# Whitehead Automorphisms and Equivalent Words

Rebecca Virnig
University of Chicago

Advisor *: Professor Dennis Garity
NSF sponsored REU Program
Oregon State University

September 25, 1998

**Abstract**

Properties of Whitehead automorphisms on equivalence classes of words are investigated with respect to word length. Factoring of Whitehead automorphisms is explored and necessary generators are determined. A bound with respect to equivalence class is determined on the number of automorphism necessary between equivalent words and specific word classes are analyzed.

*Thanks to Dennis Garity for his help and advice.

# 1 Introduction

In 1936, J.H.C. Whitehead demonstrated that if any two finitely generated words of a free group are equivalent under some automorphism of the group, then they are equivalent under a finite sequence of Whitehead automorphisms,which are a special class of automorphism.[5] [6] Specifically, Whitehead proved that if $w$ and $v$ are equivalent words, such that $v$ has the minimum length occurring in its equivalence class, then the words obtained after each step in the transformation from $w$ to $v$ are of strictly decreasing length until the minimum is obtained, after which the length remains constant.

Whitehead's automorphisms are divided into two classes, Whitehead Type I automorphisms and Whitehead Type II automorphisms. A basic definition of these classes, Whitehead's theorem, and an overview of the notation used in this paper is presented below.

**Notation**

- $F_n = F(x_1, \ldots, x_n)$ will denote the free group of rank $n$, with $x_1, \ldots, x_n$ as generators. Only the trivial relation is admitted.
- $\overline{x_k}$ will denote the inverse of $x_k$. The empty word will be represented by 1.
- $w \sim v$ indicates that the words $w, v \in F$ are equivalent under some automorphism of $F$
- $|w|$ will denote the length of a word $w$, after it has been reduced.
- The set of letters, $\{x_1, x_2, \ldots, x_n, \overline{x_1}, \overline{x_2}, \ldots \overline{x_n}\}$ will be represented by $L_n$.

**Definition 1.1** *Suppose $w \in F_n$ and $x \in \{x_1 \ldots, x_n\}$. The sum of the number of occurrences of $x$ and the number of occurrences of $\overline{x}$ in $w$ is called the* **weight** *of $x$ in $w$.*

**Definition 1.2** *The* **Whitehead Type I Automorphisms** *of $F$ are those permutations $T$ acting on $L_n$ that preserve inverses, i.e. $\overline{T(x)} = T(\overline{x})$. Namely, the intersection of $AutF$ and $SymL$.* $\mathbf{I(F_n)}$ *represents the set of all Type I automorphism on $F_n$.*

**Definition 1.3** *If $F = F_n$, then given any $x \in L_n$ and $A \subset L_n$ such that $x \in A$ and $\overline{x} \notin A$, the pair $(A, x)$ is called a* **Whitehead Type II automorphism**, *where $(A, x)$ is the automorphism determined by the following actions on each $y \in L_n$:*

$$(A,x)y = \begin{cases} yx & \text{if} \quad y \in A, \overline{y} \notin A, y \notin \{x, \overline{x}\} \\ \overline{x}y & \text{if} \quad y \notin A, \overline{y} \in A, y \notin \{x, \overline{x}\} \\ \overline{x}yx & \text{if} \quad y, \overline{y} \in A \\ y & \text{otherwise} \end{cases}$$

*The set of all Type II automophisms on $F_n$ is represented by* $\mathbf{II(F_n)}$.

**Example 1.1** *$T = (x_1\ \overline{x_3}\ \overline{x_2})(\overline{x_1}\ x_3\ x_2)$ is a Whitehead Type I automorphism, written in cyclic notation.*

**Example 1.2** *If $A = \{x_1, x_2, \overline{x_1}, \overline{x_3}\}$, then $(A, x_2)x_2x_1x_3 = x_2\overline{x_2}x_1x_2\overline{x_2}x_3 = x_1x_3$.*

**Whitehead's Theorem 1.1** *If $w, v \in F$ such that $w \sim v$ and $v$ is of minimum length for their equivalence class, then there is some sequence $S_1, S_2, \ldots, S_m$ of Whitehead automorphisms such that $S_m \cdots S_2S_1(w) = v$ and for all positive integers $k \leq m$, $|S_k \cdots S_2S_1(w)| \leq |S_{k-1} \cdots S_2S_1(w)|$ with strict inequality unless $S_{k-1} \cdots S_2S_1(w)$ is also of minimum length.*

Whitehead's orginal proof used topological methods [5], however Higgins and Lyndon [1], Rapaport [3], and others have given algebraic proofs of his theorem and extended the results. Whitehead's work demonstrated that a finite set of his automorphism would suffice to go between equivalent words. However, his work did not address the number of such automorphism required or a method other than brute-force in determining when a given word is minimal.

These questions have been investigated in some detail for free groups of rank two. In 1979, Sanchez [4] developed a counting method based on the weights of generators in a word that determines when a word is minimal. However, his proof considered only two generators, and is not directly extendable to higher ranks.

Equivalence of words in groups of higher rank becomes important in topology for determining when curves are equivalent. In his 1997 REU project, Micheal Lau began investigating equivalence classes of words in a free group of rank three. His primary result, with the help of Dr. Garity, was the creation of a computer program to implement Whitehead's algorithm to reduced words and generate their equivalence classes. This forms the basis for further study.

# 2 Background Work

The following lemmas and defintions were presented by Lau [2], and are assumed without reference through this work. In addtion the main results of Rapapart [3] and Higgens [1] are given and are refrenced in various proofs.

**Lemma 2.1** *Whitehead Type I automorphisms never change the length of a word.*

**Lemma 2.2** *If a letter $x$ and its inverse $\overline{x}$ are not present in a word $w$, then no Whitehead Type II automorphisms of the form $(A, x)$ will reduce the length of $w$.*

**Lemma 2.3** *If $w$ and $v$ are equivalent words in $F_n$ and $v$ is of minimum length, then $\exists\ S_1, S_2, \ldots, S_s$ such that*
*(i)* $S_s \cdots S_2 S_1(w) = v$
*(ii)* $S_1, S_2, \ldots, S_{s-1}$ *are Whitehead Type II automorphisms*
*(iii)* $S_s$ *is a (possibly trivial) Whitehead Type I automorphism*
*(iv)* $|S_k \cdots S_2 S_1(w)| < |S_{k-1} \cdots S_2 S_1(w)|$ *with equality when* $S_{k-1} \cdots S_2 S_1(w)$ *is of minimum length.*

**Definition 2.1** *Suppose $w, v \in F_n$ with $w = y_1, \ldots, y_m$ for some $y_i \in Ł_n$. We say that $w$ and $v$ are* **cyclically equivalent** *if $v = y_{k+1} y_{k+2} \cdots y_m y_1 \cdots y_k$ for some $k \leq m$.*

**Definition 2.2** *Two words $w_1, w_2 \in F_n$ are said to be* **CP-equivalent**[1] *if there exists a Whitehead Type I automorphism $T$ such that $T(w_1)$ is cyclically equivalent to $w_2$. This equivalence is denoted by $\overset{CP}{\sim}$.*

[1] *C*yclically equivalent after Whitehead Type I automorphisms (*P*ermutations)

**Definition 2.3** *Given a minimal-length word* $w \in F_n$*, we say that* $w$ *is in* **Reduced-CP (R-CP) form** *if it is the first word of its CP-equivalence class to occur in the natural lexicographic ordering given by the ordered (first-to-last) set* $L_n = \{x_1, x_2, \ldots, x_n, \overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}\}$.

**Remark 2.1** CP-equivalence implies Whitehead equivalence. That is, if $w \overset{CP}{\sim} v$, then $w \sim v$.

**Remark 2.2** *For groups of small rank, it is often convinent to use* $\{a, b, c, \ldots\}$ *as generators. Throughout the paper, the examples will use the free group* $F = F(a, b, c)$ *unless otherwise stated.*

**Example 2.1** *The words* $abac\overline{b}\overline{c}$ *and* $\overline{b}\overline{c}abac$ *are cyclically equivalent.*

**Example 2.2** *The word* $\overline{c}a\overline{b}\overline{a}\overline{b}\overline{c}$ *is CP-equivalent to* $aabc\overline{b}\overline{c}$*. Moreover,* $aabc\overline{b}\overline{c}$ *is in R-CP form.*

**Remark 2.3** *R-CP words always begin with a string of one or more consecutive a's. This is always the longest string of consecuative letters in a word.*

**Theorem 2.1** *If a letter* $x$ *has weight* $1$ *in a word* $w$*, then* $w$ *has minimal length* $1$.

**Corollary 2.1** *Suppose* $w$ *is a word of length* $n > 1$*. Then* $w$ *contains no more than* $\left\lfloor \frac{n}{2} \right\rfloor$ *letters of nonzero weight.*

**Theorem 2.2** *(Rapaport [3]) Suppose that* $w$ *is a minimal word in* $F_n$*, and that* $x_1, \ldots, x_n$ *have weights* $k_1, \ldots, k_n$*, respectively, in* $w$*. Then the letters* $x_1, \ldots, x_n$ *have the same weights in any minimal word* $v \sim w$*, up to permutation of subscripts.*

**Theorem 2.3** *(Higgens [1]) Let* $A \subset L_n, B = L_n \setminus A$ *Define* $A.B_w$ *as the number of pairs* $x\overline{y}$ *or* $y\overline{x}$ *where* $x \in A$*, and* $y \in B$ *in a word* $w$*. Define* $L_n.x_w$ *as the weight of* $x$ *in* $w$*. Then if* $T = (A, x)$*,* $\Delta T = |T(w)| - |w| = A.B_w - L_n.x_w$*. The subscript* $w$ *is usually supressed.*

# 3 Results on Whitehead Automorphisms

In analyzing free groups and their decompostions into Whitehead equivalence classes, it is necessary to to first understand how the Whitehead automorphisms work and interact with eachother. This section focuses primially on the properties of the Whitehead automorphisms. Unless otherwise noted the word automorphism is used to refere to Whitehead automorphisms. The first part anyalizes cancellations that automorphisms can induce within a word. Knowlege of these cancelations plays a role in the study of how certain words are affected by the automorphism and connections between various words. The second part demonstrates how Type II automorphism may be factored into a simple class of one-letter automorphisms, properties related to that class, and sufficent generators of Type II automorphisms. It also examines the different types of automorphisms, Type I, Type II, and cycles, interact with eachother.

## 3.1 Cancellations under Automorphisms

**Lemma 3.1** *In appling a sequence of automorphism to a word, the resulting reduced word is unaffected by when the trival cancelations are made. (e.g. a word may be reduced between appling each automorphism, or it may be reduced after all transformations have been applied)*

**Proof** Let T be an automorphism. First note that if $T \in I(F_n)$, then it takes inverses to inverses, so any pair of letters $x\overline{x}$ that will cancel is taken to another pair $y\overline{y}$ that will also cancel. If $T \in II(F_n)$, it suffices to show that the pair $x\overline{x}$ will reduced down to 1 under any single Type II automorphism that affects $x$. For any letter $y \notin \{x, \overline{x}\}$. If $T(x) = xy$ then $T(x\overline{x}) = xy\overline{y}\overline{x} = 1$ If $T(x) = yx$ then $T(x\overline{x}) = yx\overline{x}\overline{y} = 1$ If $T(x) = \overline{y}xy$ then $T(x\overline{x}) = \overline{y}xy\overline{y}\overline{x}y = 1$. ■

**Note** A cycle will not change the adjacentcy of canceling letters, however, if there are cycles in a sequence of automorphisms, the cycle may change depending on when the cancelations are made. Since it is generally obvious when this occurs, unless necessary, it will not be indicated when a cycle is adjusted in this way.

**Lemma 3.2** *Let $S \in II(F_n)$ such that $S(a) = ay$ or $S(a) = ya$ for some $y \notin \{a, \overline{a}\}$. Let $x_1, x_2$ be aribrtrary strings such that $x_1$ and $x_2$ do not respecitively begin or end with $\overline{a}$. Then if $a'$ and $a''$ are two consecutive as, $S(x_1 a' a'' x_2)$ will never reduce to $x_1' a a x_2'$ where one of the as is in $\{a', a''\}$ and $x_1', x_2'$ are some reduced strings.*

**Proof** If $S(a) = ay$ then $S(x_1 a' a'' x_2) = S(x_1) a' y a'' y S(x_2)$. To get $a'a''S(x_1)$ must end with $a\overline{y}\overline{a}$ or $S(x_2)$ must begin with $\overline{y}\overline{a}\overline{y}a$. $S(x_1)$ ending in $\overline{y}\overline{a}$ implies $x_1$ ends in $\overline{a}$, while $S(x_2)$ begingin with $\overline{y}\overline{a}\overline{y}a$ implies $x_2$ begins with $\overline{a}$. Thus the $y$ between the $a$s cannot be canceled out. It follows similarrly for $S(a) = ya$. ■

**Lemma 3.3** *Let $S \in II(F_n)$ such that $S(a) = ay$ or $S(a) = ya$, $y \notin \{a, \overline{a}\}$. Let $w = x_1 a x_2$ be a reduced word containg $a$. When the word $S(w) = S(x_1) a y S(x_2)$ or $S(w) = S(x_1) y a S(x_2)$ is reduced, the substring $ay$ or $ya$ will never completely cancel.*

**Proof** If $S(a) = ay$, $S(w) = S(x_1) a y S(x_2)$, to cancel any part of $ay$ from the left, $S(x_1)$ must end in $\overline{a}$. Since $\overline{a}$ comes from $y\overline{a}$ under $S$, it means that $x_1$ must end in $\overline{a}$. If so, then $w$ would contain $\overline{a}a$ and not be reduced. To cancel $ay$ from the right, requires $S(x_2)$ to begin with $\overline{y}\overline{a}$, which under $S$ comes from $\overline{a}$, again impling that $w$ was not reduced. Hence the string $ay$ will never completely cancel. It follows similarly for $S(a) = ya$ ■

**Definition 3.1** *An* **x-string**, *$x \in L_n$, is a substring of consecuative $x$s within a word.*

**Lemma 3.4** *Given an $x$-string of length $m$ in a reuced word $w$, the length of the $x$-string under a Type II automorphism which does not conjugate $x$ can not change by more than 1 and still remain an $x$-string.*

**Proof** Say

$$w = y_1 \underbrace{x \cdots x}_{m} y_2.$$

To increase the length of the $x$-string either $y_1 \to y_1 x$ or $\overline{x} y_1 x$ and/or $y_2 \to x y_2$ or $x y_2 \overline{x}$. However both $x$ and $\overline{x}$ can not bet distinguished letters so only one of these may happen. Hence the length change is at most 1. This follows similarly for a decrease in string length. Conjugation must be omitted because under the automorphism $T$, $x \to \overline{y}xy$, if

$$w = \ldots xy\underbrace{x\cdots x}_{m}\overline{y}x\ldots \text{ then } T(w) = \ldots\underbrace{xx\cdots xx}_{m+2}\ldots$$

which can have an $x$-string of any length. ■

## 3.2 Factorization of Automorphisms

**Lemma 3.5** *If $T$ is an automorphism, $\sigma$ is a cycle, and $w$ a word, then $T(\sigma(w)) = \sigma(T(w))$. (We always assume that we include all cyclic cancelations).*

**Proof** Since $\sigma$ only changes what letters begin and end a word, the inclusion of cyclic or "back" cancels mean that $\sigma$ does not affect what letters are adjacent. If $T \in I(F_n)$ the result is obvious. If $T \in II(F_n)$, then $\sigma$ is appropriately adjusted to account for the addition of extra letters. From there the result follows trivially. ■

**Notation** A special class of Type II automorphisms, call one-letter automorphisms, are of the form $(\{x_1, x_2\}, x_2)$ for convenience, these automorphisms will be represented by $(x_1 \odot x_2)$.

**Lemma 3.6** *If $S' = (x_1 \odot x), S = (x_2 \odot x)$, and $x_1, x_2 \notin \{x, \overline{x}\}$ then $S'S(w) = SS'(w)$ for any word $w$.*

**Proof** For $x_1 = x_2$ obvious. For $x_1 \neq \overline{x_2}$, then $S'$ and $S$ act independently. By Lemma 3.1 when we reduce the word will not affect anything, and by Lemma 3.5 we have $S'S(w) = SS'(w)$. For $x_1 = \overline{x_2}$ consider the word $w = w'x_1zx_2w''$ for some string $z$. Without loss of generality we can cancel all occurrances of $x_1x_2$ and $x_2x_1$ in $z$, and thus may assume $z$ contains neither $x_1$ nor $x_2$. Since we are only concerned with the image of $x_1$ and $x_2$ under the transformations, we may drop $w'$ and $w''$ to consider only $w = x_1zx_2$. $S'(w) = x_1xz\overline{x}x_2$, $S(w) = \overline{x}x_1zx_2x$ and $S'S(w) = \overline{x}x_1xz\overline{x}x_2x = SS'(w)$. ■

**Theorem 3.1** *Given any Type II automorphism $S = (A, x)$, $S$ can be broken down into a sequence $S_1 \cdots S_n$ of one-letter automorphisms, $S_i = (x_i \odot x)$, $x_i \in A \backslash \{x\}$. This factorization of $S$ is unique up to order.*

**Proof** $S = (A, x)$, for each $y \in A$ such that $\overline{y} \notin A$, $S$ acts on all the $y's$ in a word $w$ independently from all the other letters in $A$. Since cancelations may be performed at any time and these $S_i$ automorphisms commute, we do not need to worry about the order these automorphisms occur in, nor any reductions they might induce. So $S = (x_1 \odot x) \cdots (x_k \odot x)(A', x)$ where $A'$ consists of those pairs $y, \overline{y} \in A$. If $A' = \emptyset$ we're done. Otherwise it suffices to show that $(y, \overline{y}, x) = (y \odot x)(\overline{y} \odot x)$. Only $y$ and $\overline{y}$ are affected, and $y(y, \overline{y}, x) = \overline{x}yx = y(y \odot x)(\overline{y} \odot x)$. This holds similarly for $\overline{y}$ by symmetry. ■

**Lemma 3.7** *If $T \in I(F_n)$ and $S \in II(F_n)$ then $\exists T' \in I(F_n)$, $S' \in II(F_n)$, such that $TS = S'T'$ .*

**Proof** Since $S$ can be decomposed by Theorem 3.1, it suffices to show that $T(y \odot x) = (y' \odot x')T'$. Letting $y' = T(y), x' = T(x)$, and $T' = T$, yields the desired result. ■

**Lemma 3.8** *Applying the sequence of automorphisms*

$$(x \odot y)(y \odot z)(x \odot \overline{y})(y \odot \overline{z})$$

*to a word is equivalent to applying $(x \odot z)$ to the word.*

**Proof** Under the sequence, we have the following:

- $x \to xy \to xyz \to x\overline{y}yz \to x\overline{y}yz = xz$
- $y \to y \to yz \to yz \to y\overline{z}z = y$
- $z \to z \to z \to z \to z$

Inverses follow similarly. This is the definition of the action $(x \odot z)$ ■

**Remark 3.1** *From Lemma 3.8, we can easily apply the Type I automorphisms $T(x) = \overline{x}, T'(z) = \overline{z}$ to get the following identities:*

- $(x \odot y)(y \odot \overline{z})(z \odot \overline{y})(y \odot z) = (z \odot \overline{z})$
- $(\overline{x} \odot y)(y \odot z)(\overline{x} \odot \overline{y})(y \odot \overline{z}) = (\overline{x} \odot z)$

- $(\overline{x} \odot y)(y \odot \overline{z})(\overline{x} \odot \overline{y})(y \odot z) = (\overline{x} \odot \overline{z})$

**Theorem 3.2** *All Type II automorphisms of $F_n = F(x_1, x_2, \ldots, x_n)$ can be generated by a set of $4n$ one-letter automorphisms. Specifically, the following set is sufficient.*

$$\left\{ \begin{array}{c|l} (x_i \odot x_{i+1}), (x_i \odot \overline{x_{i+1}} & i = 1, \ldots, n \\ (\overline{x_i} \odot x_{i+1}), (\overline{x_i} \odot \overline{x_{i+1}}) & n+1 = 1 \end{array} \right\}$$

**Proof** By Theorem 3.1 it suffices to show that all one-letter automorphisms can be generated with this set. Using

$$(x_1 \odot x_2)(x_2 \odot x_3)(x_1 \odot \overline{x_2})(x_2 \odot \overline{x_3}) = (x_1 \odot x_3)$$

by Lemma 3.8 and Remark 3.1 gives $(x_1 \odot \overline{x_3}), (\overline{x_1} \odot x_3), (\overline{x_1} \odot \overline{x_3})$. We can proceed through the other generators on both $x_1$ and $\overline{x_1}$ to give all the $(x_1 \odot x_j)$ and $(\overline{x_1} \odot x_j)$ combinations. Continue similarly on the the other generators gives the remaining automorphisms. ■

**Corollary 3.1** *Given only the identities of Lemma 3.8 and Remark 3.1 the set in Theorem 3.2 is the smallest set necessary to produce $II(F_n)$*

**Proof** By construction the set must be symmetric in all generators and inverses. To use the identities, it is necessary to have both $(x_i \odot x_j)$ and $(x_i \odot \overline{x_j})$. So there must be at least two one-letter automorphisms for each generator and inverse. The group has rank $n$, and so there are $2n$ generators and inverses, giving a minimum of $2 * 2n = 4n$ one-letter automorphisms required. ■

When an automorphism is factored in to one-letter automorphisms, there is the potential that the length of a word may increase on intermediary steps. Investigation of that yields the following:

**Theorem 3.3** *Let $S = (A, x)$ such that $|S(w)| = |w|$ and $A' = L_n \setminus A$ . Then if $S = S_1 \cdots S_m$ with $S_i = (y_i \odot x)$, there will be no intermediary length change if $y_i.A = y_i.A'$ or $y_i.L_n = 2(y_i.x)$ for all $y_i \in A$.*

**Proof** To determine when factoring $S$ will not intermediary change the length of the word, it suffices to consider the case where $S = S_1 S_2$ with $S_1 = (A\backslash\{y\}, x), S_2 = (y \odot x)$. By Theorem 2.3, $\Delta S = A.A' - x.L_n$ For $S_1$ we get $\Delta S_1 = (A \setminus \{y\}).(A' \cup \{y\}) - x.L_n$. By properties of the . function, we get $\Delta S_1 = A.A' + y.A - y.A' - x.L_n = \Delta S + y.A - y.A' = y.A - y.A'$. Since we wish $\Delta S_2$ to be zero, and if $\Delta S_1 > 0$ then $\Delta S_2 > 0$, we must have $\Delta S_1 = 0$ giving the first possibility. For $\Delta S_2 = 0$ we have $\Delta S_2 = (\{x, y\}).(L_n \setminus \{x, y\}) - x.L_n$ which reduces to $y.L_n - y.x - y.x$. Set that equal to zero gives the second possibility. ■

# 4 Automorphisms on Words

This section explores equivalence connections between arbitrary R-CP words, and generalizes to minimal words were possible. Specifically the connection between a word $w$ and $aw$ is examined. The first section explores the transitivity of non equivalence between $w$ and $aw$. The second half considers equivalent words and the necessary number of automorphisms to move between equivalent words.

**Remark 4.1** *If $w$ is an R-CP word, the $aw$ is an R-CP word.*

**Proof** First we need that $aw$ is minimal. Assume $aw$ is not minimal. Since $w$ is in R-CP form, adding an $a$ can not create a trivial cancelation. So then there must $\exists\, S$ such that $|S(aw)| < |aw|$. If $a$ or $\overline{a}$ is distinguished we have $|S(aw)| = 1 + |S(w)|$, implying that $|S(w)| < |w|$. If $a$ is not distinguished, Then $S(a)$ may introduce a cancelation in $S(aw)$. If it does not, then $|S(aw)| = |S(a)| + |S(w)| < |aw|$ which implies that $|S(w)| < |w|$. If $S(a)$ introduces a cancelation, it must be a back cancel. (While conjugation results in a forward cancel, this cancel does not affect the counting process). This means that $S(w)$ also has a back cancel, since $w$ begins with $a$, and we similarly obtain $|S(w)| < |w|$, contradicting the minimality of $w$.

Now If $aw$ is not in R-CP form then by Lemma 4.2 $\exists\, T, \sigma$, such that $T \in I(F_n)$ and $\sigma$ is a cycle, with $T\sigma(aw)$ in R-CP form. $\sigma(aw)$ must bring one of the longest $x$-strings forward. Let $n$ be the length of the $a$-string beginning $w$. Then for $\sigma$ to change the order of $aw$, there must be an $x$-string of length $n+1$ or greater in $aw$. Since $w$ is in R-CP form, $aw$ cannot end in $a$, so the beginning $a$-string is of length $n+1$. Any other string of length

$\geq n+1$ must be entirely contained within $w$, whose longest $x$-string is of length $n$. Thus the longest $x$-string in $w$ is the beginning $a$-string, so $\sigma$ is the identity. So the only possible change on $aw$ is by $T$. $T$ must fix $a$ since R-CP words begin with $a$. $T(aw) = aT(w)$. If $T$ changes any letter in $aw$, it must be increasing the lexicographic position, but then since $aT(w)$ would have a higher lexicographic position than $aw$, it would mean $T(w)$ has a higher position than $w$, contradicting that $w$ is in R-CP form. Thus $aw$ must be in R-CP form. ■

## 4.1 Non-equivalence Results

**Theorem 4.1** *If $w, v$ are R-CP words and $T, S$ are respectively Type I and Type II automorphisms such that $TS(aw) = av$, then $w \sim v$ by $TS(w) = v$.*

**Proof** $TS(aw) = TS(a)TS(w) = av$. Since $v$ is an R-CP word, $v$ begins with $a$, thus in reduced form, $TS(a)TS(w)$ begins with $aa$. By Lemma 3.2, $S(a) = a$ or $S(a) = \overline{y}ay$, $y \notin \{a, \overline{a}\}$. If $S(a) = a$ then $TS(aw) = T(a)TS(aw) = av$, so $T(a) = a$. Hence $TS(aw) = aTS(w) = av$ which implies $TS(w) = v$ giving $w \sim v$. If $S(a) = \overline{y}ay$, we have $w = aw'$ and $v = av'$ which gives $TS(aw) = T(\overline{y}aay)TS(w') = aav'$. Since $T$ can not cancel anything and is not a cycle, the $\overline{y}$ must back cancel and $T(a) = a$. So $S(aw) = \overline{y}aayS(w') = aayS(w')_{-1}$ (where $_{-n}$ indicates the last $n$ letters have been canceled). Now $S(w) = \overline{y}ayS(w') = ayS(w')_{-1}$. This means $TS(aw) = T(aayS(w')_{-1}) = aT(ayS(w')_{-1}) = av$, which implies $v = T(ayS(w')_{-1}) = TS(w)$, giving $w \sim v$. ■

**Corollary 4.1** *If $w, v$ are R-CP words, $T, S$ Type I and II automorphisms respectively, such that $TS(w) \neq v$, then $TS(aw) \neq av$. This follows from the contrapositive of Theorem 4.1*

**Lemma 4.1** *Let $w, v$ be reduced words such that $|w| = |v|$ and $|aw| > |w|$. Let $T \in I(F_n)$, such that $T(w) \neq v$ then $T(aw) \neq av$.*

**Proof** By contradiction. Assume $T(a) = y$, so $T(aw) = yT(w)$ and $|T(aw)| = |w| + 1$. If $T(aw) = yT(w) = av$, the $a$ in $av$ cannot cancel, since if it did, $|av| < |T(aw)|$. So for $yT(w) = av$ we must have $T(w) = v$, contradiction. ■

**Lemma 4.2** *Any word $w'$ in an CP class, differs from the R-CP word $w$ of that class by a single Type I automorphism, $T$, and a cycle, $\sigma$.*

**Proof** By the definition of an R-CP class, all words in that class are related through Type I automorphisms and cycles. So $\exists\, T_1, \ldots, T_m, \sigma_1, \ldots, \sigma_m$ such that $T_1\sigma_1 \ldots T_m\sigma_m(w') = w$, $T_i \in I(F_n)$, $\sigma_j$ a cycle. Then by Lemma 3.5 we have $T_1 \ldots T_m\sigma_1 \ldots \sigma_n(w') = w$. Since $I(F_n)$ is a group, as are the cycles, this reduces to $T\sigma(w') = w$. ■

**Theorem 4.2** *If $w, v$ are R-CP words such that $|w| = |v|$, and $T, S$ are Type I and Type II automorphisms respectively, such that $S(w) \neq v$ then $TS(aw) \neq av$ for any $T$.*

**Proof** $S(a)$ can equal $a, ay, ya$, or $\overline{y}ay$ giving us four different cases to check.

Case 1a: $S(a) = a$, $\overline{a}$ not distinguished. $TS(aw) = T(aS(w))$. The $a$ can not back cancel since w is minimal and $\overline{a}$ is not distinguished. Thus by Lemma 3.3, $T(aS(w)) \neq av$ .

Case 1b: $S(a) = a$, $\overline{a}$ distinguished. Then if $a$ does not back cancel, we get Case 1a. If $a$ back cancels we have $w = aw', S(w) = aS(w') = S(w')_{-1}$ and $S(aw) = aaS(w') = aS(w')_{-1} = aS(w)$ and Lemma 3.2 yields the desired result.

Case 2: $S(a) = ay,\ y \notin \{a, \overline{a}\}$, so $TS(aw) = T(ayS(w))$. Since $w$ starts with $a$, by Lemmas 3.2 and 3.3 neither the the $y$ nor the $ay$ be canceled. Since $w$ is minimal, $|S(w)| \geq |v|$ and $T$ does not change the length. Thus we get $|T(ayS(w)| = |ayS(w)| = 2 + |S(w)| \geq 2 + |v| > |av|$ so $TS(aw) \neq av$.

Case 3: $S(a) = ya, y \notin \{a, \overline{a}\}$. So $TS(aw) = T(yaS(w))$. By Lemma 3.2 the $a$ will not forward cancel and by Lemma 3.3 the $ya$ will not back cancel. If the $y$ does not back cancel we get $|TS(aw)| > |av|$ the same as in Case 2. If the $y$ back cancels we have $w = aw'$, giving $S(w) = yaS(w') = aS(w')_{-1}$ and $TS(aw) = T(yayaS(w')) = T(ayaS(w')_{-1})$. Now $|S(w)| = |aS(w')_{-1}| \geq |v|$ and by lemmas 3.2 and 3.3 the addition of $ay$ to the front will not cancel, so $|TS(aw)| = |ayaS(w')_{-1}| > |av|$, hence $TS(aw) \neq av$.

Case 4: $T(a) = \overline{y}ay$. Let $w = aw'$. $S(w) = \overline{y}ayS(w'), S(aw) = \overline{y}aayS(w')$. If the $\overline{y}$ back cancels then we have $S(aw) = aS(w)$ and can apply Lemma 4.1 to get $TS(aw) \neq av$. If the $\overline{y}$ does not cancel we have $T(\overline{y}a) = aa$ in $av$ which cannot happen. Hence $TS(aw) \neq av$. ■

**Lemma 4.3** *If $w$ is an R-CP word, and $S \in II(F_n)$ such that $|S(w)| > |w|$ then $|S(aw)| > |aw|$.*

**Proof** Let $S = (A, x), andB = L_n \backslash A$. By Theorem 2.3 we know that $|S(w)| - |w| = A.B_w - L_n.x_w > 0$. If $x \notin \{a, \overline{a}\}$, then $|S(aw)| - |aw| = A.B_{aw} - L_n.x_{aw} = A.B_w + a.a - L_n.x_w = A.B_w - L_n.x_w > 0$. If $x \in \{a, \overline{a}\}$ then $S(a) = a$, so $|S(aw)| = |aS(w)| > 1 + |w| = |aw|$. ■

**Notation** A cycle that cycles the first $l$ letters of a word $w$ is denoted $\sigma^{(l)}$. Specifically, if $w = y_1 \cdots y_m$, $y_i \in L_n$ then $\sigma^{(l)} = y_{l+1} \cdots y_m y_1 \cdots y_l$.

**Theorem 4.3** *Let $w, v$ be minimal words of the same length such that $\sigma^{(k)}(w) \neq v$ for any $k$ . If $|aw| > |w|$ then $\sigma^{(k)}(aw) \neq av$ except when there $\exists\, i$ such that if $w = x_1 \cdots x_m$, $x_k \in L_n$, we have $x_{i+1} = a$ and $x_{i+2} \cdots x_n x_{i+1} x_1 \cdots x_i = v$*

**Proof** First note that if $|av| \neq |aw|$ the conclusion follows trivially, so we may assume $|aw| = |av|$. Let $\sigma^{(i)}(aw) = x_{i+1} \cdots x_m a x_1 \cdots x_i$. Since $\sigma^{(k)}(w) \neq v$ for all $k$, then let $j$ be the $j^{th}$ letter from the left in $\sigma^{(i+1)}(w)$ that differs from $v$. If $j < m - i$ then $\sigma^{(i)}(aw) \neq av$ , i.e.

$$\begin{array}{rcccccc} \sigma^{(i+1)}(w) & = & x_{i+2} & \cdots & x_m & x_1 & \cdots & x_{i+1} \\ v & = & v_1 & \cdots & v_{m-i-1} & v_{m-i} & \cdots & v_m \end{array}$$

$$\begin{array}{rcccccccc} \sigma^i(aw) & = & x_{i+1} & x_{i+2} & \cdots & x_m & a & x_1 & \cdots & x_i \\ av & = & a & v_1 & \cdots & v_{m-i-1} & v_{m-i} & v_{m-i+1} & \cdots & v_m \end{array}$$

if $j < m - i$ then the $j^{th}$ entry in $\sigma^{(i+1)}(w)$ corresponds to the $j + 1^{rst}$ entry in $\sigma^{(i)}(aw)$, which will differ from the $j + 1^{rst}$ entry of $av$.

If $j \geq m - i$ let $l$ be the first letter from the right in $\sigma^{(i)}(w)$ that differs from $v$. If $k \leq i$, $\sigma^{(i)}(aw) \neq av$ by the same reasoning as above. If $k > i$ and $j \geq m - i$ we get

$$\begin{array}{rcccccccc} \sigma^{(i)}(aw) & = & \boxed{x_{i+1}} & x_{i+2} & \cdots & x_m & \boxed{a} & x_1 & \cdots & x_i \\ av & = & \boxed{a} & v_1 & \cdots & v_{m-i-1} & \boxed{v_{m-i}} & v_{m-i+1} & \cdots & v_m \end{array}$$

where the boxes are around those elements which have not yet been shown to be equal. If $x_{i+1} = v_{m-i}$ then $x_{i+2} \cdots x_m x_{i+1} x_1 \cdots x_i = v$ and so if $x_{i+1} = a$

we get $\sigma^{(i)}(aw) = \sigma^{(i)}(x_{i+1}w) = av = x_{i+1}v$ as claimed. This analysis generalizes to the addition of letters other than $a$ and to positions other than the beginning of the word. ■

## 4.2 Automorphism Steps between Words

**Definition 4.1** *Given words $w, v$ such that $\exists\,\sigma$ a cycle, $T$ Type I automorphisms, and $S$, Type II automorphism, with $\sigma TS(w) = v$ then $w$ and $v$ are* **equivalent by a single (Type II) automorphism**, *denoted* $\mathbf{w} \overset{\mathbf{II}}{\sim} \mathbf{v}$.

**Theorem 4.4** *Given R-CP words $w, v, w \sim v$, then there exists R-CP words $w_1, \ldots w_n$, $w \sim w_i$ such that $w \overset{II}{\sim} w_1 \overset{II}{\sim} \ldots \overset{II}{\sim} w_n \overset{II}{\sim} v$.*

**Proof** $w \sim v$ so there exists Type II automorphisms $S_1, \ldots, S_m$, a Type I automorphism $T$ and cycle $\sigma$ such that $\sigma TS_1 \cdots S_m(w) = v$. Now $S_m(w) = w_1'$, and $w_1'$ is in some CP class, so by Lemma 4.2 there exists $\sigma_1, T_1$ such that $\sigma_1 T_1 S_m(w) = \sigma_1 T_1(w_1') = w_1$, so $w \overset{II}{\sim} w_1$. From $w_1$ we apply $T_1^{-1}\sigma_1^{-1}$ to get $w_1'$ to which we apply $S_{m-1}$. Thus $S_{m-1}T_1^{-1}\sigma_1^{-1}(w_1) = w_2'$ and similarly $\sigma_2 T_2 S_{m-1} T_1^{-1}\sigma_1^{-1}(w_1) = w_2$. By Lemma 3.5 we have $\sigma^* T_2 S_{m-1} T_1^{-1}(w_1) = w_2$ for some cycle $\sigma^*$ and by Lemma 3.7 this becomes $\sigma^* T^* S^*(w_1) = w_2$ for some Type I and Type II automorphisms $T^*, S^*$. Proceeding inductively yields the desired result. ■

**Corollary 4.2** *Theorem 4.4 holds similarly for equivalent non-R-CP words, through the addition of Type I automorphisms and cycles.*

**Theorem 4.5** *Suppose $w, v$ are R-CP words such that $w \sim v$ then there exists $S_1, \ldots, S_m \in II(F_n)$, $T \in I(F_n)$ and a cycle $\sigma$ such that $\sigma TS_m \cdots S_1(w) = v$ and $m \leq$ the number of R-CP words in the equivalence class of $w$.*

**Proof** Assume $m >$ number of R-CP words. Then $\exists\, i, k, i \neq k$ such that $S_i \cdots S_1(w) \overset{CP}{\sim} S_k \cdots S_1(w)$. With out loss of generality, $i > k$. Then $\exists\,\sigma', T'$ a cycle and Type I automorphism, such that $S_i \cdots S_1(w) = \sigma' T' S_k \cdots S_1(w)$. This means $\sigma TS_m \cdots S_{i+1}\sigma' T' S_k \cdots S_1(w) = v$. By Lemmas 3.5 and 3.7 we can factor the $\sigma', T'$ to the beginning and under closure and relabeling we get $\sigma TS_{m-i+k} \cdots S_1(w) = v$. Proceed inductively to $m \leq$ number of R-CP words. ■

**Corollary 4.3** *Theorem 4.5 holds similarly for $w \sim v$, $w, v$ not R-CP words, by converting them to R-CP words with cycles and Type I automorphisms, applying Theorem 4.5, and factoring the added cycles and Type I automorphisms to the initial ones.*

# 5 Specific Word Results

This section deals with results specific to certain words. If begins by providing the entire set of equivalent R-CP words for a word of the form $a \cdots abb$, the automorphisms that connect those R-CP words, and the minimum number of automorphisms possible to equate words in that R-CP class. Next the interconnections of words with length $\leq 6$ are explored with respect to one-letter automorphisms. Finally, the relationship between R-CP words in free groups of different orders is delineated.

## 5.1 Equivalence Class of $a \cdots abb$

**Theorem 5.1** *Given a free group of rank $\geq 2$ the word $a \cdots abb$ of length $m$ has exactly $\lfloor \frac{m}{2} \rfloor + 1$ R-CP classes, with the following as R-CP words.*

$$\underbrace{a \cdots a}_{m-2} bb,\ \underbrace{a \cdots a}_{m-3} b\overline{a}b,\ \ldots,\ \underbrace{a \cdots a}_{\lceil \frac{m}{2} \rceil} b \underbrace{\overline{a \cdots a}}_{\lfloor \frac{m}{2} \rfloor} b$$

**Proof** The one letter automorphisms $S = (b \odot \overline{a})$ will generate all the listed R-CP words by iteration on $a \cdots abb$ , and the listed words are all in R-CP form so there are no duplicates. The $\lfloor \frac{m}{2} \rfloor + 1$ time $S$ is applied it takes

$$\underbrace{a \cdots a}_{\lceil \frac{m}{2} \rceil} b \underbrace{\overline{a \cdots a}}_{\lfloor \frac{m}{2} \rfloor} b \text{ to } \underbrace{a \cdots a}_{\lceil \frac{m}{2} \rceil - 1} b \underbrace{\overline{a \cdots a}}_{\lfloor \frac{m}{2} \rfloor + 1} b$$

and $\lceil \frac{m}{2} \rceil - 1 \leq \lfloor \frac{m}{2} \rfloor + 1$, so it is not equivalent to a new R-CP word. This accounts for the $\lfloor \frac{m}{2} \rfloor + 1$ R-CP words claimed. It remains to be shown that no other R-CP words can be produced. First note that only automorphism using just $a$ *or* $\overline{a}$ and $b$ *or* $\overline{b}$ might produce another R-CP word. Any automorphism that as $a$ or $\overline{a}$ as a distinguished letter can only produced the R-CP words of the form listed. Thus it suffices to consider application of a single automorphism around $a$, and to show that all the giving R-CP words are increased on application of these automorphisms. The words are of the form

$$w = \underbrace{a \cdots a}_{k} b \underbrace{\overline{a \cdots a}}_{i} b, \ |w| = 2 + i + k$$

<u>Case 1:</u> Consider a transformation $S(a) = ay, y \in \{b, \overline{b}\}$ then

$$S(w) = \underbrace{ay \cdots ay}_{2k} b \underbrace{\overline{ya \cdots ya}}_{2i} b$$

Now $b$ will cancel either $y$ or $\overline{y}$, but no $a$ or $\overline{a}$ will cancel. So $|S(w)| = 2k + 2i + 2 - 2 = 2(k + i)$ and $2(k + i) > k + i + 2$ when $k + i > 2$, i.e. when $|w| > 2$.
<u>Case 2:</u> $T(a) = ya, y \in \{b, \overline{b}\}$ follows similarly.
<u>Case 3:</u> $T(a) = \overline{y}ay, y \in \{b, \overline{b}\}$. Then

$$T(w) = \underbrace{\overline{y}a \cdots ay}_{k+2} b \underbrace{\overline{ya} \cdots \overline{a}y}_{i+2} b$$

Now both of the $b$'s will cancel, but since we have $yb\overline{y}$ the cancel $b$ will be replaced by another $b$, thus leaving the word unchanged.
Words $|w| \leq 2$, are not of the form described and thus can be ignored. ■

**Corollary 5.1** *To get from*

$$\underbrace{a \cdots a}_{m-2} bb \text{ to } \underbrace{a \cdots a}_{\lceil \frac{m}{2} \rceil} b \underbrace{\overline{a \cdots a}}_{\lfloor \frac{m}{2} \rfloor} b$$

*it is necessary to use* $\lfloor \frac{m}{2} \rfloor$ *Type II automorphism.*

**Proof** We have already shown that automorphisms around $a$ or $\overline{a}$ either leave a word unchanged or increase the length. Any automorphism around $b$ or $\overline{b}$ either leaves a word unchanged, switch between steps,

$$\underbrace{a \cdots a}_{k} b \underbrace{\overline{a \cdots a}}_{i} b \leftrightarrow \underbrace{a \cdots a}_{k-1} b \underbrace{\overline{a \cdots a}}_{i+1} b$$

requires a cycle to switch between steps, or increase word length. Thus to get between the "end" words, it is necessary to go through all the steps. To get each $\overline{a}$ between the $bb$'s requires 1 automorphism. Since we wish to insert $\lfloor \frac{m}{2} \rfloor$ $\overline{a}$'s, $\lfloor \frac{m}{2} \rfloor$ automorphisms are required. ■

**Note** This indicates that when using Whitehead's Algorithm, there will be cases when at least $\lfloor \frac{m}{2} \rfloor$ Type II automorphism are necessary to get between some minimal words of length $m$.

## 5.2 R-CP words of length $\leq 6$

R-CP words of length $\leq 6$ have at most two generators. As Appendix II shows, nearly all of the R-CP words are the sole R-CP word in their equivalence class. With the exception of the $a \cdots abb$ class, the only equivalence class with more than one R-CP word is the R-CP words of length 6 with weight distribution $2-2-2$. So to determine the necessary set of Type II automorphisms that will connect any two equivalent R-CP words, it suffices to consider only those automorphism required for the class of weight $2-2-2$ and the addition of $\{(\overline{b} \odot a), (b \odot a)\}$ or $\{(\overline{b} \odot a), (\overline{b} \odot \overline{a})\}$ which are need for the $a \cdots abb$ class.

**Theorem 5.2** *To get from $w$ to $v$, where $|w|, |v| \leq 6$ and $w \sim v$, only cycles, Type I automorphisms, and 5 one-letter automorphisms are required. Any 5 automorphisms obtained by reading the chart in Appendix III from left to right will be a sufficient set. Moreover, this cannot be done with a smaller set of one-letter automorphisms.*

**Proof** It suffices to show that any of these groups of 5 automorphisms, with the assistance of cycles and Type I automorphisms, can connected all the minimal words weighted $2-2-2$. Appendix III contains directed graphs indicating how these R-CP words are connected by one-letter automorphisms. The computer programs in Appendix IV were used to calculate all the possible inter-word connections by a single one-letter automorphism. The chart was constructed through a heuristic method determining the minimal necessary automorphisms to provide a connected directed graph. ■

## 5.3 R-CP words and Arbitrary $F_n$

**Definition 5.1** *Let* $\mathbf{R_n}$ *be the set of all R-CP words in* $F_n$*, the free group of rank n with generators* $x_1, x_2, \ldots x_n$*. If* $k \leq n$*, then* $R_k$ *is the set of all R-CP words in,* $R_k$ *the free group of rank k with generators corresponding to* $x_1, \ldots x_k$ *in the generators of* $R_n$*.*

**Remark 5.1** *Let* $k \leq n$*. If* $w \in R_k$ *then* $w \in R_n$*.*

**Proof** Let $w \in R_k$. First, $w$ is minimal in $F_n$. If not then $\exists\, S \in II(F_n)$ such that $|S(w)| < |w|$. Since $w \in R_k$, only $x_1 \ldots x_n$ are possible in $w$, thus the distinguished letter of $S$ must be in $L_k$. So $S = (A, y), y \in L_k$. Since $w \in F_k$, $S(w) = S'(w)$, $S' = (A \cap F_k, y)$. Then $|S'(w)| < |w|$ and $S' \in II(F_k)$, contradicting $w \in R_k$. Similarly, assume $w$ is not in R-CP form in $F_n$. So $\exists \sigma$ cycle and $T \in I(F_n)$ such that $\sigma T(w)$ is in R-CP form in $F_n$. Since the generators $x_1 \ldots x_k$ of $F_k$ precede the remaining generators of $F_n$ and $T$ must preserve inverses, any arrangement $T$ that replaces a letter in $w$ by an element of $L_n \backslash L_k$ reduces the lexicographic order of $w$. Other wise $T \in I(F_k)$, contradicting that $w \in R_k$. So $T$ is the identity, hence if $\sigma$ is required, we would once again have $w \notin R_k$. ■

**Remark 5.2** *If* $w \in R_n$ *and the weight of* $x_{k+1}, \ldots x_n = 0$ *in* $w$*, then* $w \in R_k$ *for* $k \leq n$

**Proof** Minimality follows trivially from $II(F_k) \subseteq II(F_n)$. Similarly, R-CP form is trivial as $I(F_k) \subseteq I(F_n)$ and $\sigma_{F_k} = \sigma_{F_n}$. Thus there is a one to one correspondence between R-CP words in $F_k$ and R-CP words generated by $x_1, \ldots, x_k$ in $F_n$. ■

**Note:** All elements in $R_n$ that are not CP-equivalent to a word in $R_k$ contain at least $k+1$ generators. Let $v \overset{CP}{\sim} w, w \in R_k$ then $v \overset{CP}{\sim} w \in R_n$. The remaining words equivalent to $w \in R_n$ are obtained by applying $I(F_n) \backslash I(F_k)$ to the words CP-equivalent to $w \in R_k$.

**Remark 5.3** *If* $w, v$ *are R-CP words with weights* $x_{k+1}, \ldots x_n = 0$ *and* $w \sim v \in F_n$ *then* $w \sim v \in F_k, k \leq n$*.*

**Proof** By Whitehead's theorem, we have that $\exists\, \sigma$, $T \in I(F_n), S_1, \ldots S_m$, $S_i \in II(F_n)$ such that $\sigma T S_1 \cdots S_m(w) = v$ and $|T_i \cdots T_m(w)| = |w|$ for all $i$. Since the length is constant we know there are not distinguished letters in $L_n \backslash L_k$. As before, we can replace each $S_i = (A, y)$ with $S_i' = (A \cap L_k, y) \in II(F_k)$. Hence no elements in $L_n \backslash L_k$ have an opportunity to appear until $T$. Since by Remark 5.2 $v \in R_n \Rightarrow v \in R_k$, $T$ may be replaced by $T' \in I(F_k)$, thus we get $w \sim v$ in $F_k$. ■

# 6 Comments, Conjectures, & Counterexamples

This section provides a springboard for further research ideas in dealing with equivalent words. Initially, the notion of an isolated word is introduced, and basic properties explained. Conjectures based on partial data analysis are presented, and, where known, counterexamples to apparent patterns are given. Then a method of counting cyclic words not containing inverses is explored and its connection to the number of R-CP words in an equivalence class is conjecture. Finally, a trend occurring in equivalence classes is noted and explained.

## 6.1 Isolated R-CP Words

**Definition 6.1** *A word is* **isolated** *if there is exactly 1 CP class in its equivalence class. So a single isolated word in an CP class means the entire class is isolated.*

**Example 6.1** *The word* a *is isolated, since any automorphism which does not increase the size simply takes* a *to a different generator, which can be achieved by a Type I automorphism. Simple inspection shows that the word* aa $\cdots$ aa *of any length is always isolated.*

**Lemma 6.1** *If $w$ is an isolated R-CP word such that $|S(w)| = |w|, S \in II(F_n)$ and $S(w)$ does not require a cycle to be put in R-CP form, i.e. $\exists T \in I(F_n)$ such that $TS(w) = w$, then $S(aw)$ does not need a cycle to be put in R-CP form.*

**Proof** If $S(aw)$ requires a cycle it means there either an $x$-string in $S(aw)$ of equal or longer length than the beginning $a$-string, or a cycle is needed to bring the initial $a$-string to the front. There can't be a string longer than the $a$-string because then $S(w)$ would have a string longer than the starting string and then needs a cycle. Similarly since both $aw$ and $w$ begin with $a$ if $S(aw)$ need a cycle to put the beginning $a$-string in the front, so would $S(w)$. ∎

**Conjecture 6.1** *Words of length $m$ and weights $\lceil \frac{m}{2} \rceil - \lfloor \frac{m}{2} \rfloor$ are isolated R-CP words when $m > 4$.*

This has been observed to hold for all words of length $\leq 8$, as shown in Appendix II. The reason $m > 4$ occurs, is because at $m = 4$, the words with weight $2 - 2$ can be isolated or equivalent to $aabb$. However, it is trivially true for $m < 3$.

**Conjecture 6.2** *If $w$ is and isolated word, $w \not\sim a \cdots a$, then $w$ has exactly two generators.*

All isolated words of length $\leq 7$ have been contained in $R_2$ as indicated in Appendix II.

**Observation and Counter-Example** For words of length $\leq 7$ it was observed that if $w$ was an isolated R-CP word, then $aw$ was an isolated R-CP word. Similarly if $w$ was an isolated R-CP word ending with $x$, then $wx$ was an isolated word, though not necessarily in R-CP form. It was conjecture that this always holds, however a set of counter examples was constructed. The words $a^l bab\overline{a}^k \overline{b} a^n \overline{b}\overline{b}\overline{a}^k b$ for $l \geq n > k$ is isolated when $l = n + 1$, but in other cases the automorphism $(b \odot \overline{a})$ yields $a^{l-1} bb\overline{a}^k \overline{b} a^{n+1} \overline{b} a \overline{b} \overline{a}^k$, which is not CP-equivalent to the original word unless $l = n+1$. It is likely that a similar set of counter examples is constructible for the $wx$ conjecture. However, we do have the following result:

**Fact** Let $w$ be and isolated R-CP word such that $S(w)$ does not need a cycle to be put in R-CP form for all $S \in II(F_n)$ where $|S(w)| = |w|$. Then $aw$ is an isolated R-CP word.

**Proof** Assume $aw$ is not isolated, then there exists $S \in II(F_n)$ such that $S(w) \overset{CP}{\not\sim} w$. By Lemma 4.3, $|S(w)| = |w|$, and by lemma 6.1 no cycles will be necessary of R-CP form. If $S(a) = a$ or $S(a) = xa$, or $S(a) = xa\overline{x}$ where $x$ back cancels occurring then $T(a) = a$, giving $TS(aw) = aTS(w) = aw$ which implies that $w \overset{CP}{\sim} S(w)$. If $S(a) = ax$ or $S(a) = xa$ with no back cancels occurring, then $|S(aw)| = |xaS(w)| = 2 + |w| > |aw|$ and similarly for $S(a) = ax$. If $S(a) = xa\overline{x}$ with no back cancels, then $S(w)$ requires a cycle to be put in R-CP form, contradicting the contrapositive of Lemma 6.1. If $S(a) = a$ and the $a$ back cancels, we get $TS(aw) = TS(aaw') = T(aaS(w')) = T(aS(w')_{-1}) = aTS(w')_{-1} = aw$. Hence $aw$ is isolated. ■

## 6.2 Estimation on Number of R-CP words

**Theorem 6.1** *Let $\bigcirc_{n,\{q_i\}}$ be the number of cyclic words of length $n$ with the weight of the generator $x_i = q_i$. Let $\sharp_{l,n,\{r_i\}}$ be the number of linear words $w$ of length $n$ and weights $\{r_i\}$ such that $l$ is the smallest positive integer where $\sigma^{(l)}(w) = w$. Then*

- $\bigcirc_{n,\{q_i\}} = \frac{n!}{\prod_i q_i!} - \sum_{k=2,k|n}^{n}(k-1)\sharp_{k,n,\{q_i\}}$
- $\sharp_{l,n\{r_i\}} = \sharp_{l,l,\{\frac{r_i l}{n}\}}/l$ *for* $l \neq n$.
- $(n, q_i) = 1\forall\, i$, *then* $\bigcirc_{n,\{q_i\}} = \frac{(n-1)!}{\prod_i q_i!}$
- *Let* $d = gcd_i(q_i)$, *then* $\sharp_{n,n,\{q_i\}} = \frac{n!}{\prod_i q_i!} - \sum_{k=1,k|d}^{d-1} \sharp_{\frac{kn}{d},n,\{q_i\}}$

**Proof** $\frac{n!}{\prod_i q_i}$ represents the number of linear combinations of $n$ things with weights $\{q_i\}$. This over counts the number of cyclic words, so we need to subtract off a factor. If a word has no cycles (i.e. $\nexists i \neq n$ such that $\sigma^{(i)}(w) = w$) then all $n$ possibilities will appear on the linear list. To only count 1 of these, we need to subtract $(n-1)$ for each non-cyclic word. Similarly if the first cycle to occur in the word is at $k$ (i.e $\sigma^{(k)}(w) = w$), there will be $k$ distinct linear words produced by $\sigma^{(i)}(w), i = 1 \ldots k$. Thus we will need to subtract $(k-1)$ for each $k$ cycle word. This accounts for the $-\sum_{k=2}^{n}(k-1)\sharp_{l,n,\{q_i\}}$. Note that a word is counted by the first occurring $k$ cycle.. This eliminates the possibility of miscounting words containing more than one cycle. Using the first $k$ cycle suffices because if $\sigma_k(w) = w$ then $w = z \cdots z$ of length $n$ where $z$ is a string of length $k$. A cycle of a number larger than $k$ can not produce more that $k$ distinct linear words because of this. Hence it suffices to consider only $k$ such that $k|n$, which yields $\bigcirc_{n,\{q_i\}} = \frac{n!}{\prod_i q_i!} - \sum_{k=2,k|n}^{n}(k-1)\sharp_{k,n,\{q_i\}}$

Now $\sharp_{l,n,\{r_i\}}$ represents, by definition the number of distinct cyclic worlds of length $n$ with smallest cycle length $l$ and weights $\{r_i\}$. If $A$ is the set of all such words, $w \in A \Leftrightarrow w = z\cdots z, |w| = n, |z| = l, z$ has weights $r_i = \frac{q_i l}{n}$ and $z$ has no cycles of length $< l$. We know these weights, because $\frac{r_i n}{l} = q_i$ by definition. Furthermore, if $z$ has a cycle $< l$ then $w$ would have a cycle $< l$ and not be in $A$. Since each $z$ is counted as a linear word, $\sharp_{l,n,\{q_i\}} = \frac{\#A}{l}$ because cycles on $z$ do not differentiate $w$ cyclically. Thus we just need to calculate $^{\#}A$ which by definition is $\sharp_{l,l,\{\frac{q_i l}{n}\}}$, as desired.

If $(q_i, n) = 1 \forall i$ then for $l < n$ $\frac{q_i l}{n} \notin \mathcal{N}$, hence the number is 0. This means there are no inner cycles possible with the letter weights. Thus all cyclic words produce exactly $n$ linear words, giving a total of $\frac{n!}{n\prod_i q_i!} = \frac{(n-1)!}{\prod_i q_i!}$ distinct words.

For $\sharp_{n,n\{q_i\}}$ we know the number of linear words possible is $\frac{n!}{\prod_i q_i!}$. It is then necessary to subtract off all linear words bearing cycles. Since a cycle must equally distribute the weights, the $gcd_i(q_i) = d$ is the maximum number of partitions the $q_i$'s can be separated into. This gives cycles of length $\frac{n}{d}$. Since this is the smallest possible cycle length we also need to remove words withy larger minimum cycles, namely, those of length $\frac{kn}{d}$. However since these cycle lengths must divide $n$, we have $k|d$. Subtracting off these words, we are left with the number of linear words containing no cycles, as desired. ■

These formulas were developed only for estimation purposes. It is likely that the formulas may be simplified. Moreover, a computer program could easily be produced to facilitate this counting procedure on words of long lengths.

**Conjecture 6.3** *The number of R-CP words in a CP class having weights* $\{q_i\}$ *is* $\leq \bigcirc_{n,\{q_i\}}$.

This has been observed for all words of length $\leq 7$ and for words of length 8 on 2 generators. A chart of comparison between these quantities is presented in Appendix V.

**Conjecture 6.4** *The number of CP-classes (and the number of equivalence classes) is monotonic increasing, likely strictly increasing after* $|w| = 3$*.*

**Remark 6.1** *Although we don't yet have a formula for the number of R-CP words, it is possible to count the number of words that are CP-equivalent to a given R-CP word. For* $w \in F_n$*, let* $k$ *be the number of generators in* $w$*. If* $s$ *is the number of rotational symmetries in* $w$ *with respect to the generators, then the number of words CP-equivalent to* $w$ *is* $2n(2n-2)\cdots(2n-2(c-1))\frac{|w|}{s}$ *.*

**Proof** Since we wish to count all CP-equivalent words, it is necessary to determine how many words are equivalent through Type I automorphisms. Specifically, how many ways can we interchange the generators. This is obtained by $2n(2n-2)\cdots(2n-2(c-1))$. Multiplication by $|w|$ counts all possible cycles. However, this over counts the number of words when $w$ has symmetry. To account for this we divide by the number of symmetries. Symmetries are calculated with respect to the generators, for instance *aabb* has two lines of symmetry: the identity and and one between $a$ and $b$. Specifically, this method of counting symmetries equates *aabb* with *bbaa*. Only the relative differentiation between the positioning of generators and inverses is considered.

**Note:** Symmetries are not constant within an equivalence class; equivalent R-CP words may have different symmetry values. However, the number of symmetries will always divide the length of the word.

## 6.3 Patterns within CP-Classes

**Definition 6.2** *Two words* $w, v$ *differ by a* **transposition** *if they are the same except for a substring* $xy$ *in* $w$ *which has be replaced by* $\overline{y}x$ *or* $y\overline{x}$ *in* $v$*. For example* $aaabbcc$ *differs from* $aaabc\overline{b}c$ *by a transposition.*

**Observation** In the list of equivalent R-CP words, there a many groups of words that appear to differ only by a transposition. Since a transposition in a word does not guarantee that the transposed word will be in R-CP form, words may "differ" by a transposition without the transposition being apparent in R-CP form. A transposition between words will occur when a Type II one-letter automorphism , $S$, such that $S(y) = y$ and $S$ maintains word length is applied as follows:

- If $S(x) = x\overline{y}$ and we have the substring $xxy$ in $w$ and for a single transposition there are no other occurrances of $x$ in $w$. The string $xxy$ goes to $x\overline{y}x$.
- If $S(x) = \overline{y}x$ the string $yxy$in $w$ will go to $x\overline{y}x$.
- If $S(x) = x\overline{y}$ the string $xyx$ in $w$ will go to $xx\overline{y}$.
- If $S(x) = \overline{y}x$ the string $xyx$ in $w$ will go to $\overline{y}xx$.

Note that conjugation does not allow transpositions. There may be other automorphisms that induce multiple transpositions in a word, or affect letters other than those involved in the transposition.

# 7 Connections to Other Fields

In addition to an algebraic interest, the equivalence of free group words arises in the topological study of tori. Curves on a torus can be represented as words in their fundamental group. For instance, single hole torus that is missing a point has $F_2$ as its fundamental group. Classification of equivalent words in $F_2$ allows for easy determination of homotopic curves on the torus. Classification of these homotopic curves can also provide insight into the Markov Spectrum in number theory. The curves were not explored in development of this paper, however the list of equivalent words in Appendix II could easily be translated to homotopic curves on a torus minus a point. In addition, Appendix VI lists some homotopic classes on a complete one-hole and two-hole torus using the relation that $xy\overline{xy} = 1$, which occurs when the missing point is replaced, and allows some minimal words to reduce down.

# 8 Conclusion

This paper developed from ideas and data presented in Lau's work [2]. Initial work was made in extending Lau's classification from words of length $\leq 6$ to words of length $\leq 8$ through the use of Maple programs. (See Appendix II and IV). Various patterns and facts about these equivalence classes were noted and explored, while the classification tables have also been adapted for use in topology. (See Appendix VI)

Besides derivations from data accumulation, the properties of Whitehead Type II automorphisms have been studied. How and when cancelations occur was determined, which provides the basis for further studies. In addition a method of factorization of an automorphisms into one-letter automorphisms and a sufficient generating set one-letter automorphisms is provided.

The effects of these automorphism on R-CP words was investigated, and a number of cases of transitivity of non-equivalence was determined. Moreover, both an upper and lower bound on the number of automorphisms between two arbitrary equivalent words is given. Results pertaining to certain classes of words is also provided.

Along with these results, areas of potential investigation have been noted. A method for counting circular words is developed and possible connections to the number of R-CP words indicated. The concept of isolated words was developed and initial results, along with further conjectures presented.

In all, this paper provides the ground work for research on Whitehead automorphisms. It also begins the investigation of what the equivalence of words implies about the equivalence of other words. Specific patterns and trends have been noted, and many conjectures on which to base further research are given. This ground work and conjectures are only the beginning of equivalence exploration and indicate that Whitehead equivalence provides a rich field for further study.

# Appendix I

**Whitehead's Algorithm**
Suppose $w, v \in F = F_n$.

1. Find minimum length words $w^*$ and $v^*$ equivalent to $w$ and to $v$, respectively. Since there are only finitely many Whitehead automorphisms of $F_n$, this may be done as follows:

   Check to see if $w$ or $v$ is equivalent , under a single Whitehead automorphism, to a word of strictly lesser length. If one of them is, then replace it with this shorter word and repeat this process. If not, then $w$ and $v$ are of minimum length.

2. Make a list of all the minimum length words equivalent to $w$ by executing the following steps:

   (a) Using the minimum length word $w^* \sim w$ found in Step 1, make a list consisting of $w^*$ and all the same-length words equivalent to $w^*$ under a single Whitehead automorphism.

   (b) Make a new list by appending (to the old list created in Step 2(i)) all the same-length words equivalent to a word on the old list via a single Whitehead automorphism. If this new list contains no words not already found on the old list, then clearly there are no other minimum length words that are equivalent to $w$ under a finite sequence of Whitehead automorphisms. If, on the contrary, it contains new words, then reapply this step, using the "new list" in place of the "old list". Note that since there are only finitely many words of a given length in $F_n$, Step 2 must eventually terminate.

3. If $v^*$ is on the list produced in Step 2, then clearly $w \sim v$. Otherwise, it follows directly from Whitehead's Theorem that $w \not\sim v$.

# Appendix II

**Classification of Words of Length** $\leq 8$ These is the set of all R-CP words in $F_2$ of length $\leq 8$ The classification results for words of length $\leq 6$ is due to Lau [2]

| Length | Class | R-CP elements |
|---|---|---|
| 1 | 1 | $a$ |
| 2 | 1 | $aa$ |
| 3 | 1 | $aaa$ |
| 4 | 1 | $aaaa$ |
| | 2 | $ab\bar{a}\bar{b}$ |
| | 3 | $aabb$<br>$ab\bar{a}b$ |
| 5 | 1 | $aaaaa$ |
| | 2 | $aaba\bar{b}$ |
| | 3 | $aab\bar{a}\bar{b}$ |
| | 4 | $aaabb$<br>$aab\bar{a}b$ |
| 6 | 1 | $aaaaaa$ |
| | 2 | $aaabbb$ |
| | 3 | $aabaa\bar{b}$ |
| | 4 | $aab\bar{a}\bar{a}\bar{b}$ |
| | 5 | $aaaba\bar{b}$ |
| | 6 | $aaab\bar{a}\bar{b}$ |
| | 7 | $aabba\bar{b}$ |
| | 8 | $aaba\overline{bb}$ |
| 6 | 9 | $aabb\bar{a}\bar{b}$ |
| | 10 | $aaaabb$<br>$aaab\bar{a}b$<br>$aab\bar{a}\bar{a}b$ |
| 7 | 1 | $aaaaaaa$ |
| | 2 | $aaaaba\bar{b}$ |
| | 3 | $aaaabbb$ |
| | 4 | $aaaab\overline{ab}$ |
| | 5 | $aaabaa\bar{b}$ |
| | 6 | $aaabba\bar{b}$ |
| | 7 | $aaabb\bar{a}b$ |
| | 8 | $aaabb\overline{ab}$ |
| | 9 | $aaab\bar{a}bb$ |
| | 10 | $aaab\overline{aab}$ |
| | 11 | $aaab\overline{abb}$ |
| | 12 | $aabbaa\overline{bb}$ |
| | 13 | $aabba\overline{bb}$ |
| | 14 | $aabb\overline{aab}$ |
| | 15 | $aaaaabb$<br>$aaaab\bar{a}b$<br>$aaab\bar{a}\bar{a}b$ |

| Length | Class | R-CP elements |
|---|---|---|
| 8 | 1 | $aaaaaaaa$ |
| | 2 | $aaaaaba\overline{b}$ |
| | 3 | $aaaabbb$ |
| | 4 | $aaaab\overline{ab}$ |
| | 5 | $aaaabaa\overline{b}$ |
| | 6 | $aaaaba\overline{bb}$ |
| | 7 | $aaaabba\overline{b}$ |
| | 8 | $aaaabbbb$ |
| | 9 | $aaaabb\overline{a}b$ |
| | 10 | $aaaabb\overline{ab}$ |
| | 11 | $aaaab\overline{a}bb$ |
| | 12 | $aaaab\overline{aab}$ |
| | 13 | $aaaab\overline{abb}$ |
| | 14 | $aaabaa\overline{bb}$ |
| | 15 | $aaaba\overline{bbb}$ |
| | 16 | $aaabbaa\overline{b}$ |
| | 17 | $aaabba\overline{bb}$ |
| | 18 | $aaabbba\overline{b}$ |
| | 19 | $aaabbb\overline{ab}$ |
| | 20 | $aaabb\overline{a}bb$ |
| | 21 | $aaabb\overline{aa}b$ |
| | 22 | $aaabb\overline{aab}$ |
| | 23 | $aaabb\overline{abb}$ |
| | 24 | $aaab\overline{aa}bb$ |
| | 25 | $aaab\overline{aaab}$ |
| | 26 | $aaab\overline{aabb}$ |
| | 27 | $aabbaa\overline{bb}$ |
| | 28 | $aabb\overline{aabb}$ |
| | 29 | $aabb\overline{abab}$ |
| 8 | 30 | $ab\overline{ab}ab\overline{ab}$ |
| | 31 | $aabab\overline{a}bb$<br>$aab\overline{a}bba\overline{b}$ |
| | 32 | $aaba\overline{b}a\overline{bb}$<br>$aaba\overline{bbab}$ |
| | 33 | $aab\overline{abba}b$<br>$aba\overline{ba}b\overline{ab}$ |
| | 34 | $aabbaabb$<br>$aba\overline{b}aba\overline{b}$ |
| | 35 | $aabba\overline{ba}b$<br>$aaba\overline{b}ab\overline{ab}$ |
| | 36 | $aaabbabb$<br>$aaba\overline{bb}ab$<br>$ababa\overline{b}a\overline{b}$ |
| | 37 | $aababba\overline{b}$<br>$aababb\overline{a}b$<br>$aaba\overline{babb}$ |
| | 38 | $aabaab\overline{ab}$<br>$aabab\overline{abb}$<br>$aabb\overline{ab}a\overline{b}$ |
| | 39 | $aaaaaabb$<br>$aaaaab\overline{a}b$<br>$aaaab\overline{aa}b$<br>$aaab\overline{aaa}b$ |
| | 40 | $aaba\overline{ba}bb$<br>$aaba\overline{bba}b$<br>$aabb\overline{ab}ab$<br>$aab\overline{a}bb\overline{ab}$<br>$abab\overline{a}b\overline{ab}$ |
| | 41 | $aaba\overline{b}abb$<br>$aabba\overline{b}ab$<br>$aabba\overline{bab}$<br>$aabb\overline{aba}b$<br>$abab\overline{a}ba\overline{b}$ |

| Length | Class | R-CP elements |
|---|---|---|
| 6 | 1 | $aabbcc$ |
| | | $aabcb\overline{c}$ |
| | | $aabccb$ |
| | | $aabc\overline{b}c$ |
| | | $aabc\overline{b}\overline{c}$ |
| | | $abacbc$ |
| | | $abacb\overline{c}$ |
| | | $abac\overline{b}\overline{c}$ |
| | | $abca\overline{b}\overline{c}$ |
| 7 | 1 | $abac\overline{acb}$ |
| | | $aba\overline{b}c\overline{ac}$ |
| | | $abca\overline{cab}$ |
| | 2 | $aaabbcc$ |
| | | $aaabcb\overline{c}$ |
| | | $aaabccb$ |
| | | $aaabc\overline{b}c$ |
| | | $aaabc\overline{bc}$ |
| | | $aabacb\overline{c}$ |
| | | $aabac\overline{bc}$ |
| | | $aabbacc$ |
| | | $aabbc\overline{a}c$ |
| | | $aabbc\overline{b}c$ |
| | | $aabcacb$ |
| | | $aabca\overline{bc}$ |
| | | $aabcb\overline{ac}$ |
| | | $aabccb\overline{c}$ |
| | | $aabcc\overline{a}b$ |
| | | $aabc\overline{a}b\overline{c}$ |
| | | $aabc\overline{ab}c$ |
| | | $aabc\overline{b}ac$ |
| | | $aabc\overline{b}a\overline{c}$ |
| | | $aabc\overline{b}cb$ |
| | | $aab\overline{a}c\overline{b}c$ |
| | | $abaca\overline{bc}$ |
| | | $abacbc\overline{b}$ |
| | | $abacb\overline{ac}$ |
| | | $abac\overline{a}b\overline{c}$ |



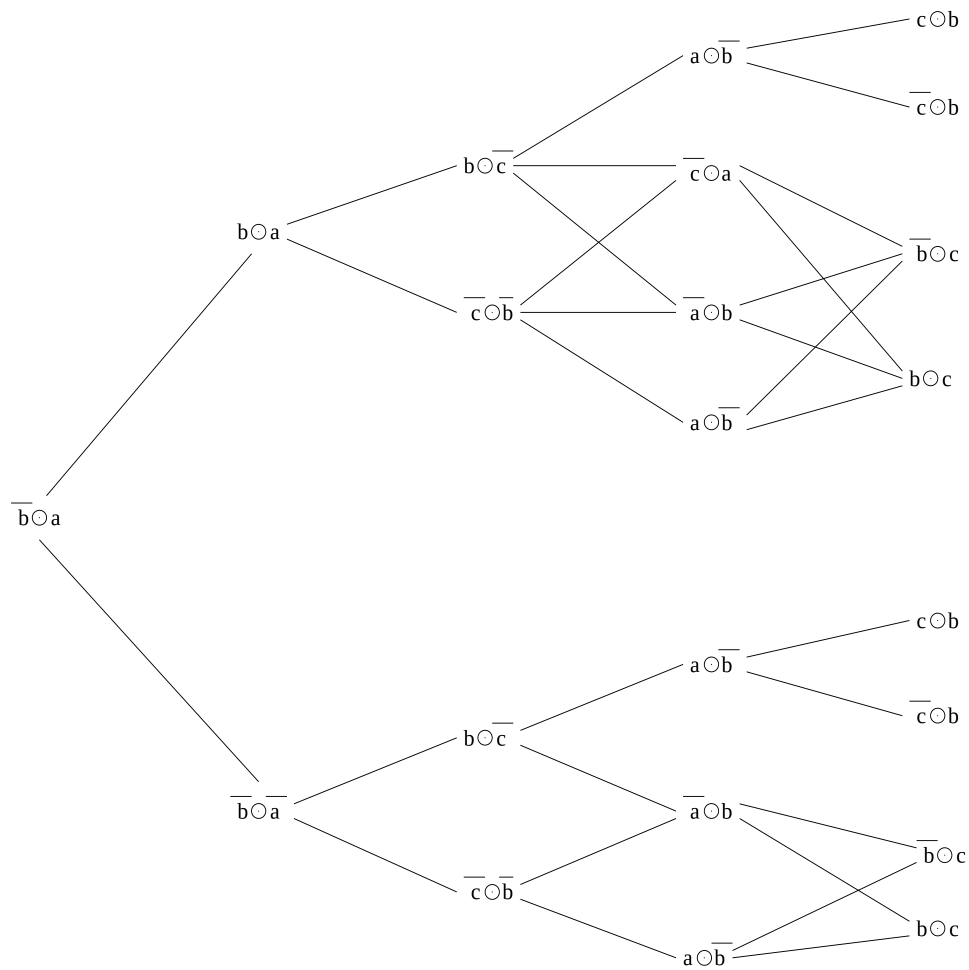
These tables contain the R-CP words of length $\leq 8$ on two generators. By lemma 2.1 these words must be at least of length 6.

| Length | Class | R-CP elements |
|---|---|---|
| 7 | 3 | $aabacbc$ |
| | | $aabacc\overline{b}$ |
| | | $aaba\overline{b}cc$ |
| | | $aabbca\overline{c}$ |
| | | $aabbc\overline{ac}$ |
| | | $aabbc\overline{bc}$ |
| | | $aabcac\overline{b}$ |
| | | $aabca\overline{c}b$ |
| | | $aabcbac$ |
| | | $aabcbb\overline{c}$ |
| | | $aabcb\overline{a}c$ |
| | | $aabcb\overline{c}b$ |
| | | $aabcc\overline{ab}$ |
| | | $aabcc\overline{bc}$ |
| | | $aabc\overline{bc}b$ |
| | | $aab\overline{a}cbc$ |
| | | $aabc\overline{a}c\overline{b}$ |
| | | $aabc\overline{ac}b$ |
| | | $abaca\overline{cb}$ |
| | | $abacb\overline{a}c$ |
| | | $abacb\overline{cb}$ |
| | | $abac\overline{a}bc$ |
| | | $abac\overline{a}c\overline{b}$ |
| | | $aba\overline{b}acb\overline{c}$ |
| | | $aba\overline{b}c\overline{a}c$ |

# Appendix III

**One Letter Automorphism on R-CP words of length 6**

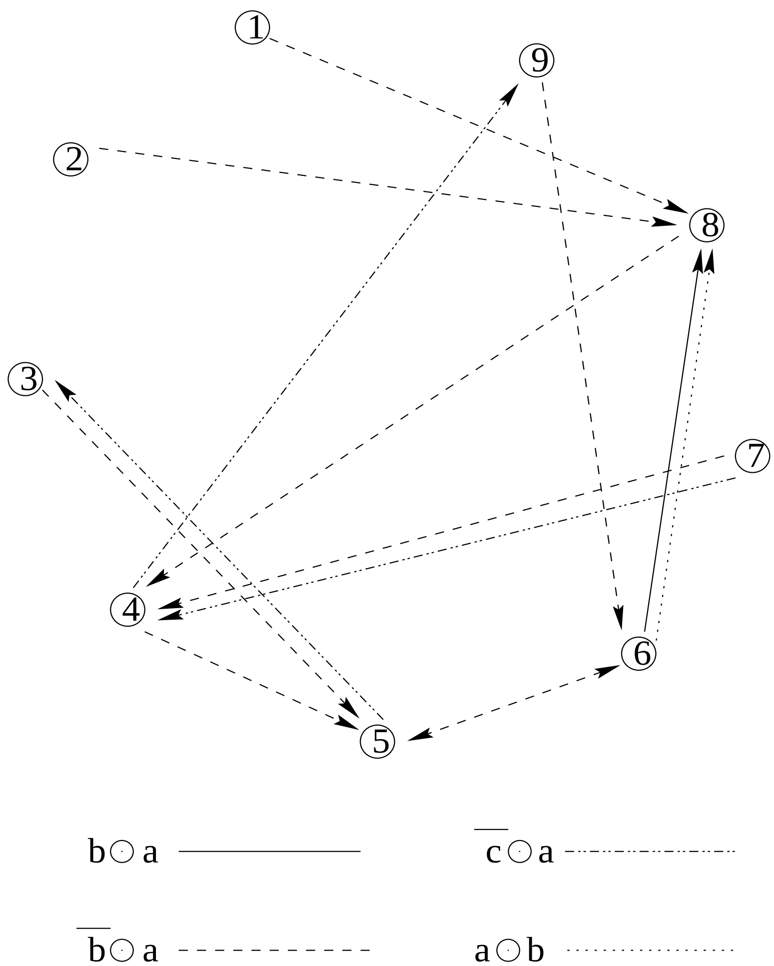
This chart gives all possible collections of five one-letter automorphisms that are necessary to connect any two equivalent words of length ≤ 6 Read the chart from left to right to obtain a set of five automorphisms.

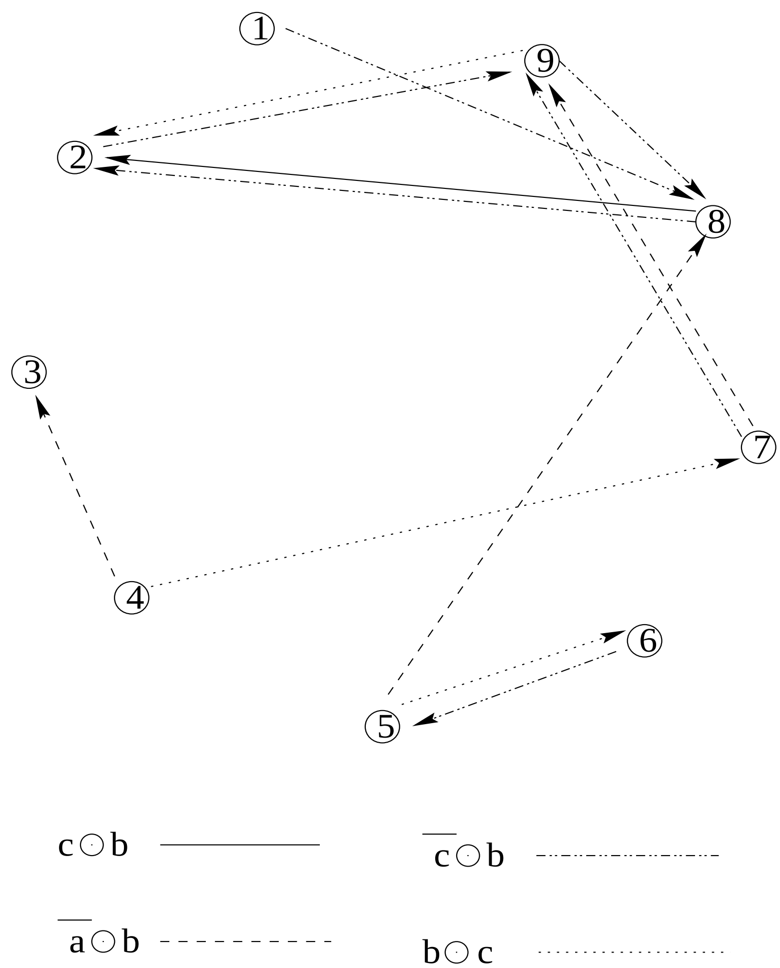


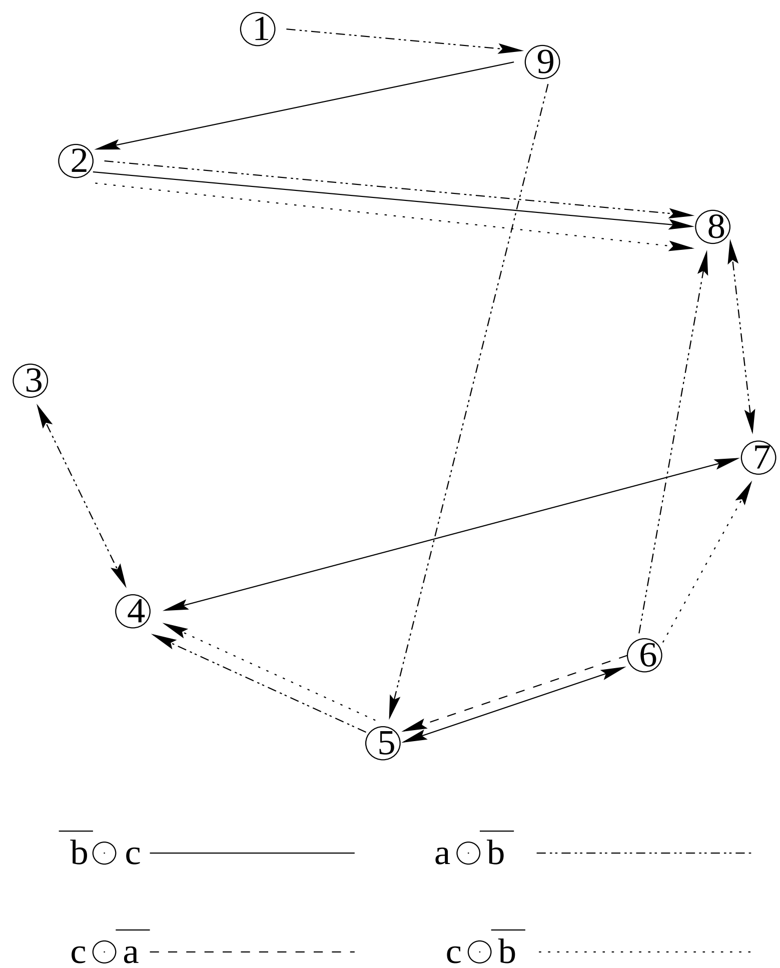
These graphs represent the one-letter automorphisms on the equivalence class with weight $2-2-2$. An arrow between words indicates that when applying the designated transformation to the word at the base of the arrow, the resulting word is CP-equivalent to the word at the tip of the arrow. One-letter automorphisms that increase word length are not included.

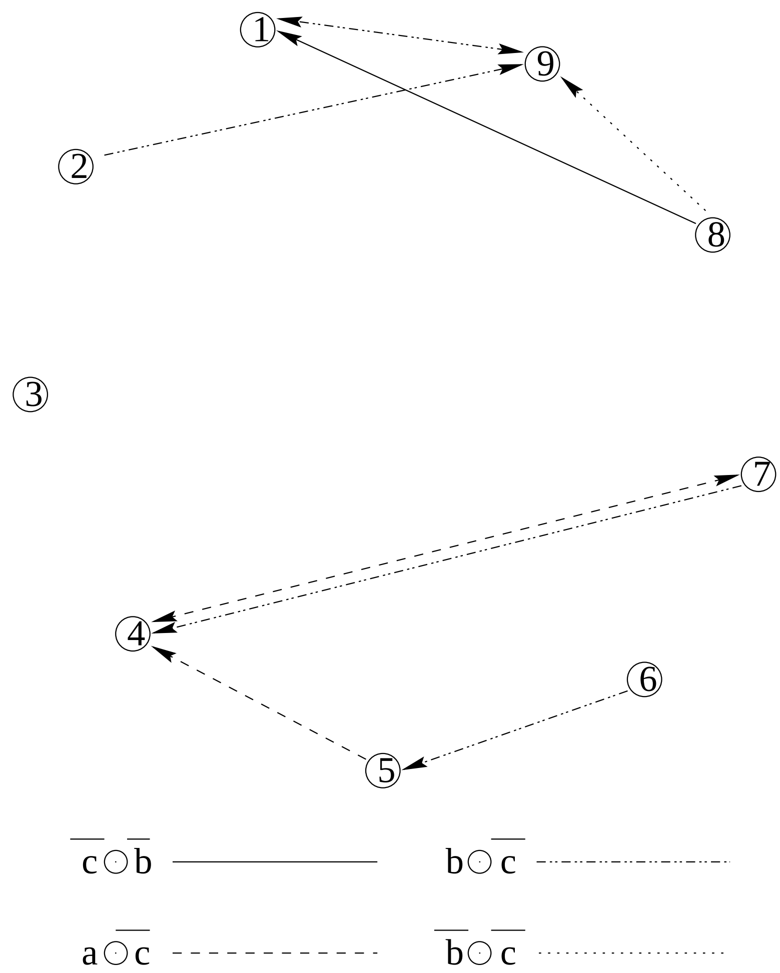
The numbering on the graphs corresponds to the following R-CP words:

| | | |
|---|---|---|
| $1 = aabbcc$ | $4 = abacb\overline{c}$ | $7 = abacbc$ |
| $2 = aabccb$ | $5 = abac\overline{bc}$ | $8 = aabcb\overline{c}$ |
| $3 = aabc\overline{bc}$ | $6 = abca\overline{bc}$ | $9 = aabcb\overline{c}$ |


1
9
2
8
3
7
4
6
5
$b \odot a$
$\overline{c} \odot a$
$\overline{b} \odot a$
$a \odot b$


1
9
2
8
3
7
4
6
5
c⊙b
$\overline{c}$⊙b
$\overline{a}$⊙b
b⊙c


1
9
2
8
3
7
4
6
5
$\overline{b} \odot c$
$a \odot \overline{b}$
$c \odot \overline{a}$
$c \odot \overline{b}$


1
9
2
8
3
7
4
6
5
$\overline{c} \odot \overline{b}$
$b \odot \overline{c}$
$a \odot \overline{c}$
$\overline{b} \odot \overline{c}$

# Appendix IV

**Maple Programs Used**

Maple programs developed by Lau and Garity [2] were used and extended upon in finding and classify R-CP words in this paper. A Maple V worksheet, programs.mws, containing all the programs used is available at ftp://ftp.math.orst.edu/publications/garity/REU. Do to the larger number of subroutines used in these programs, the code is not provided here. The following is a brief description of the useful programs developed in research for this paper.

Creating Equivalence Classes of R-CP Words

- *CPLIST:* This process takes in a list and returns the list with all the words in R-CP form.
- *TOTAL:* TOTAL takes in a word and returns a set of all R-CP words that are equivalent to the word under a single Type II automorphism.

Doing a Set of One-Letter Automorphisms

- *AL, AL1, AL2:* These programs input a word, a distinguished letter x, and a letter y. They all apply the one-letter automorphism $(y \odot x)$ to the given words. These programs differ in their output formats. AL returns the equivalent R-CP word if the length of the word is not increase. AL1 returns the output word after trivial cancelations have been made. AL2 simply yields the exact output word.
- *SAL, SAL1, SAL2:* These functions require a word, followed by five pairs of letters representing one-letter automorphisms. The programs output the words obtained under these automorphisms using AL, AL1, and AL2, respectively.
- *run7:* This program inputs a list of any length and five pairs of letters representing one-letter automorphisms. SAL1 is applied to each of the words in the list.

Finding Equivalent R-CP Words

- *compare, comp:* compare is an extremely redundant recursive procedure that takes in a list and returns all the words in the list sorted by equivalence classes. Comp is a less redundant version of compare. Both of these procedures call EQW, by Lau. For large lists or long words these programs encounter problems due to too many main loops in EQW.

# Appendix V

**Table of Comparison between R-CP words and $\bigcirc_{n,\{q_i\}}$**

This table compares the maximum number of R-CP words in an equivalence class to $\bigcirc_{n,\{q_i\}}$ for their respective word lengths and weights.

| Length | Weights | R-CP | $\bigcirc$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 2 | 2 | 1 | 1 |
| 3 | 3 | 1 | 1 |
| 4 | 4 | 1 | 1 |
| | 2-2 | 2 | 2 |
| 5 | 5 | 1 | 1 |
| | 3-2 | 2 | 2 |
| 6 | 6 | 1 | 1 |
| | 4-2 | 3 | 3 |
| | 3-3 | 1 | 5 |
| | 2-2-2 | 9 | 24 |
| 7 | 7 | 1 | 1 |
| | 5-2 | 3 | 3 |
| | 4-3 | 1 | 5 |
| | 3-2-2 | 25 | 30 |
| 8 | 8 | 1 | 1 |
| | 6-2 | 4 | 4 |
| | 5-3 | 5 | 7 |
| | 4-4 | 5 | 7 |

# Appendix VI

**List of Homotopic Curves on a Torus**

| Length | Class | R-CP elements |
|---|---|---|
| 1 | 1 | $a$ |
| 2 | 1 | $aa$ |
| 3 | 1 | $aaa$ |
| 4 | 1 | $aaaa$ |
| | 2 | $aabb$<br>$ab\overline{a}b$ |
| 5 | 1 | $aaaaa$ |
| | 2 | $aaba\overline{b}$ |
| | 3 | $aaabb$<br>$aab\overline{a}b$ |
| 6 | 1 | $aaaaaa$ |
| | 2 | $aaabbb$ |
| | 3 | $aabaa\overline{b}$ |
| | 4 | $aab\overline{aa}\overline{b}$ |
| | 5 | $aaaba\overline{b}$ |
| | 6 | $aabba\overline{b}$ |
| | 7 | $aaba\overline{bb}$ |
| | 8 | $aaaabb$<br>$aaab\overline{a}b$<br>$aab\overline{aa}b$ |

These tables represent homotopic curves on a one hole torus. They use the tables from Appendix II and the reduction $xy\overline{x}\overline{y} = 1$ for $x,\ y \in L_2$

| Length | Class | R-CP elements |
|---|---|---|
| 7 | 1 | $aaaaaaa$ |
| | 2 | $aaaaba\overline{b}$ |
| | 3 | $aaaabbb$ |
| | 4 | $aaabaa\overline{b}$ |
| | 5 | $aaabba\overline{b}$ |
| | 6 | $aaabb\overline{a}b$ |
| | 7 | $aaab\overline{a}bb$ |
| | 8 | $aaab\overline{aab}$ |
| | 9 | $aabbaa\overline{bb}$ |
| | 10 | $aabba\overline{bb}$ |
| | 11 | $aabb\overline{aab}$ |
| | 12 | $aaaaabb$<br>$aaaab\overline{a}b$<br>$aaab\overline{aa}b$ |

| Length | Class | R-CP elements |
|---|---|---|
| 8 | 1 | $aaaaaaaa$ |
| | 2 | $aaaaaba\overline{b}$ |
| | 3 | $aaaabbb$ |
| | 4 | $aaaabaa\overline{b}$ |
| | 5 | $aaaaba\overline{bb}$ |
| | 6 | $aaaabba\overline{b}$ |
| | 7 | $aaaabbbb$ |
| | 8 | $aaaabb\overline{a}b$ |
| | 9 | $aaaab\overline{a}bb$ |
| | 10 | $aaaab\overline{aab}$ |
| | 11 | $aaabaa\overline{bb}$ |
| | 12 | $aaaba\overline{bbb}$ |
| | 13 | $aaabbaa\overline{b}$ |
| | 14 | $aaabba\overline{bb}$ |
| | 15 | $aaabbba\overline{b}$ |
| | 16 | $aaabb\overline{a}bb$ |
| | 17 | $aaabb\overline{aa}b$ |
| | 18 | $aaabb\overline{aab}$ |
| | 19 | $aaab\overline{aa}bb$ |
| | 20 | $aaab\overline{aaab}$ |

| Length | Class | R-CP elements |
|---|---|---|
| 8 | 21 | $aaab\overline{aabb}$ |
| | 22 | $aabbaa\overline{bb}$ |
| | 23 | $aabb\overline{aabb}$ |
| | 24 | $aabb\overline{abab}$ |
| | 25 | $aabab\overline{a}bb$<br>$aab\overline{a}bba\overline{b}$ |
| | 26 | $aaba\overline{b}a\overline{bb}$<br>$aaba\overline{bbab}$ |
| | 27 | $aabbaabb$<br>$aba\overline{b}aba\overline{b}$ |
| | 28 | $aaabbabb$<br>$aaba\overline{bb}ab$<br>$ababa\overline{b}a\overline{b}$ |
| | 29 | $aababba\overline{b}$<br>$aababb\overline{a}b$ |
| | 30 | $aaaaaabb$<br>$aaaaab\overline{a}b$<br>$aaaab\overline{aa}b$<br>$aaab\overline{aaa}b$ |
| | 31 | $aaba\overline{b}abb$<br>$aabba\overline{b}ab$ |

| Length | Class | R-CP elements |
|---|---|---|
| 6 | 9 | $aabbcc$ |
| | | $aabcb\bar{c}$ |
| | | $aabccb$ |
| | | $aabc\bar{b}c$ |
| | | $abacbc$ |
| | | $abacb\bar{c}$ |
| | | $abac\bar{b}\bar{c}$ |
| | | $abca\bar{b}\bar{c}$ |
| 7 | 13 | $abac\overline{acb}$ |
| | | $aba\bar{b}c\overline{ac}$ |
| | 14 | $aaabbcc$ |
| | | $aaabcb\bar{c}$ |
| | | $aaabccb$ |
| | | $aaabc\bar{b}c$ |
| | | $aabacb\bar{c}$ |
| | | $aabac\overline{bc}$ |
| | | $aabbacc$ |
| | | $aabbc\bar{a}c$ |
| | | $aabbc\bar{b}c$ |
| | | $aabcacb$ |
| | | $aabca\overline{bc}$ |
| | | $aabcb\overline{ac}$ |
| | | $aabccb\bar{c}$ |
| | | $aabcc\bar{a}b$ |
| | | $aabc\bar{a}b\bar{c}$ |
| | | $aabc\overline{ab}c$ |
| | | $aabc\bar{b}ac$ |
| | | $aabc\bar{b}a\bar{c}$ |
| | | $aabc\bar{b}cb$ |
| | | $aab\bar{a}c\bar{b}c$ |
| | | $abaca\overline{bc}$ |
| | | $abacbc\bar{b}$ |
| | | $abacb\overline{ac}$ |
| | | $abac\bar{a}b\bar{c}$ |

These tables represent the homotopic curves on a two hole torus. They uses the tables from Appendix II with the reduction $xy\overline{xy} = 1$ for $x, y \in L_3$ .

| Length | Class | R-CP elements |
|---|---|---|
| 7 | 15 | $aabacbc$ |
| | | $aabacc\bar{b}$ |
| | | $aaba\bar{b}cc$ |
| | | $aabbca\bar{c}$ |
| | | $aabcac\bar{b}$ |
| | | $aabca\bar{c}b$ |
| | | $aabcbac$ |
| | | $aabcbb\bar{c}$ |
| | | $aabcb\bar{a}c$ |
| | | $aabcb\bar{c}b$ |
| | | $aabcc\overline{ab}$ |
| | | $aabcc\overline{bc}$ |
| | | $aab\bar{a}cbc$ |
| | | $aabc\bar{a}c\bar{b}$ |
| | | $aabc\overline{ac}b$ |
| | | $abaca\overline{cb}$ |
| | | $abacb\bar{a}c$ |
| | | $abac\bar{a}bc$ |
| | | $abac\bar{a}c\bar{b}$ |
| | | $aba\bar{b}acb\bar{c}$ |
| | | $aba\bar{b}c\bar{a}c$ |

# References

[1] P.J. Higgins and R.C. Lyndon. Equivalence of elements under automorphisms of a free group. *J. London Math. Soc.*, pages 8: 254–258, 1974.

[2] Micheal Lau. A Computer Implementation of Whitehead's Algorithm. *Proceedings of the 1997 Research Experience for Undergraduates Program in Mathematics, Oregon State University*, pages 41-66, 1997.

[3] E.S. Rapaport. On free groups and their automorphisms. *Acta. Math.*, pages 99:139–163, 1958.

[4] C.M. Sanchez Minimal Words in the Free Group of Rank Two. *J. Pure Applied Algebra*, pages 17: 333-337, 1980.

[5] J.H.C. Whitehead. On certain sets of elements in a free group. *Proceedings of the London Math. Society*, pages 41:48–56, 1936.

[6] J.H.C. Whitehead. On equivalent sets of elements in a free group. *Ann. of Math.*, pages 37:782–800, 1936.