

(ii) The second transformation comes from $\tau_b \circ \rho_{a,b,c} = {}^{a\to ac}_{b\to c^{-1}}$. We then apply $\phi = {}^{c\to a}_{a\to b}$ and we get $\theta_{a,b,b} : {}^{a\to ba}_{b\to a^{-1}}$

(iii) The third transformation comes from $\tau_a \circ \rho_{a,b^{-1},c} = {}^{a\to c}_{b\to bc}$. We then apply $\phi = {}^{c\to a}_{b\to b}$ and we get $\theta_{a,b^{-1},a} : {}^{a\to a}_{b\to ba}$

(iv) The fourth transformation comes from $\tau_b \circ \rho_{a,b^{-1},c} = {}^{a\to ac}_{b\to c}$. We then apply $\phi = {}^{c\to a}_{a\to b}$ and we get $\theta_{a,b^{-1},b} : {}^{a\to ba}_{b\to a}$

Notice that applying $\rho_{a,a,c}$ and τ_a or τ_b will generate a one-generator word, so we will not include it in the above list of transformations.

The following lemmas show that there is duplication when applying replacements to the two-generator words produced by the four above transformations. The next lemma shows that after transformation $\theta_{a,b,a}$, only transformations $\theta_{a,b,a}$ and $\theta_{a,b,b}$ can be applied without causing duplication.

Lemma 6. *For a two-generator word U in a and b, the word $\rho_{a,b^{-1},c}(\theta_{a,b,a}(U))$ is a proper cyclic image of $\rho_{a,b,c}(U)$.*

Proof. The transformation $\rho_{a^{-1},b,c^{-1}} \circ \theta_{a,b,a}$ is defined as $\substack{a\to ca \\ b\to a^{-1}b}$. If we let σ be the automorphism that interchanges a and c, then notice that $\sigma \circ \rho_{a,b,c}$ is also defined as $\substack{a\to ca \\ b\to a^{-1}b}$. So $\rho_{a^{-1},b,c^{-1}}(\theta_{a,b,a}(U)) = \sigma(\rho_{a,b,c}(U))$. By Lemma 4 $\rho_{a,b^{-1},c}$ and $\rho_{a^{-1},b,c^{\pm 1}}$ produce the same words up to monoid automorphisms, so $\rho_{a,b^{-1},c}(\theta_{a,b,a}(U))$ is a proper image of $\rho_{a,b,c}(U)$ via the automorphism σ. □

The next lemmas show that after transformations $\theta_{a,b,b}$, $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ only $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ can be applied without causing duplication.

Lemma 7. *For a two-generator word U in a and b, the word $\rho_{a,b,c}(\theta_{a,b,b}(U))$ is a proper cyclic image of $\rho_{a,b,c}(U)$.*

Proof. By Lemma 4, $\rho_{a,b,c}(U)$ is the same word, up to monoid automorphisms, as $\rho_{b,a,c}(U)$. The transformation $\rho_{b,a,c} \circ \theta_{a,b,b}$ is defined as $\substack{a\to ba \\ b\to a^{-1}c}$. If we let σ be the monoid automorphism that sends a to b, b to c and c to a then we get that $\sigma \circ \rho_{b,a,c} \circ \theta_{a,b,b}$ is defined as $\substack{a\to ac \\ b\to c^{-1}b}$, which is the same as $\rho_{a,b,c}$. Thus $\rho_{a,b,c}(\theta_{a,b,b}(U))$ is a proper cyclic image of $\rho_{a,b,c}(U)$. □

Lemma 8. *For a two-generator word U in a and b, the words $\rho_{a,b,c}(\theta_{a,b^{-1},a}(U))$ and $\rho_{a,b,c}(\theta_{a,b^{-1},b}(U))$ are proper cyclic images of $\rho_{a,b^{-1},c}(U)$.*

Proof. The transformation $\rho_{b,a,c^{-1}} \circ \theta_{a,b^{-1},a}$ is defined as $\substack{a\to ca \\ b\to ba}$. If we let σ be the monoid automorphism that interchanges a and c then $\sigma \circ \rho_{b,a,c^{-1}} \circ \theta_{a,b^{-1},a}$ is defined as $\substack{a\to ac \\ b\to bc}$, which is equivalent to $\rho_{a,b^{-1},c}(U)$. Thus $\rho_{a,b,c}(\theta_{a,b^{-1},a}(U))$ is a proper cyclic image of $\rho_{a,b^{-1},c}(U)$.

The transformation $\rho_{b,a,c^{-1}} \circ \theta_{a,b^{-1},b}$ is defined as $\substack{a\to ba \\ b\to ca}$. If we let ψ be the monoid automorpshism that sends a to c, c to b and b to a we get that $\psi \circ \rho_{b,a,c^{-1}} \circ \theta_{a,b^{-1},b}$ is defined as $\substack{a\to ac \\ b\to bc}$, which is the same as $\rho_{a,b^{-1},c}$. Thus $\rho_{a,b,c}(\theta_{a,b^{-1},b}(U))$ is a proper cyclic image of $\rho_{a,b^{-1},c}(U)$. □

6.3. **Tree representation of the two-generator words in C'.** Let W be a two-generator word in a and b. The set C_W contains three-generator words in $\{a^{\pm 1}, b^{\pm 1}, c^{\pm 1}\}$, two-generator words in $\{a^{\pm 1}, b^{\pm 1}\}$ and one one-generator

word (see section 4.1). As pointed out before, we are mostly interested in the two-generator words in C. In particular, we want to find the lengths of these words. For this purpose we will consider the set C' instead of C, because in the set C' all the transformations determine exactly where each generator goes, while in C the $\widehat{\ }$ operation varies for each word. We strip the set C' of all the three-generator and one-generator words and we get a tree composed only of two-generator words. Call this tree T_W. This tree is not a subtree of $T_{C'}$, since although the nodes are a subset of the nodes in $T_{C'}$, the edges corresponding to transformations between words are different. In T_W the edges represent the transformations $\theta_{a,b,a}$, $\theta_{a,b,b}$, $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ (drawn in this order from left to right for the nodes where the transformations are applied) that send two-generator words to two-generator words.

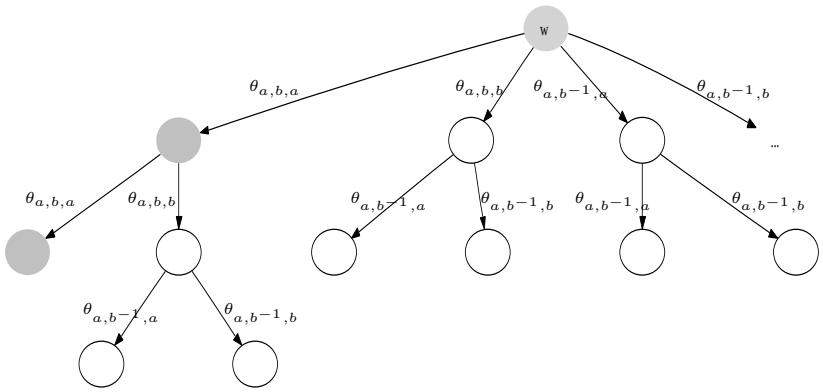

FIGURE 2. The tree T_W

The only possible transformations between the words are as in Figure 2. For example, the only transformations $\theta_{a,b,a}$ appear on the leftmost edges, and the only transformations $\theta_{a,b,b}$ appear on edges leaving nodes obtained from transformation $\theta_{a,b,a}$, since the replacement endomorphism $\rho_{a,b,c^{\pm 1}}$ (and therefore $\theta_{a,b,a}$ and $\theta_{a,b,b}$) is not needed anywhere else by Lemma 8.

7. Overview of T_W for a two-generator word W

In this section we describe the main steps involved in finding the complexity of the set C_W. Our approach relies on analyzing the structure of the tree T_W. In the next section we provide all the technical results needed to show the dynamics of word length in T_W. Recall that in order to find the complexity of the algorithm we need to find how many words have been

generated before finding the last one of a certain length n. We do this in two steps, first by approximating the number of words of length m in the tree T_W, and second, by finding bounds on the length of the words generated before finding the last word of length n.

As far as the first step is concerned, we show that the number of words of length m in T_W is linear in m. Since all the words in T_W are automorphic images of each other, we were guaranteed a quadratic upper bound by a conjecture of Shpilrain and Myasnikov [13] proved by B.Khan [9]. Their statement is the following.

Theorem 4 ([13], [9])**.** *Let u be a word in the free group F_2 on two generators. Assume that the length of u is irreducible by any automorphism of F_2, which in particular implies that u is cyclically reduced. Then the number of automorphic images of u that have the same length as u does, is bounded by the polynomial $8|u|^2 - 40|u|$, for large enough $|u|$.*

In order to be able to describe the tree T_W in greater detail, we analyze how the transformations $\theta_{a,b,a}$, $\theta_{a,b,b}$, $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ affect the length of the two-generator words. This tree contains only two-generator words. For any node m in the tree T_W we will call $(a)_m$ and $(b)_m$ the *coordinates* of m. We say that two nodes m and p have the same coordinates if $(a)_m = (a)_p$ and $(b)_m = (b)_p$. Note that the length of word m is $(a)_m + (b)_m$. We can predict changes in the coordinates based on the maps that connect the parent to the children nodes.

Lemma 9. *Let m be a node in the tree T_W and let m_1 and m_2 be its children. If m_1 and m_2 have been obtained from m by applying transformations $\theta_{a,b,a}$ and $\theta_{a,b,b}$, respectively, and $(ab)_m$ is the number of times syllable ab appears in m, then we get the following relations:*

$$(a)_{m_1} = (a)_m + (b)_m - 2(ab)_m, \quad (b)_{m_1} = (b)_m, \tag{6}$$

$$(a)_{m_2} = (a)_m + (b)_m - 2(ab)_m, \quad (b)_{m_2} = (a)_m. \tag{7}$$

Proof. Recall that transformation $\theta_{a,b,a}$ is given by $\begin{smallmatrix} a \to a \\ b \to a^{-1}b \end{smallmatrix}$. Thus the a's in m_1 come from both a's and b's in m, while the b's in m_1 can only come from b's in m. Every occurence of ab in m will produce a syllable aa^{-1} in $\theta_{a,b,a}(m)$ which gets cancelled when cyclically reducing $\theta_{a,b,a}(m)$ to form m_1.

Transformation $\theta_{a,b,b}$ is given by $\begin{smallmatrix} a \to ba \\ b \to a^{-1} \end{smallmatrix}$. Thus the a's in m_2 come from both a's and b's in m, while the b's in m_2 can only come from a's in m. Every occurence of ab in m will produce a syllable aa^{-1} in $\theta_{a,b,b}(m)$ which gets cancelled when cyclically reducing $\theta_{a,b,b}(m)$ to form m_2. □

Lemma 10. *Let m be a node in the tree T_W and let m_1 and m_2 be its children. If m_1 and m_2 have been obtained from m by applying transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$, respectively, and $(ab^{-1})_m$ is the number of times syllable ab^{-1} appears in m, then we get*

$$(a)_{m_1} = (a)_m + (b)_m - 2(ab^{-1})_m, \quad (b)_{m_1} = (b)_m, \tag{8}$$

$$(a)_{m_2} = (a)_m + (b)_m - 2(ab^{-1})_m, \quad (b)_{m_2} = (a)_m. \tag{9}$$

Proof. Analogous to previous lemma. □

The above relations show that in order to determine the length of words in C' we need to find out how the numbers of syllables ab and ab^{-1} change when we apply the four transformations. For example, it will be shown that when $\theta_{a,b,a}$ is applied repeatedly to some word U, the number of syllables ab in the images of U decreases until it reaches a minimum. If the minimum is equal to zero, then we don't apply $\theta_{a,b,a}$ anymore, so the existence of syllable ab is critical for the transformation $\theta_{a,b,a}$ applied to U. This is why we call syllable ab the *critical syllable* of word U for transformations $\theta_{a,b,a}$ and $\theta_{a,b,b}$. The syllable ab^{-1} is the *critical syllable* of some word V for transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$, if these transformations are applied to V.

For a two-generator cyclic word U in generators a, b and some transformation T let $\min_T(xy, U)$ be the minimum number of times syllable xy occurs in $T^n(U)$, where n is any positive integer and $x, y \in \{a^{\pm 1}, b^{\pm 1}\}$. That is,

$$\min_T(xy, U) = \min_{n \geq 1}((xy)_{T^n(U)})$$

If for a word V we have $(xy)_V = \min_T(xy, V)$, call V *critical* with respect to T and *critical* with respect to the syllable xy.

Example. *Let $U = a^2bab^{-1}ab^{-1}$ and let $T = \theta_{a,b^{-1},a}$. Then $\min_T(ab^{-1}, U) = 1$ since $T(U) = a^2bab^{-1}b^{-1}$ and one can check that $(ab^{-1})_{T^n(U)} = 1$ for any $n \geq 1$.*

Notice that once we have iterated a transformation T enough times for some word U and have obtained $\min_T(xy, U)$, where xy is some subword of U, then we can get a recursive relation for the coordinates of all further iterates of U. That is, for n large enough, we can accurately describe the growth of $|T^n(U)|$. It turns out that there exists a class of special subtrees in T_W which we call 'almost perfect trees' and we define below. We show that for all the nodes in such a subtree the number of critical syllables with respect to the corresponding transformations can be completely described. The advantage of knowing the number of critical syllables is that we can

then accurately predict the coordinates and lengths of the nodes in the subtree.

In general, let the binary subtree of T_W rooted at some node w be T_w. Let Lb_w (the left branch starting at w) be the infinite path starting at w on which each node is the left child of the previous node. As before, we use the convention that for every node the left child is produced by $\theta_{x,y,a}$ and the right child is produced by $\theta_{x,y,b}$, where $x = a$ and $y = b$ or $y = b^{-1}$. For example, Lb_W consists of the nodes obtained by applying $\theta^i_{a,b,a}$ to W, where $i \geq 0$.

Definition.

(1) We call T_w *perfect* if the number of critical syllables with respect to the corresponding transformations is the same in the entire subtree. That is, for any two nodes u and v in T_w, if xy is the critical syllable with respect to the transformations $\theta_{x,y,a}$ and $\theta_{x,y,b}$ applied at u and zt is the critical syllable with respect to the transformations $\theta_{z,t,a}$ and $\theta_{z,t,b}$ applied at v then we have $(xy)_u = (zt)_v$, where $\{xy, zt\} \subseteq \{ab, ab^{-1}\}$.

(2) We call T_w *almost perfect* if:

(a) for any two nodes u and v in Lb_w with xy as critical syllable we have $(xy)_u = (xy)_v$, where $xy = ab$ or $xy = ab^{-1}$ and

(b) for any two nodes u and v in $T_w \setminus \mathrm{Lb}_w$, $(ab^{-1})_u = (ab^{-1})_v$.

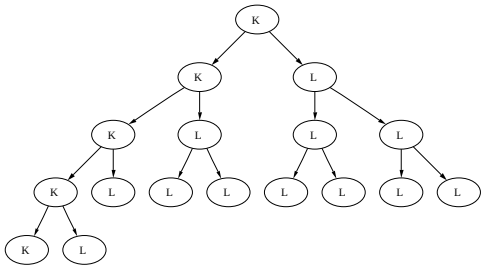

FIGURE 3. An almost perfect tree

For example, in the tree in Figure 3 let each node contain the number of critical syllables with respect to the transformations leaving that word. Then the tree in the figure is an almost perfect tree, and a perfect tree would have $K = L$.

The great advantage of an almost perfect tree is that in such a tree no two different nodes have the same coordinates (see Theorem 5). We will

show this by establishing a series of inequalities between the coordinates of nodes in an almost perfect tree. This property leads to the fact that in an almost perfect tree the number of nodes of length n is linear in n.

Finally, in Section 9, we show that the tree T_C is made up of a finite number of maximal almost perfect trees and a finite number of additional nodes, and we give bounds on these numbers in terms of the length of W (see Theorem 6). This helps us characterize the number of words of a certain length in the set C_W, thus leading to the final goal of finding the complexity of the endomorphism problem algorithm for two-generator words. We achieve this goal by analyzing the set C'_W, and then translating the results into statements about the set C_W.

8. Technical Lemmas

In this section we provide the results that will help determine the structure of the tree T_W. We establish these results by analyzing the words and transformations in T_W. One can view the four θ transformations as automorphisms in the free groups on two generators. As such, we can easily compute their inverses: $\theta^{-1}_{a,b,a} : \substack{a\to a\\ b\to ab}$, $\theta^{-1}_{a,b,b} : \substack{a\to b^{-1}\\ b\to ab}$, $\theta^{-1}_{a,b^{-1},a} : \substack{a\to a\\ b\to ba^{-1}}$, $\theta^{-1}_{a,b^{-1},b} : \substack{a\to b\\ b\to ab^{-1}}$.

Throughout this section we will be interested in finding out the origin of a subword via a specific automorphism. We will often use the following lemma.

Lemma 11. *Let U be a reduced two-generator word of the form $U = U_1U'U_2$, where U_1, U_2 and U' are non-empty subwords. Let ϕ^{-1} be one of the four above automorphisms, that is, $\phi = \theta_{a,b,a}$, $\theta_{a,b,a}$, $\theta_{a,b^{-1},a}$ or $\theta_{a,b^{-1},b}$, and let V be the reduced word equal to $\phi^{-1}(U)$. Then $\phi^{-1}(U')$ appears in V entirely, except perhaps for the first or last letter in $\phi^{-1}(U')$.*

Proof. We will prove the stronger result that in $\phi^{-1}(U)$ the length of the maximal subwords that get cancelled when forming V is at most 1. That is, we cannot find words of the type $xyy^{-1}x^{-1}$ inside $\phi^{-1}(U)$, where x and y are letters with $x \neq y^{-1}$.

Assume that $xyy^{-1}x^{-1}$ does occur. It is easy to see that we cannot have $x = y$. Since U is a reduced word, y and y^{-1} must be part of the images of different, non-inverse, letters in U. Therefore, without lack of generality, we can assume that $xy = \phi^{-1}(b)$. That implies $y^{-1} = \phi^{-1}(a)$ or $y^{-1} = \phi^{-1}(a^{-1})$, which means x^{-1} is the beginning of the image of some letter in the group under ϕ^{-1}. By a quick inspection, we notice that cannot be the case. □

The following two lemmas give characterizations of $\min_T(xy, U)$, where $T = \theta_{a,b,a}$, $\theta_{a,b^{-1},a}$ or $\theta_{a,b^{-1},b}$, and xy is the critical syllable for T.

Lemma 12. *For a two-generator cyclic word U in a and b,*

$$\min_{\theta_{a,x,a}}(ax, U) = \sum_{n\geq 1}(x^{-1}a^n x)_U,$$

where $x = b$ or $x = b^{-1}$. This minimum occurs in $\theta^q_{a,x,a}(U)$, where q is a positive integer such that no subwords of the type $(xa^m x)^{\pm 1}$ are present in $\theta^q_{a,x,a}(U)$, for any positive integer m.

Proof. Recall that transformation $\theta_{a,x,a}$ is $\underset{x\to a^{-1}x}{a\to a}$, when $x = b$ or $x = b^{-1}$. Notice that the number of ax's in U cannot increase when we apply $\theta_{a,x,a}$. This happens because any syllable ax in $\theta_{a,x,a}(U)$ can only come from a subword $a^k x$ in U, where $k \geq 2$. We have to analyze the subwords of U that contain $(ax)^{\pm 1}$. Clearly

$$(ax)_U = \sum_{n\geq 1}(x^{-1}a^n x)_U + \sum_{m\geq 1}(xa^m x)_U.$$

Since $\theta^l_{a,x,a}(x^{-1}a^n x) = x^{-1}a^n x$ and $\theta^l_{a,x,a}(xa^m x) = xa^{m-l}x$, for all positive integers m, n, and l, we get that if $q = \max_{m\geq 1}\{(xa^m x)_U \neq 0\}$, then

$$\sum_{m\geq 1}(xa^m x)_{\theta^q_{a,x,a}(U)} = 0 \text{ and } (ax)_{\theta^q_{a,x,a}(U)} = \sum_{n\geq 1}(x^{-1}a^n x)_U. \qquad \square$$

A consequence of Lemma 12 is that if a cyclic word V satisfies the relation $(ax)_V = \min_{\theta_{a,x,a}}(ax, V)$, then V is of the form $a^{n_1}x^{m_1}\dots a^{n_k}x^{m_k}$, where m_i and n_i are nonzero integers, and m_{i-1}, n_i, m_i cannot all three have the same sign, for $i = 1, \dots, k$ and $m_0 = m_k$.

The corresponding result for $\theta_{a,b^{-1},b}$ is more complicated, since the transformation $\theta_{a,b^{-1},b}$ produces more mixing of the two letters a and b.

Lemma 13. *For a two-generator cyclic word U, $\min_{\theta_{a,b^{-1},b}}(ab^{-1}, U)$ is reached in $\theta^k_{a,b^{-1},b}(U)$, where k is such that no subwords of the form $(b^{-1}ab^{-1})^{\pm 1}$, $(a^{-1}ba^{-1})^{\pm 1}$, or $(a^{-1}b^2a^{-1})^{\pm 1}$ appear in $\theta^k_{a,b^{-1},b}(U)$.*

Proof. Recall that transformation $\theta_{a,b^{-1},b}$ is $\underset{b\to a}{a\to ba}$. If we think of the word U as a sequence of positive and negative subwords (a negative subword consists of a^{-1}'s and b^{-1}'s, while a positive subword consists of a's and b's only), then the number of transitions from a positive subword to a negative subword is equal to the number of $(ab^{-1})^{\pm 1}$. Thus the number of $(ab^{-1})^{\pm 1}$ decreases only when a positive or negative subword disappears entirely. We will determine which positive subwords have the potential to vanish under transformation $\theta_{a,b^{-1},b}$. The same argument is valid for negative subwords.

Let p be a positive subword in U. As noted before, the transition from p to a negative segment is made by a subword of the type $(ab^{-1})^{\pm 1}$. When we apply $\theta_{a,b^{-1},b}$, we get $|\theta_{a,b^{-1},b}(p)| = 2(a)_p + (b)_p - 1$ since there is only

one cancellation due to $(ab^{-1})^{\pm 1}$. In particular we have

$$(a)_{\theta_{a,b^{-1},b}(p)} = (a)_p + (b)_p - 1,\ (b)_{\theta_{a,b^{-1},b}(p)} = (a)_p. \tag{10}$$

Thus $|\theta_{a,b^{-1},b}(p)| \geq |p|$. So if $(a)_p > 1$ then the length of the positive subword increases. We need to analyze on the cases $(a)_p = 0$ and $(a)_p = 1$. If $(a)_p = 0$ then $p = b^n$ for some positive integer n. We have $\theta_{a,b^{-1},b}(a^{-1}b^n a^{-1}) = b^{-1}a^{n-1}b^{-1}$, so if $n > 2$, by (10) the transformation $\theta_{a,b^{-1},b}$ will increase the length of the a-segment in $b^{-1}a^{n-1}b^{-1}$. Thus only $p = b$ and $p = b^2$ can vanish under $\theta_{a,b^{-1},b}$.

If $(a)_p = 1$ and $(b)_p = m$ for some positive integer m, we get $(a)_{\theta_{a,b^{-1},b}(p)} = (b)_p$ and $(b)_{\theta_{a,b^{-1},b}(p)} = 1$. Thus if $n \geq 1$, by (10) we have $|\theta_{a,b^{-1},b}(p)| \geq |p|$ and therefore only $p = a$ can vanish under $\theta_{a,b^{-1},b}$.

Now take into account the letters that can preceed and follow a positive subword. We get that the number of ab^{-1}'s in U can decrease when applying $\theta_{a,b^{-1},b}$ only if $(a^{-1}ba^{-1})^{\pm 1}$, $(a^{-1}b^2a^{-1})^{\pm 1}$ or $(b^{-1}ab^{-1})^{\pm 1}$ are subwords of U. □

We do not need to analyze $\min_{\theta_{a,b,b}}(ab, U)$ since transformation $\theta_{a,b,b}$ is never applied twice consecutively, as can be most clearly seen from Figure 2.

The following lemmas provide examples of nodes in T_W which are the roots of almost perfect trees. We start by displaying candidates for perfect trees.

Lemma 14. *Let w be a node in $T_W \setminus \mathrm{Lb}_W$ such that w contains no $(b^{-1}a^n b^{-1})^{\pm 1}$ or $(a^{-1}b^n a^{-1})^{\pm 1}$, where n is any positive integer. Then the tree T_w rooted at w is a perfect tree.*

Proof. We show that if for a node u in $T_W \setminus \mathrm{Lb}_W$ we have

$$(b^{-1}a^n b^{-1})_u = (a^{-1}b^n a^{-1})_u = 0 \tag{11}$$

for any positive integer n, then its children have the same property. That is, if $u_1 = \theta_{a,b^{-1},a}(u)$ and $u_2 = \theta_{a,b^{-1},b}(u)$ are u's children, then

$$(b^{-1}a^n b^{-1})_{u_i} = (a^{-1}b^n a^{-1})_{u_i} = 0, \text{where } i = 1, 2. \tag{12}$$

When obtaining u_1, the map $\theta_{a,b^{-1},a}$ can produce $b^{-1}a^n b^{-1}$ only from $b^{-1}a^{n+1}b^{-1}$ since $\theta^{-1}_{a,b^{-1},a}(b^{-1}a^n b^{-1}) = ab^{-1}a^{n+1}b^{-1}$, and $a^{-1}b^n a^{-1}$ from $a^{-1}(ba^{-1})^n$ since $\theta^{-1}_{a,b^{-1},a}(a^{-1}b^n a^{-1}) = a^{-1}(ba^{-1})^n a^{-1}$, and these subwords are not allowed in u. When obtaining u_2, the map $\theta_{a,b^{-1},b}$ can produce $b^{-1}a^n b^{-1}$ only from $a^{-1}b^{n+1}a^{-1}$ since $\theta^{-1}_{a,b^{-1},b}(b^{-1}a^n b^{-1}) = ba^{-1}b^{n+1}a^{-1}$, and $a^{-1}b^n a^{-1}$ from
$b^{-1}(ab^{-1})^n$ since $\theta^{-1}_{a,b^{-1},b}(a^{-1}b^n a^{-1}) = b^{-1}(ab^{-1})^n b^{-1}$, and these subwords are not allowed in u.

By Lemmas 12 and 13 for any node u with property (11) the number of critical syllables ab^{-1} in u cannot decrease when we apply transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$. This fact applied to the node w, together with the iteration of equalities (12), shows that the number of critical syllables throughout the tree T_w is constant. That is, T_w is a perfect tree. □

Lemma 15. *Let $w_{\theta_{a,b,a}}$ be the word on the leftmost branch of the tree T_W for which $\min_{\theta_{a,b,a}}(ab, W)$ has been reached. Let $w^* = \theta_{a,b,a}(w_{\theta_{a,b,a}})$. Then w^* contains no $b^{\pm m}$ with $m \geq 2$ and the tree T_{w^*} rooted at w^* is almost perfect.*

Proof. By hypothesis, the words $w_{a,b,a}$ and w^* are both critical with respect to ab, so the number of syllables ab in w^* cannot be decreased by applying $\theta_{a,b,a}$. Therefore the number of critical syllables ab in each node in Lb_{w^*} is constant and equal to $\min_{\theta_{a,b,a}}(ab, W)$. Let the nodes in Lb_{w^*} be $w_i, i \geq 0$, where $w^* = w_0$ and let the right child of each w_i be $u_i = \theta_{a,b,b}(w_i)$ (Figure 5).

We get $(b^m)_{w^*} = 0$ for $m \geq 2$ because $\theta^{-1}_{a,b,a}(b^m) = (ba)^m$, so the only way to obtain b^m in w^* is from subwords of the type bab in $w_{\theta_{a,b,a}}$. This is not possible since $w_{\theta_{a,b,a}}$ is a critical word with respect to $\theta_{a,b,a}$, and by Lemma 12 subword bab is not allowed in $w_{\theta_{a,b,a}}$.

Claim. For $i \geq 0$, we have $(b^{-1}a^n b^{-1})_{u_i} = (a^{-1}b^n a^{-1})_{u_i} = 0$, where n is any positive integer.

We have $\theta^{-1}_{a,b,b}(b^{-1}a^n b^{-1}) = b^{-1}a^{-1}b^{-(n+1)}a^{-1}$, and $\theta^{-1}_{a,b,b}(a^{-1}b^m a^{-1}) = b(ab)^m b$, but $(bab)_{w_i} = 0$ since all w_i are critical with respect to ab. Also $(b^{\pm m})_{w_i} = 0$, where $m \geq 2$ by the same argument as for w^*. Thus we have proved the claim.

By Lemma 14, our claim implies that the tree T_{u_i} rooted at u_i is perfect for any $i \geq 0$. One can check that $\theta_{a,b,b}(ab^{-1}a^{-1}) = ab^{-1}$, and there is no other way of obtaining ab^{-1} when applying $\theta_{a,b,b}$. Notice that the structure of any node in Lb_{w^*} is of the form $a^{n_1}b^{m_1}\dots a^{n_k}b^{m_k}$, where the m_i and n_i are integers, $m_i = \pm 1$, and m_{i-1}, n_i, m_i cannot all three have the same sign. Moreover, all nodes in Lb_{w^*} have the same k and the same m_i's. For any two nodes in Lb_{w^*} the signs of the n_i's in the two nodes are the same. Since the structure of w_l is the same as that of w_j for any l, j, we get $(ab^{-1})_{u_l} = (ab^{-1})_{u_j}$. This also shows that the tree T_{w^*} is almost perfect. □

Lemma 16. *Let w be a node in $T_W \setminus \mathrm{Lb}_W$ for which $\min_{\theta_{a,b^{-1},a}}(ab^{-1}, w) = (ab^{-1})_w$. Let $w^* = \theta_{a,b^{-1},a}(w)$. Then $(b^m)_{w^*} = 0$ for $m \geq 2$ and the tree T_{w^*} rooted at w^* is almost perfect.*

Proof. By hypothesis, the words w and w^* are critical with respect to ab^{-1}, so the number of syllables ab^{-1} in w^* cannot be decreased by applying $\theta_{a,b^{-1},a}$. Therefore the number of critical syllables ab^{-1} for words in Lb_{w^*} is constant and equal to $\min_{\theta_{a,b^{-1},a}}(ab^{-1}, w)$. Let the nodes in Lb_{w^*} be $w_i, i \geq 0$, where $w^* = w_0$ and let the right child of each w_i be u_i, as in Figure 4.

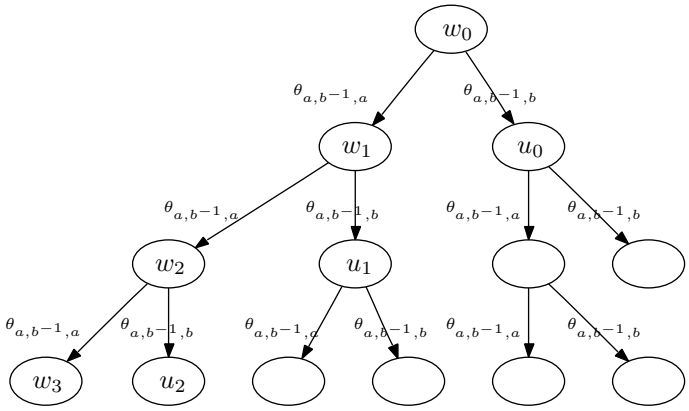

FIGURE 4. Tree T_{w^*}

We get $(b^m)_{w^*} = 0$ for $m \geq 2$ because the only way to obtain b^m in w^* is from subwords of the type $ba^{-1}b$ in w. This is not possible, since w is a critical word with respect to $\theta_{a,b^{-1},a}$, and by Lemma 12 the subword $ba^{-1}b$ is not allowed in w.

Claim. For $i \geq 1$, the word $\theta_{a,b^{-1},b}(w_i)$ does not contain subwords of the form $b^{-1}a^nb^{-1}$ or $a^{-1}b^na^{-1}$, where n is any positive integer.

We can only produce $b^{-1}a^nb^{-1}$ from $a^{-1}b^{n+1}a^{-1}$ since $\theta^{-1}_{a,b^{-1},b}(b^{-1}a^nb^{-1}) = ba^{-1}b^{n+1}a^{-1}$, and $a^{-1}b^na^{-1}$ from $b^{-1}(ab^{-1})^n$ since $\theta^{-1}_{a,b^{-1},b}(a^{-1}b^na^{-1}) = b^{-1}(ab^1)^nb^{-1}$, which are not allowed in Lb_w since subwords $b^{-1}ab^{-1}$ and b^m are not allowed in Lb_w for $m \geq 2$. Thus we have proved the claim.

By Lemma 14, our claim implies that the tree T_{u_i} rooted at u_i is perfect for any $i \geq 0$. Notice that the structure of any node in Lb_{w^*} is of the form $a^{n_1}b^{m_1}\dots a^{n_k}b^{m_k}$, where m_i and n_i are integers with all $|m_i| = 1$, $m_im_{i+1} = -1$, or m_{i-1}, n_i, m_i have the same sign. Moreover, all nodes in Lb_{w^*} have the same k and the same m_i's and the signs of the n_i's in any two nodes in Lb_{w^*} are the same. Since the structure of w_l is the same as that of w_j for any integers l, j, the number of critical syllables ab^{-1} in u_l is the same as in u_j. This also shows that the tree T_{w^*} is almost perfect. □

The following corollary will be useful for the proof of Proposition 5.

Corollary 1. *Let w be a node in $T_W \setminus \mathrm{Lb}_W$. If the tree T_w rooted at w is not an almost perfect tree then w must contain subwords of the type $(b^{-1}a^nb^{-1})^{\pm 1}$ or $(a^{-1}b^ma^{-1})^{\pm 1}$, where $n \geq 1$ and $m \geq 2$.*

Proof. The proof is similar to the proof of Lemma 16. More specifically, it is a stronger version of the proof that T_{w^*} is an almost perfect tree. Suppose w does not contain any subwords of the type $(b^{-1}a^nb^{-1})^{\pm 1}$ and $(a^{-1}b^ma^{-1})^{\pm 1}$, where $n \geq 1$ and $m \geq 2$. We show that T_w is then an almost perfect tree. Since w does not contain any $(b^{-1}a^nb^{-1})^{\pm 1}$ with n positive, it means that w is critical with respect to $\theta_{a,b^{-1},a}$. Therefore the number of critical syllables ab^{-1} for words in Lb_w is constant and equal to $\min_{\theta_{a,b^{-1},a}}(ab^{-1}, w)$.

Let the nodes in Lb_w be $w_i, i \geq 0$, where $w = w_0$ and let the right child of each w_i be u_i. As before, we claim that the words $\theta_{a,b^{-1},b}(w_i)$ cannot have subwords of the form $b^{-1}a^nb^{-1}$ or $a^{-1}b^na^{-1}$, where n is any positive integer. This happens because $\theta^{-1}_{a,b^{-1},b}(b^{-1}a^nb^{-1}) = ba^{-1}b^{n+1}a^{-1}$, and $\theta^{-1}_{a,b^{-1},b}(a^{-1}b^na^{-1}) = b^{-1}(ab^{-1})^nb^{-1}$, which are not allowed in Lb_w since subwords $b^{-1}ab^{-1}$ and $a^{-1}b^ma^{-1}$ are not allowed in Lb_w for $m \geq 2$. Thus we have proved the claim. By Lemma 14, our claim implies that the tree T_{u_i} rooted at u_i is perfect for any $i \geq 0$.

Notice that the structure of any node in Lb_w, except for w_0, is of the form $a^{n_1}b^{m_1}\ldots a^{n_k}b^{m_k}$, where m_i and n_i are integers with all $|m_i| = 1$, $m_im_{i+1} = -1$, or m_{i-1}, n_i, m_i have the same sign. Since the structure of w_l is the same as that of w_j for any integers $l, j > 0$, the number of critical syllables ab^{-1} in u_l is the same as in u_j. It remains to show that $(ab^{-1})_{u_0} = (ab^{-1})_{u_1}$. One can check that aba^{-1} and bba^{-1} are sent to ab^{-1} by $\theta_{a,b^{-1},b}$, and there is no other way of obtaining ab^{-1} when applying $\theta_{a,b^{-1},b}$. Then it is easy to see that $(aba^{-1})_{u_0} + (bba^{-1})_{u_0} = (aba^{-1})_{u_1} + (bba^{-1})_{u_1}$, by applying $\theta_{a,b^{-1},a}$ and counting the number of appearances of such subwords.

Thus the tree T_w is almost perfect, which proves the corollary. □

The two above lemmas can be combined to get the following proposition.

Proposition 4.
(i): *If a word w contains no $b^{\pm m}$, $m > 1$, and the number of critical syllables ab in w is equal to $\min_F(ab, w)$, where $F = \theta_{a,b,a}$, then the tree T_w rooted at w is almost perfect.*

(ii): *If a word w contains no $b^{\pm m}$, $m > 1$, and the number of critical syllables ab^{-1} in w is equal to $\min_F(ab^{-1}, w)$, where $F = \theta_{a,b^{-1},a}$, then the tree T_w rooted at w is almost perfect.*

Call the tree in Proposition 4(i) of type one and call the tree in Proposition 4(ii) of type two.

We now work towards proving in Theorem 5 that in an almost perfect tree of type one or two no two different nodes have the same coordinates. We establish a series of inequalities between the coordinates of nodes in an almost perfect tree. We will use the notation from the proofs of Lemmas 15 and 16 for the remainder of this section. That is, let T_w be an almost perfect subtree rooted at w. Let the nodes in Lb_w be w_i, $i \geq 0$, where $w_0 = w$, and let the right child of each w_i be u_i, as in Figure 4. Also let $k = (ab)_{w_i}$ if the tree is of type one, and $k = (ab^{-1})_{w_i}$ if the tree is of type two, and let $l = (ab^{-1})_u$ for any node u in $T_w \setminus \mathrm{Lb}_w$, as in Figure 3.

Lemma 17. *Let T_w be an almost perfect tree of type one. Let the number of critical syllables in Lb_w be k, and let the number of critical syllables ab^{-1} in the rest of the tree T_w be l. Then for any node m in T_w the following inequalities hold:*

$$(a)_m \geq (b)_m \geq 2l.$$

Proof. We first show that for each node w_i in Lb_w, where $i \geq 0$, the inequality $(a)_{w_i} \geq (b)_{w_i} \geq 2k$ holds. By hypothesis, no $b^{\pm m}$'s occur in w_i. However, subwords of the form $a^{\pm m}$'s are allowed in w_i, where $m \geq 2$. Thus we see that $(a)_{w_i} \geq (b)_{w_i}$ for all i. Since w_i are critical with respect to $\theta_{a,b,a}$, by Lemma 12 we get that $(ba^nb)_{w_i} = 0$, where $n \geq 1$. Now think of the b's as marking the division between a segments. Then for any subword (not syllable) ab in the word the closest left occurence of $b^{\pm 1}$ must be b^{-1} and not b because subwords of the form $(ba^nb)^{\pm 1}$ are not allowed. This left b^{-1} must be preceeded by $a^{\pm 1}$, which makes it not be part of a syllable ab. We therefore get that $(b)_{w_i} \geq 2k$.

We now show that $(a)_{u_i} \geq (b)_{u_i} \geq 2l$. We get u_i from w_i by applying transformation $\theta_{a,b,b}$. Thus $(a)_{u_i} = (a)_{w_i} + (b)_{w_i} - 2k$, $(b)_{u_i} = (a)_{w_i}$, and one can easily check that $(a)_{u_i} \geq (b)_{u_i}$. By the Claim in Lemma 15, $(b^{-1}a^nb^{-1})_{u_i} = 0$, and similar to the previous argument we get $(b)_{u_i} \geq 2(ab^{-1})_{u_i} = 2l$.

If m_1 and m_2 have been obtained from m by applying transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$, respectively, it is immediate to check that $(a)_m \geq (b)_m \geq 2l$ implies $(a)_{m_1} \geq (b)_{m_1} \geq 2l$ and $(a)_{m_2} \geq (b)_{m_2} \geq 2l$. Since the nodes u_i satisfy the inequalities, all the nodes in the tree rooted at u_i will satisfy the inequalities.

It remains to show that $k \geq l$. This can be seen from the fact that the subword ab^{-1} can be obtained only from the subword $ab^{-1}a^{-1}$ in w_i when applying transformation $\theta_{a,b,b}$. This implies that $(a)_m \geq (b)_m \geq 2l$ for all nodes m in T_w. □

Lemma 18. *Let T_w be an almost perfect tree of type two. Let the number of critical syllables in Lb_w be k, and let the number of critical syllables ab^{-1}*

in the rest of the tree T_w be l. Then for any node m in T_w the following inequalities hold:

$$(a)_m \geq (b)_m \geq 2l.$$

Proof. The proof is similar to the proof of the previous lemma. By Lemma 12 none of the w_i nodes contains subwords of the form $(b^{-1}a^nb^{-1})^{\pm 1}$, $n \geq 1$, or any $b^{\pm m}$, $m \geq 2$. Since no $b^{\pm m}$'s occur, but $a^{\pm m}$'s are allowed in w_i, $m \geq 2$, we see that $(a)_{w_i} \geq (b)_{w_i}$ for all i. For the second inequality think of the b's as marking the division between a segments. Then for any subword (not syllable) ab^{-1} in the word the closest left occurence of $b^{\pm 1}$ must be a b and not a b^{-1} because subwords of the form $(b^{-1}a^nb^{-1})^{\pm}$ are not allowed. This left b must be preceeded by $a^{\pm 1}$, which makes it not be part of a syllable ab^{-1}. We therefore get that $(b)_{w_i} \geq 2k$.

We then use arguments identical to the ones in the previous lemma. In this case it is clear that $k \geq l$ since when applying transformation $\theta_{a,b^{-1},b}$ the number of syllables ab^{-1} decreases. □

A consequence of Lemmas 17 and 18 together with the coordinate formulas is the following.

Corollary 2. *Let T_w be an almost perfect tree of type one or two. Let m be a node in T_w and m_1 and m_2 be its children. Then*

$$(a)_{m_j} \geq (a)_m \text{ and } (b)_{m_j} \geq (b)_m, \text{where } j = 1, 2. \tag{13}$$

Thus in an almost perfect tree the coordinates of any child are larger than or equal to the respective coordinates of the parent, which makes the length of any child be larger than or equal to the length of the parent word.

We now show that with the exception of a finite number of cases, the inequalities in Lemmas 17 and 18 are strict inequalities: $(a)_m > (b)_m > 2l$ for all nodes m in T_w.

Lemma 19. *Let T_w be an almost perfect tree of type one or two and let l be the number of critical syllables in $T_w \setminus \mathrm{Lb}_w$. Then either the tree T_w is finite or for any node m (except possibly for w and $\theta_{a,b^{-1},b}(w)$) in T_w the following inequalities hold:*

$$(a)_m > (b)_m > 2l.$$

Proof. Let $w_0 = w, w_1, w_2, \ldots$ be the nodes in Lb_w, and let k be the number of critical syllables in the nodes in Lb_w. It turns out that the coordinate inequalities in the entire T_w depend on the coordinate inequalities for the root w. There are four possible cases.

First, if for w we have $(a)_w > (b)_w > 2k$, then we get strict inequalities in the entire tree by using the coordinate formulas 6, 7, 8 and 9. Second,

suppose $(a)_w = (b)_w > 2k$. As in the first case we get strict inequalities in the rest of the tree by using the coordinate formulas.

Third, suppose $(a)_w = (b)_w = 2k$. Here we show that $w_0 = w_1 = w_2 = \dots$. If T_w is a type one tree, the word w has the form $a^{\epsilon_1}b^{\xi_1}\dots a^{\epsilon_k}b^{\xi_k}$, with $\epsilon_{i-1}, \xi_i, \epsilon_i \in \{-1, 1\}$ not all of the same sign. By applying $\theta_{a,b,a}$ to w_0 we get the same structure and same values for the ϵ_i's and ξ_i's for w_0 as for w_1. A similar argument is valid for a tree of type two. For both type one and two trees, since the word w_j is a cyclic proper image of w_{j-1} none of the w_j's with $j > 0$ appear in the tree. If $k = l$, then for all the nodes in the tree we have $(a)_m = (b)_m = 2l$, which means the tree is finite and not of significance in the larger context (See next section.) If $l < k$ then $(a)_{u_0} = (b)_{u_0} > 2l$, which by the second case above imply strict inequalities in the remainder of the tree.

Fourth and finally, if $(a)_w > (b)_w = 2k$, then we get that $w_0 = w_1 = w_2 = \dots$ as in the third case, we get $(a)_{u_0} = (b)_{u_0} > 2l$ and we get strict inequalities in the rest of tree as in the second case. □

One concept that we need in the proof of Proposition 5 is that of *alternating* words. These are cyclic words, but when viewed as subwords of longer words, they lose this property because we cannot perform cyclic permutations on subwords. Alternating words are words in which the a- and the b-segments have alternate signs, that is, all a-segments consist of positive powers of a, and all b-segments consist of negative powers of b, or vice versa. More formally, alternating words are of the form

$$(a^{k_1}b^{-l_1}a^{k_2}b^{-l_2}\dots a^{k_r}b^{-l_r})^{\pm 1}, \text{ where } r \geq 2,\ k_i,\ l_i > 0,$$

for $1 \leq i \leq r$. The above requirements can be relaxed to allow $k_1 = 0$ and $l_r = 0$, since the words can start and end in either a or b. Lemma 20 describes the behavior of some particular alternating subwords inside the tree T_W.

Lemma 20. *Let w be a node in $T_W \setminus \mathrm{Lb}_W$ and let v be a node in the tree T_w rooted at w such that v contains subwords of type $(b^{-1}a^nb^{-1})^{\pm 1}$ or $(a^{-1}b^ma^{-1})^{\pm 1}$, where $n \geq 1$, $m \geq 2$. Let s be a maximal alternating subword of v that contains $(b^{-1}a^nb^{-1})^{\pm 1}$ or $(a^{-1}b^ma^{-1})^{\pm 1}$. If $s_v = s, \dots, s_w$ are the preimages of s in the words on the path from v to w inside T_w, then*

$$|s| < |s_u| < \dots < |s_w|.$$

Proof. We show that in each parent the length of the preimage of the subword s is strictly larger than the length of the subword s in the child, by proving that preimages under transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ of

alternating words are essentially alternating words. The preimage of an alternating word $a^{k_1}b^{-l_1}a^{k_2}b^{-l_2}\dots a^{k_r}b^{-l_r}$, where $k_i, l_i > 0$ and $r \geq 2$, under $\theta_{a,b^{-1},a}$ is an alternating word because
$\theta^{-1}_{a,b^{-1},a}(a^{k_1}b^{-l_1}a^{k_2}b^{-l_2}\dots a^{k_r}b^{-l_r}) = a^{k_1}(ba^{-1})^{-l_1}a^{k_2}(ba^{-1})^{-l_2}\dots a^{k_r}(ba^{-1})^{-l_r} =$

$$a^{k_1+l_1}b^{-l_1}a^{k_2+l_2}b^{-l_2}\dots a^{k_r+l_r}b^{-l_r}$$

The preimage of an alternating word under $\theta_{a,b^{-1},b}$ is alternating because
$\theta^{-1}_{a,b^{-1},b}(a^{k_1}b^{-l_1}a^{k_2}b^{-l_2}\dots a^{k_r}b^{-l_r}) = b^{k_1}(ab^{-1})^{-l_1}b^{k_2}(ab^{-1})^{-l_2}\dots b^{k_r}(ab^{-1})^{-l_r} =$

$$b^{k_1+l_1}a^{-l_1}b^{k_2+l_2}a^{-l_2}\dots b^{k_r+l_r}a^{-l_r}$$

By Lemma 11 the preimage is the entire word except perhaps for the first or last letter. In all the above examples the length of the preimage is larger than the length of the initial word. This can be seen by noticing that the length of the preimage is longer than $|a^{k_1}b^{-l_1}a^{k_2}b^{-l_2}\dots a^{k_r}b^{-l_r}|$ by $l_1 + l_2 + \dots + l_k$, or by $l_1 - 1 + l_2 + \dots + l_k$, depending on whether we consider the first letter to be part of the preimage. Both expressions are positive, since the $l_i > 0$, and the only way we get zero is by having $l_1 = 1$ and $l_2 = 0$. This can happen if the word s is of the form $a^{k_1}b^{-1}a^{k_2}$, which is not consistent with the way we defined s. In general, for every syllable ab^{-1} or bb in the word s, one letter must have cancelled when applying one of the transformations θ to the preimage, so the preimage is longer. □

A key ingredient in the proof of Theorem 8 is the following lemma.

Lemma 21. *Let v be a node in T_W with $|v| = n$. Let $v = v_0, v_1, \dots, v_r = W$ be the nodes on the path from v to W, where r is a positive integer. Then we have $|v_i| < n + c$, where $0 \leq i \leq r$ and c is a constant that depends on $|W|$.*

Proof. Let us assume that $v = v_0$ belongs to some almost perfect tree. Then there exists a positive integer q, where $0 \leq q \leq r$, such that nodes $v_0, v_1, \dots, v_q$ are in an almost perfect tree. By Theorem 5 in an almost perfect tree the length of a parent is smaller than or equal to the length of the child, so $n = |v_0| \geq |v_1| \geq \dots \geq |v_q|$. We therefore need to find bounds on the length of the nodes $v_{q+1}, \dots, v_r$. If v does not belong to any almost perfect tree then $q = 0$ and we find bounds on all the nodes $v_0, v_1, \dots, v_r$ by using the same arguments.

Recall that Lemmas 9 and 10 provide a formula for the length of a node in terms of the length of its parent. Let m be a node in the tree T_W and m_1 and m_2 be its children. If m_1 and m_2 have been obtained from m by applying transformations $\theta_{a,b,a}$ and $\theta_{a,b,b}$ respectively, and $(ab)_m$ is the

number of times syllable ab appears in m then we get the following relations:

$$|m_1| = (a)_{m_1} + (b)_{m_1} = (a)_m + 2(b)_m - 2(ab)_m,$$
$$|m_2| = (a)_{m_2} + (b)_{m_2} = 2(a)_m + (b)_m - 2(ab)_m,$$
$$\text{so} \quad |m| - |m_1| = 2(ab)_m - (b)_m \ \text{ and } \ |m| - |m_2| = 2(ab)_m - (a)_m, \quad (14)$$
$$\text{which implies} \quad |m| - |m_1| < 2(ab)_m \ \text{ and } \ |m| - |m_2| < 2(ab)_m. \quad (15)$$

When the transformations involved are $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ we get

$$|m| - |m_1| = 2(ab^{-1})_m - (b)_m \ \text{ and } \ |m| - |m_2| = 2(ab^{-1})_m - (a)_m \quad (16)$$

which implies

$$|m| - |m_1| < 2(ab^{-1})_m \ \text{ and } \ |m| - |m_2| < 2(ab^{-1})_m. \quad (17)$$

By Lemma 12 the number of critical syllables when we apply the same transformation can only decrease. That is, for a two-generator word U if we apply transformation $\theta_{a,b,a}$, then the number of ab's in $\theta_{a,b,a}(U)$ is smaller than or equal to $(ab)_U$. The same inequality is valid for the number of ab^{-1}'s when we apply $\theta_{a,b^{-1},a}$ or $\theta_{a,b^{-1},b}$. Thus there exist bounds, which we make precise in the next paragraph, for both $(ab)_m$ and $(ab^{-1})_m$, where m is any node in the tree T_W with ab or ab^{-1} as the critical syllable. As in Theorem 6 and in the entire section, we use the notation illustrated in Figure 5. Each of the finite number of nodes in Lb_W between W and w^*, the critical word with respect to $\theta_{a,b,a}$, is denoted by $W = w_0, w_1, \ldots, w_p = w^*$. The right children of the w_i's, where $0 \leq i \leq p$, are denoted by $u_0 = \theta_{a,b,b}(w_0)$, $u_1 = \theta_{a,b,b}(w_1)$, $\ldots$, $u_p = \theta_{a,b,b}(w_p)$. Node $W = w_0$ might have two additional children: $\theta_{a,b^{-1},a}(W)$ and $\theta_{a,b^{-1},b}(W)$.

For some node m in one of the binary trees T_{u_i} rooted at u_i we have $(ab^{-1})_m \leq (ab^{-1})_{u_i}$, where $0 \leq i \leq p$. For m in Lb_W, the left branch of T_W, or m in the almost perfect tree rooted at w_p, ab is the critical syllable and the upper bound on $(ab)_m$ is $(ab)_W$.

Let $k_1 = (ab)_W$, and $k_2 = \max\limits_{0 \leq i \leq p}((ab^{-1})_{u_i})$.

Then the number of critical syllables for any node m in T_W is smaller than $\max(k_1, k_2)$ and the inequalities (15) and (17) imply that the length of a parent is at most $2k_1$ or $2k_2$ larger than the length of the child. By Lemma 20, if v_q is in T_{u_i}, then the length of the path from v_q to u_i is less than $|u_i|$, which implies that the length of the path between v_q and W is at most $|u_i| + i$, where $0 \leq i \leq p$. Since the length of the path between v_q and $v_r = W$ is at most $\max\limits_i(|u_i| + i)$, we get that $|v_j| < n + c$ for $q < j < r$, where $c = 2 \ \max(k_1, k_2) \ \max\limits_{0 \leq i \leq p}(|u_i| + i)$.

We conclude this proof by providing bounds for c in terms of $|W|$. For any two-generator word the number of times syllable ab can appear is at

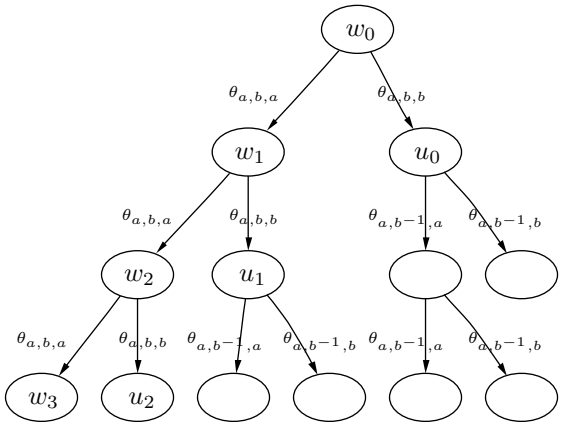

FIGURE 5. Tree T_W

most half the length of the word. Thus $k_1 = (ab)_W \leq \frac{|W|}{2}$. By Lemma 15 $(ab^{-1})_{u_i} \leq (ab^{-1}a^{-1})_{w_i}$, and clearly $(ab^{-1}a^{-1})_{w_i} \leq (ab)_{w_i}$. Since $(ab)_{w_i} \leq (ab)_W$, we also get that $k_2 \leq \frac{|W|}{2}$. Finding bounds on $\max_{0\leq i\leq p}(|u_i|+i)$ is more tedious. Since w_i is u_i's parent we can use formula 7 and we get $|u_i| = |w_i| + (a)_{w_i} - 2(ab)_{w_i}$, and the fact that w_{i-1} is w_i's parent provides $|w_i| = |w_{i-1}| + (b)_{w_{i-1}} - 2(ab)_{w_i}$. Using the above formula for all $0 \leq i \leq p$ and the fact that $(b)_{w_0} = (b)_{w_1} = \cdots = (b)_{w_p}$ gives $|w_i| = |w_0| + i(b)_{w_0} - 2((ab)_{w_0} + \cdots + (ab)_{i-1})$, so

$$\begin{aligned} |u_i| + i &= |w_i| + (a)_{w_i} - 2(ab)_{w_i} + i = 2|w_i| - (b)_{w_i} - 2(ab)_{w_i} + i \\ &= 2(|W| + i(b)_W - 2((ab)_{w_0} + \cdots + (ab)_{w_{i-1}}) - (b)_W - 2(ab)_{w_i} + i. \end{aligned}$$

Now we use the inequalities $(ab)_{w_i} \geq 1$, $(b)_W < \frac{|W|}{2}$ and $p < |W|$ and we get $|u_i| + i < 2|W| + |W|^2$, which gives $c < \frac{W}{2}(2|W| + |W|^2) = |W|^2 + \frac{|W|^3}{2}$. □

9. MAIN RESULTS

Throughout this section we use the tree like structure we have assigned to the set C'_W. Recall that T_W is the two-generator word tree corresponding to the set C'_W as described in Section 6.3. We show that the tree T_W is made up of a finite number of maximal almost perfect trees and a finite number of additional nodes, and we give bounds on these numbers in terms of the length of W. We start by showing that in an almost perfect tree no

two different nodes can have the same coordinates. This will lead to the fact that the number of nodes of a certain length n in an almost perfect tree is linear in n.

Theorem 5. *Let T_w be an almost perfect tree of type one or two with $(a)_w > (b)_w > 2k$, where k is the number of critical syllables in w. Then no two different nodes in T_w have the same coordinates.*

Proof. Let us note that our assumption that $(a)_w > (b)_w > 2k$ is not restricting the number of almost perfect trees that we can consider. By Lemma 19 the inequalities (13) become strict inequalities in essentially all the cases.

Let p' and q' be nodes in T_w, and let p and q be their respective parents. We will use double induction on the distance from p' to w, and q' to w.

Suppose neither p' nor q' is equal to w. If p' and q' have been obtained by applying the same transformations (both by $\theta_{a,b^{-1},a}$ or both by $\theta_{a,b^{-1},b}$), then p and q have the same coordinates by formulas (8) and (9). Hence $p = q$, which implies that $p' = q'$. (See Figure 6.)

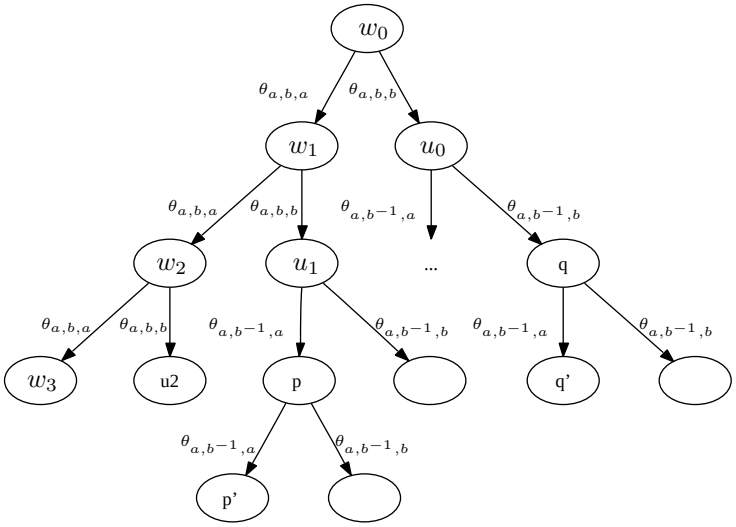

FIGURE 6. Tree T_w

If p' and q' have been obtained by applying different transformations, then we get $(a)_p = (b)_q$ and $(a)_q = (b)_p$ which contradicts the inequalities $(a)_p > (b)_p > 2l$ and $(a)_q > (b)_q > 2l$ of Lemma 19.

Now assume $p' = w_0$. But then q' has different coordinates from p' since on any path from w to a node in the tree the coordinates are strictly increasing. □

Theorem 5 is the first step in characterizing the number of words of a certain length in the set C_W. We now give bounds on the number of maximal almost perfect trees in T_W and on the number of nodes in T_W that do not belong to any almost perfect trees.

Proposition 5. *Let w be a node in $T_W \setminus \mathrm{Lb}_W$. Then the tree T_w rooted at w consists of a finite number c_w of nodes together with at most c_w almost perfect trees, where c_w is a number that depends on the length of w.*

Proof. We have to show that a tree with edges corresponding to transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ only can be written as the union of a finte number of nodes and a finite number of almost perfect trees.

Recall from Lemmas 12, 13 and 16 that a tree in which only transformations $\theta_{a,b^{-1},a}$ and $\theta_{a,b^{-1},b}$ are allowed is almost perfect when the root contains no subwords of the form $(b^{-1}a^nb^{-1})^{\pm 1}$ and $(a^{-1}b^ma^{-1})^{\pm 1}$, where $n \geq 1$ and $m \geq 2$. Notice that any node in an almost perfect tree is itself the root of an almost perfect tree.

Now choose a word v in T_w which is not the root of an almost perfect tree, and therefore does not belong to any almost perfect tree. By Corollary 1 word v must contain a subword of the form $(b^{-1}a^nb^{-1})^{\pm 1}$ or $(a^{-1}b^ma^{-1})^{\pm 1}$, where $n \geq 1$ and $m \geq 2$. Consider a maximal alternating subword of v that contains $(b^{-1}a^nb^{-1})^{\pm 1}$ or $(a^{-1}b^ma^{-1})^{\pm 1}$, call it s, and trace the preimages $s_v = s, \ldots, s_w$ of s in the words on the path from v to w inside T_w. That is, if u is the parent of v in the tree T_w, let s_u be one of the possible preimages in u of the subword s, and we iterate this notation.

Lemma 21 shows that any word v which is not the root of an almost perfect tree in T_w is at most $|w|$ steps away from w. In fact, if λ_w is the sum of the length of all maximal alternating subwords in w, then the number of steps is less than $\lambda_w - |s|$. The length of the path between w and v depends on the kind of transformations that connect the nodes in this path. For example, if $s = b^{-1}a^nb^{-1}$ and all transformations on the edges in the path between v and w are $\theta_{a,b^{-1},a}$, then the length of the preimage increases by one at each step. This happens because $\theta^{-1}_{a,b^{-1},a}(b^{-1}a^nb^{-1}) = ab^{-1}a^{n+1}b^{-1}$. On the other hand, if $s = a^{-1}b^ma^{-1}$ and all the transformations connecting v to w are $\theta_{a,b^{-1},b}$, then the length of the preimage increases by at least m at each step and thus there are fewer nodes on this path than in the previous case.

Let c_w be the number of nodes in T_w that are the roots of maximal almost perfect trees. By maximal almost perfect subtrees we mean trees

that do not belong to any bigger almost perfect trees. Since there are 2^n nodes at height or level n in a binary tree, we get that there are at most $2^{\lambda_w} < 2^{|w|}$ almost perfect trees. This shows that $c_w < 2^{|w|}$, and in fact c_w is often much smaller since some paths are shorter than $|w|$ as noticed in the previous paragraph.

We can also find a bound on the number of nodes in T_w that do not belong to any almost perfect tree. These are among the nodes of height at most $\lambda_w - 1$. Since T_w is a binary tree there are at most 2^{λ_w} such nodes, we can use the bound c_w for there nodes as well as for the number of maximal almost perfect trees. This concludes the proof of Proposition 5. □

Theorem 6. *The tree T_W consists of a finite number of nodes together with at most $|W|2^{|W|}$ maximal almost perfect trees.*

Proof. In order to exhibit the almost perfect subtrees we need to display the nodes that can be the potential roots for such subtrees.

By Lemma 12 we get a critical word with respect to $\theta_{a,b,a}$ when we apply this transformation to W repeatedly. Call this critical word $w_{\theta_{a,b,a}}$. By Lemma 15, the tree rooted at $w^* = \theta_{a,b,a}(w_{\theta_{a,b,a}})$ is almost perfect. Denote each of the finite number of nodes in Lb_W between W and w^* by $W = w_0, w_1, \ldots, w_p = w^*$ and denote their right children by $u_0 = \theta_{a,b,b}(w_0)$, $u_1 = \theta_{a,b,b}(w_1), \ldots, u_p = \theta_{a,b,b}(w_p)$ as in Section 3.5, and illustrated by Figure 5. Node $W = w_0$ might have two additional children: $\theta_{a,b^{-1},a}(W)$ and $\theta_{a,b^{-1},b}(W)$.

By Proposition 5, for every $0 \leq i \leq p$ the tree rooted at u_i consists of a finite number of nodes together with a finite number of almost perfect trees. In particular, the tree T_W consists of the nodes on Lb_W (lying between W and w^*), the nodes in the T_{u_i} that do not belong to any almost perfect tree, and $c_{u_1} + c_{u_2} + \cdots + c_{u_p}$ almost perfect trees.

We now give a bound for $c_{u_1} + c_{u_2} + \cdots + c_{u_p}$ in terms of $|W|$. Recall that p is the number of times we need to apply transformation $\theta_{a,b,a}$ to W in order to get a critical word with respect to $\theta_{a,b,a}$. That is, p is such that $\theta^p_{a,b,a}(W)$ contains no ba^mb, where $m > 0$. By Lemma 12 we get that p is equal to the length of the a-segment in the largest subword of type ba^kb in W, where $k > 0$.

Recall that λ_{u_i} is the sum of the lengths of all maximal alternating subwords in u_i, where $0 \leq i \leq p$. By Proposition 5 we have $c_{u_i} < 2^{\lambda_{u_i}}$. We claim that $\lambda_{u_i} < |W|$ for each i. We show this by using the fact that $\theta^{-1}_{a,b,b}(b^{-1}a^nb^{-1}) = b^{-1}a^{-1}b^{-(n+1)}a^{-1}$, and $\theta^{-1}_{a,b,b}(a^{-1}b^ma^{-1}) = b(ab)^mb$. Thus λ_{u_i} depends on the number of words of type $a^{-1}b^{-(n+1)}a^{-1}$ and $b(ab)^mb$ in w_i. By Lemma 12 such words can only come from subwords of type ba^kb or $ba^{k_1}ba^{k_2}\ldots b$ in W, so λ_{u_i} is smaller than the total length of

all subwords of type ba^kb in W, where $k, k_i > 0$. In particular, $\lambda_{u_i} < |W|$. This shows that $c_{u_i} < 2^{|W|}$ for each $0 \leq i \leq p$, so the number of almost perfect trees in T_W is less than $p2^{|W|}$, and since $p < |W|$ we get a bound of $|W|2^{|W|}$ for the number of maximal almost perfect trees.

By Proposition 5 the number of nodes in T_W not belonging to any almost perfect tree is also bounded by $|W|2^{|W|}$, since the number of such nodes in any of the trees T_{u_i} is bounded by c_{u_i}, for all $0 \leq i \leq p$. □

Theorem 7. *The number of words in C'_W of length n is linear in n.*

Proof. Let α be the number of maximal almost perfect trees in T_W and let β be the number of nodes not belonging to any almost perfect tree. We first prove that the number of two-generator words in C' of length n is linear in n.

A two-generator word u in a and b has length n if and only if $(a)_u+(b)_u = n$, so in order to find all the two-generator words of length n we need to determine the number of pairs $((a)_u, (b)_u)$ for which $(a)_u + (b)_u = n$. For any positive integer n there are $n-1$ pairs (x, y) of positive integers such that $x + y = n$. By Theorem 5 in an almost perfect subtree no two nodes have the same coordinates.

The above fact combined with Theorem 5 shows that there can be at most n words of length n in an infinite almost perfect tree. Thus Theorem 6 shows that there can be at most $\alpha n + \beta$ nodes of length n in T_W.

Now let us introduce the three-generator words into the picture. Notice that for every two-generator word u in the set C' the set L'_u will contain at most two words that are three-generator words, one produced by $\rho_{a,a,c}$ and one produced by $\rho_{a,b^{-1},c}$ (see Proposition 3). In fact $\rho_{a,a,c}$ will produce mostly redundant words because a node in an almost perfect tree contains no $b^{\pm m}$, where $m \geq 2$.

Let us assume that u is a node in an almost perfect tree with number of critical syllables ab^{-1} equal to l. Recall that this means the number of critical syllables for any node in the almost perfect tree is the same and equal to l. Then the length of $\rho_{a,b^{-1},c}(u)$ is $2((a)_u + (b)_u) - 2l = 2|u| - 2l$. Thus the number of three-generator words of length n in the almost perfect subtree corresponds to the number of two-generator words of length $\frac{n+2l}{2}$ in that subtree. We have shown that the number of two-generator words of length $\frac{n+2l}{2}$ is less than n for $n > 2l$ in any almost perfect subtree, since the number of nodes of length k is less than k in any almost perfect subtree. Therefore the number of three-generator words in C' is less than $\alpha n + \beta$ for n sufficiently large. □

Corollary 3. *The number of words in C'_W of length smaller than or equal to n is quadratic in n.*

Proof. Follows immediately from Theorem 7. □

Here we present the main theorem, which states that the number of words in C'_W that are generated before finding the last word of length n is quadratic in n. This result helps to completely determine the complexity of the endomorphism for two-generator words.

Theorem 8. *The number of words in C'_W that are generated before finding the last word of length n is quadratic in n, i.e. it is smaller than γn^2, where γ is a constant depending on the length of W.*

Proof. We start with some word $v \in C'_W$ of length n and we want to count all the words produced by our algorithm before v has been produced. Notice that while three-generator words are being produced, they do not generate any further words, so we first count only the two-generator words produced before v. Let St_v be the finite subtree of T_W containing all the two-generator words produced before v. St_v is a subtree since at any point in the algorithm a two-generator word is produced by its parent. Recursively, this means that there is a path from any node in St_v to W. It is connected since we consider the set C' and we do not remove proper cyclic images other than the ones already removed in order to get T_W, as explained in Section 4.2. We count the number of nodes in St_v by finding upper bounds on their length. Since St_v consists of portions of almost perfect trees and a finite number of nodes that do not belong to any almost perfect tree, knowing upper bounds on the length of the words will give us the number of nodes in the almost perfect trees.

Let us assume that v belongs to L'_{v_1}, where v_1 is a two-generator word in C'_W. If v is a three-generator word, then clearly $|v_1| < n$, so we have to count all the two-generator words that have been produced before v_1. We can therefore assume that v is a two-generator word of length n, where as before $v \in L'_{v_1}$.

First consider all the terminal nodes in St_v, that is, the nodes of degree one. Let x be a terminal node. We use the property of the Edmunds-Sims algorithm that given two words in C', the shorter one is chosen to produce the corresponding L set first. Let y be the parent of x, that is, x is in L'_y. We find bounds on $|y|$ and in general, on the length of all non-terminal vertices. Let $v = v_0, v_1, \ldots, v_r = W$ be the nodes on the path from v to W, where r is a positive integer. If x was created by the algorithm after v_i and before v_{i+1} for some positive integer i, then $|y| < |v_i|$. For every terminal node $x' \in \mathrm{St}_v$ there exist two consecutive nodes v_j and v_{j+1} on the path from v to W such that x' was generated by the algorithm after v_j and before v_{j+1}. So if $x' \in L'_{y'}$ then we get $|y'| \leq |v_j|$. Thus the length of each non-terminal node is bounded by the length of some node on this path.

By Lemma 21 all the non-terminal vertices in St_v that belong to almost perfect subtrees must be words of length smaller than $N = n + c$. Since by Corollary 3 there are only a quadratic number of words of length smaller than or equal to N, we get that there exists a constant γ' such that there are at most $\gamma' n^2$ non-terminal vertices in St_v. Counting the terminal vertices and the non terminal degree one vertices that have been produced before v adds only at most $2\gamma' n^2$ to get the total number of two-generator words that have been produced before v. The number of three-generator words produced is at most double the number of two-generator words, and therefore we get a bound of $5\gamma' n^2$ on the number of words produced before finding the last word of length n. Let $\gamma = 5\gamma'$ and we get the statement of the theorem. □

Although all our results are about the set C'_W, they are valid for the set C_W since the set C_W contains words of the same length and in the same orbit of $Aut(M)$ as the words in C'_W. In particular, we have the following corollaries.

Corollary 4. *The number of words in C_W of length n is linear in n.*

Corollary 5. *The number of words in C_W of length smaller than or equal to n is quadratic in n.*

Corollary 6. *The number of words in C_W that are generated before finding the last word of length n is quadratic in n, i.e. it is smaller than γn^2, where γ is a constant depending on the length of W.*

10. Conclusion

Recall that in order to determine whether an element $[U]$ of F is an endomorphic image of an element $[W]$, it is sufficient to check whether there exists W' in C_W such that U is a proper cyclic image of W'. As the length of a proper image of a word is longer than or equal to the length of the word, we only need to perform the check for the words W' in C_W of length at most the length of U.

The results from the previous section show that one only needs to check a quadratic number of words when W is a two-generator word, and since all the other operations used in the implementation of the algorithm (see the Appendix) use polynomial time and amount of space, the endomorphism problem for two-generator words is polynomial in the length of U.

The following argument produces an estimate for the degree of the polynomial in $|U|$. A key ingredient of such an estimate is the complexity of the procedure that determines whether a word V is a proper cyclic image of a word V'. One can find details about this procedure in the Appendix.

Here we need the fact that the time is $O(|V|^{m+1})$, where m is the number of generators that appear in V'.

Let us choose a word W' in C_W with $|W'| = n$, where n is some integer smaller than or equal to $|U|$. By the definition of C_W, before adding the word W' to the set C one needs to make sure W' is not a proper cyclic image of any of the words already in C_W. However, our results depend only on a constant number of local checks for proper cyclic images in consecutive L sets (consecutive in the sense that we consider L_V and $L_{V'}$, where V' is in L_V). Each check takes at most $O(n^4)$ time, since all words in C_W have at most 3 generators. Since we perform a constant number of such checks, we get a complexity of $O(n^4)$ for the number of operations needed to decide whether W' should be added to the set C_W. By Theorem 7 there are $O(n)$ words of length n in C_W. Thus the time it takes to check for all words of length n whether they should be added to C_W is $O(n^5)$. All other operations involved in producing the set C_W require less time than checking for proper images and will therefore not change the degree of the polynomial. The total time for producing the words in C_W that have length smaller than or equal to $|U|$ comes from the summation of the number of operations required for producing the words of length n, where n runs from 0 to $|U|$. Since we obtained a time of $O(n^5)$ for handling all the words of length n, the complexity for handling all words of length smaller than or equal to $|U|$ in the set C_W will be $O(|U|^6)$.

Notice that in implementing a program that produces the set C'_W we do not need to generate the words we proved are (locally) proper images. In this approach we can completely avoid the checks for proper images, and therefore the complexity will be better.

Finally, for every word W' in C_W or C'_W with $|W'| \leq |U|$ we need to check whether U is a proper cyclic image of W'. This takes $O(|U|^4)$ time for each W' because the number of generators in C_W or C'_W is at most three. As pointed out in the proof of the main result, words of length greater than $|U|$ are generated before producing the last word of length $|U|$. While the operations involved in producing these words increase the running time, they also do not modify the degree of the polynomial because we do not need to check whether U is a proper image of any of these words. By Corollary 3 there are about $O(|U|^2)$ words in C_W or C'_W of length smaller than or equal to $|U|$, so checking whether U is a proper image of any of the words in C_W or C'_W will take $O(|U|^6)$. The running time for producing the relevant portion of the set C_W or C'_W and the running time for checking whether U is a proper image of words in C_W or C'_W together give a complexity of $O(|U|^6)$ for checking whether U is an endomorphic image of W.

As far as the constants involved in the complexity are concerned, Theorem 6 provides the high bound of $|W|2^{|W|}$ on the number of almost perfect trees in the composition of the set C'_W. This translates into a rather impractical complexity constant for our algorithm. However, we would like to note that in most examples we looked at this constant has turned out to be much lower, of the order of $|W|$.

We can avoid the above exponential constant in our final complexity calculation if we use the result of Khan (Theorem 4), which gives us an $O(n^2)$ bound on the number of words of length n in an automorphic orbit of a cyclically reduced word of length n. Since all two-generator words in C'_W are cyclically reduced automorphic images of W, this gives a $O(|U|^3)$ bound on the number of words in C_W or C'_W of length smaller than or equal to $|U|$, with a numerical constant. Taking this into account, we obtain a final complexity bound of $O(|U|^7)$ for our algorithm, with a small constant.

Appendix

Implementation of the procedures

This appendix provides a description of the basic algorithms on words that are being used in the implementation of the Edmunds-Sims approach. A similar account of the procedures can be found in [3], section 4. We also give a rough estimate of the computational complexity of these algorithms. Although the free group and monoid are infinitely generated, the estimate of the running time for all procedures assumes that we handle words on the first million generators in X only, which can be represented by fixed precision signed integers.

First element in orbit. Let M be the free monoid with basis $X^{\pm}$ and let F be the free group generated by X. We fix a linear order on $X^{\pm}$, say $a_1 < a_1^{-1} < a_2 < a_2^{-1} < \ldots$, and we compare words using the corresponding length-plus-lexicographic ordering, which is a well-ordering on M.

Given a word V, we can find the first element $\overline{V}$ in the orbit of $Aut(M)$ containing V with the following procedure. Let $V = v_1 v_2 \ldots v_r$, where all v_i are in $X^{\pm}$. Our goal is to assign to each v_i an element a_j, where $a_j \in X^{\pm}$. Let f be the assignment map. Initially no value of f has been defined.

- For i starting with $i = 1$ check whether $f(v_i)$ has been defined. Let j equal the number of generators in $\{f(v_1), \ldots, f(v_{i-1})\}$.
- If an assignment has already been made, then store $f(v_i)$ in z_i and process v_{i+1}.
- If not, then assign v_i to a_{j+1}, that is, let $f(v_i) = a_{j+1}$ and $f(v_i^{-1}) = a_{j+1}^{-1}$, and store $f(v_i)$ in z_i.

The resulting word $z_1 z_2 \dots z_r$ is $\overline{V}$. For example, if $V = a_2^{-1} a_1 a_2 a_1^{-1} a_3^{-1} a_2 a_3$, then $\overline{V} = a_1 a_2 a_1^{-1} a_2^{-1} a_3 a_1 a_3^{-1}$. Since we process each letter in the word V the running time for this procedure is linear in $|V|$.

In order to find $\widehat{V}$ we need to apply the above procedure to each of the cyclic permutations of V. Since there are $|V|$ such cyclic permutations, the running time for computing $\widehat{V}$ is quadratic in $|V|$.

Test for redundancy. In order to test whether a word V is redundant we use the following procedure. We assume that $V = \overline{V}$, which means the set of generators occuring in V is of the form $\{a_1, a_2, \dots, a_m\}$ for some integer $m \leq |V|$.

• For each generator x in V check whether previously x or x^{-1} has been seen only next to a single letter y. Record y.

• For x and y as above check whether all occurrences of x, x^{-1}, y and y^{-1} in V are in subwords xy or $y^{-1}x^{-1}$. If the answer is positive, then the word is redundant. Otherwise continue with the next generator.

Since we process each letter in the word V the running time for this procedure is linear in $|V|$.

Test for proper image. Let U and V be two cyclically reduced words in M. Here we provide a procedure that checks whether V is a proper image of U.

For $U = u_1 u_2 \dots u_r$, where the u_i are letters, we are looking for a monoid endomorphism ϕ such that $\phi(U) = V$ and $\phi(u_i)$ is nonempty for all i.

• Choose a value for $\phi(u_1)$ using the fact that it must be a nonempty prefix of V. Consider the maximal prefix $u_1 u_2 \dots u_{i-1}$ of U such that $u_1 = u_2 = \dots = u_{i-1}$. Then $p = \phi(u_1 u_2 \dots u_{i-1})$ is defined.

• If p is not a prefix of v, then try another choice for $\phi(u_1)$. If p is a a prefix for V, then $V = pq$ for some word q and $\phi(u_i)$ must be a prefix of q, so make a choice of $\phi(u_i)$ as in the previous step.

• Obtain ϕ by a backtrack search as in the above steps.

The procedure runs in time polynomial in the length of V. More precisely, the time is $O(|V|^{m+1})$, where m is the number of generators that appear in U. We obtain this running time since for every generator u there are at most $|V|$ subwords of V as possible choices for $\phi(u)$, and once ϕ has been determined for all generators in U, one must still check whether $\phi(U)$ is V. It takes time $|V|$ to see if the choice of ϕ is correct. This time can be reduced to $O(|V|^m)$ since knowing the values of ϕ for $m-1$ generators determines the value of ϕ for the last generator.

Determining whether V is a proper cyclic image of U will then take $O(|V|^{m+1})$ since there are $|V|$ cyclic permutations of V.

Acknowledgments

This paper is a revised version of the thesis written under the direction of Charles Sims and submitted to the Department of Mathematics of Rutgers University. The author thanks Prof. Charles Sims for suggesting this problem and for lots of helpful comments along the way. The author was partially supported by the Marie Curie Intra-European Fellowship number 515027 and thanks the Centre de Recerca Matemàtica in Bellaterra for its hospitality.

References

[1] C. C. Edmunds. On the endomorphism problem for free groups. *Comm. Algebra 3*, 1975.

[2] C. C. Edmunds. On the endomorphism problem for free groups II. *Proc. London Math. Soc. 38*, 1979.

[3] Charles C. Sims. On the complexity of the endomorphism problem for free groups. *Ohio State Univ. Math. Res. Inst. Publ. 8*, 2001.

[4] Laura Ciobanu. On the complexity of the endomorphism problem in free groups. *PhD Thesis, Rutgers University, Mathematics Department*, pages 1–76, 2005.

[5] L. P. Comerford and C. C. Edmunds. Solutions of equations in free groups. *Sympos. in Group Theory, Singapore*, pages 347–355, 1989.

[6] Charles C. Edmunds. A condition equivalent to the solvability of the endomorphism problem for free groups. *Proc. Amer. Math. Soc.*, 76(1):23–24, 1979.

[7] G. S. Makanin. Equations in a free groups (Russian). *Izv. Akad. Nauk SSSR Ser. Mat. 46*, 1982.

[8] R. I. Grigorchuk and P. F. Kurchanov. On quadratic equations in free groups. In *Proceedings of the International Conference on Algebra, Part 1 (Novosibirsk, 1989)*, volume 131 of *Contemp. Math.* Amer. Math. Soc., 1992.

[9] Bilal Khan. The structure of automorphic conjugacy in the free group of rank two. In *Computational and experimental group theory*, volume 349, pages 115–196. Amer. Math. Soc., 2004.

[10] A. Koscielski and L. Pacholski. Makanin's algorithm is not primitive recursive. *Theoretical Computer Science*, 191, 1998.

[11] R. C. Lyndon. The equation $a^2b^2 = c^2$ in free groups. *Michigan Math. Journal*, 6:155–164, 1959.

[12] A. I. Mal'cev. On the equation $zxyx^{-1}y^{-1}z^{-1} = aba^{-1}b^{-1}$ in a free group. *Algebra i Logika Sem.*, 1(5):45–50, 1962.

[13] Alexei G. Myasnikov and Vladimir Shpilrain. Automorphic orbits in free groups. *Journal of Algebra*, 269(1):18–27, 2003.

[14] A. A. Razborov. On systems of equations in a free group. *Math USSR - Izv.*, 25(1):115–162, 1985.

[15] Vitaly A. Roman'kov. On nondecidability of endomorphism problem in free nilpotent groups and free rings. *Algebra i Logika*, 16:457–471, 1977.

[16] P. Schupp. Quadratic equations in groups, cancellation diagrams on compact surfaces, and automorphisms of surface groups. In *Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976)*, volume 95 of *Stud. Logic Foundations Math.*, pages 347–371. North-Holland, Amsterdam, 1980.

[17] Paul E. Schupp. On the substitution problem for free groups. *Proc. Amer. Math. Soc*, 23:421–24, 1969.
[18] M. J. Wicks. Commutators in free products. *J. London Math. Soc.*, 37, 1962.

Laura Ciobanu, Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, New Zealand
E-mail: ciobanu@math.auckland.ac.nz