7-1-2004

# Combinatorial properties of Thompson's group F

Sean Cleary
*City College of New York*

Jennifer Taback
*State University of New York Albany*

TRANSACTIONS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 356, Number 7, Pages 2825–2849
S 0002-9947 03)03375-0
Article electronically published on October 28, 2003

# COMBINATORIAL PROPERTIES OF THOMPSON'S GROUP $F$

SEAN CLEARY AND JENNIFER TABACK

Abstract. We study some combinatorial consequences of Blake Fordham's theorems on the word metric of Thompson's group in the standard two generator presentation. We explore connections between the tree pair diagram representing an element $w$ of , its normal form in the in nite presentation, its word length, and minimal length representatives of it. We estimate word length in terms of the number and type of carets in the tree pair diagram and show sharpness of those estimates. In addition we explore some properties of the Cayley graph of with respect to the two generator nite presentation. Namely, we exhibit the form of "dead end" elements in this Cayley graph, and show that it has no "deep pockets". Finally, we discuss a simple method for constructing minimal length representatives for strictly positive or negative words.

## 1. Introduction

Thompson's group $F$ has been studied extensively in many different branches of mathematics, including group theory, dynamics, homotopy theory and logic. Algebraically, it is most commonly understood in two different forms: via a finite presentation and an infinite presentation. The infinite presentation $\mathcal{P}$ has simple relators which are conveniently manipulated and understood, as well as a unique normal form for elements. The finite presentation $\mathcal{F}$ has two generators and two relators, but there is no longer a convenient set of normal forms. The elements of $F$ can also be interpreted as pairs of finite binary rooted trees with the same number of carets.

In this paper we discuss many interesting combinatorial properties of Thompson's group $F$. These properties are derived from the relationship between the normal form of elements of $F$ and the pairs of finite binary rooted trees used to represent elements of $F$. The combinatorial properties we describe have applications to estimating word length in $F$ and counting and determining caret types. These properties lead to algorithms for constructing minimal length paths in the standard two generator presentation. The sections of this paper are organized as follows.

- In **Section 2** we present a brief introduction to Thompson's group $F$, including Fordham's method of calculating word length [6]. We detail the bijective process which transforms a tree pair diagram representing an element of $F$ into its unique normal form in the infinite presentation.

Received by the editors August 22, 2002 and, in revised form, March 20, 2003.
2000 *Mathematics Subject Classi cation.* Primary 20F65.
The rst author acknowledges support from PSC-CUNY grant #63438-0032. The second author thanks the University of Utah for their hospitality during the writing of this paper.

- In **Section 3** we apply Fordham's method to immediately obtain an estimate of the word length $|w|$ of an element $w \in F$ from the tree pair diagram representing $w$ in the word metric arising from the finite presentation $\mathcal{F}$. We give examples which show that the constants in the estimate are sharp.
- In **Section 4** we use Fordham's method of calculating word length to explore an interesting phenomenon which occurs in the Cayley graph of $F$ with respect to the standard two generator finite presentation. Namely, there are *dead end elements* $w$ with the property that $|w\;\;| = |w| \;\; 1$ for all generators $\;\in \{x_0^{\pm 1},\ x_1^{\pm 1}\}$, where $|w|$ denotes word length. Fordham [6] remarks that some of these dead end elements have a particular form; we give a general form for all dead end elements in $F$ and describe some limits to stronger forms of this behavior, called "deep pockets."
- In **Section 5** we describe the combinatorial relationship between the normal form of an element $w \in F$ and the number and types of carets in the tree pair diagram representing $w$.
- In **Section 6** we present a method of constructing minimal length paths in the standard two generator presentation for strictly positive or negative words in $F$.

## 2. Thompson s group $F$

Thompson's group is best understood combinatorially using the two presentations mentioned above, the finite presentation

$$\mathcal{F} = \langle x_0, x_1 | x_1^{\;1} x_2 x_1 = x_3, x_1^{\;1} x_3 x_1 = x_4 \rangle,$$

where we define $x_n = x_0^{\;1} x_{n\;\;1} x_0$ for $n > 1$, and the infinite presentation

$$\mathcal{P} = \langle x_k,\ k \geq 0 | x_i^{\;1} x_j x_i = x_{j+1} \ \text{ if } i < j \rangle.$$

A convenient set of normal forms for elements of $F$ in the infinite presentation $\mathcal{P}$ is given by $x_{i_1}^{r_1} x_{i_2}^{r_2} \dots x_{i}^{r}\; x_{j_l}^{\;\;s_l} \dots x_{j_2}^{\;\;s_2} x_{j_1}^{\;\;s_1}$, where $r_i, s_i > 0$, $i_1 < i_2 \dots < i_k$ and $j_1 < j_2 \dots < j_l$. To obtain a unique normal form for each element, we add the condition that when both $x_i$ and $x_i^{\;1}$ occur, so does $x_{i+1}$ or $x_{i+1}^{\;1}$, as discussed by Brown and Geoghegan [1]. We will always mean unique normal form when we refer to a word $w$ in normal form.

Analytically, we can regard $F$ as the group of orientation-preserving piecewise-linear homeomorphisms from $[0, 1]$ to itself, where each homeomorphism has only finitely many singularities of slope, all such singularities lie in the dyadic rationals $\mathbf{Z}[\frac{1}{2}]$, and, away from the singularities, the slopes are powers of 2.

2.1. **Tree pair diagrams.** An element of $F$ can be interpreted geometrically via a tree pair diagram, which is a pair of rooted binary trees $(T\;\;, T_+)$, each with the same number of exposed leaves, as described in Cannon, Floyd and Parry [4]. An *exposed* leaf ends in a vertex of valence 1, and we number these exposed leaves from left to right, beginning with 0. We refer to a node together with the two downward-directed edges from the node as a *caret*. A caret $C$ may have a *right child*, a caret $C_R$ which is attached to the right edge of $C$. We can similarly define the *left child* $C_L$ of the caret $C$. The set of all carets which stem from the right leaf of a caret $C$ is called the right subtree of $C$, and we can analogously define the left subtree of $C$.

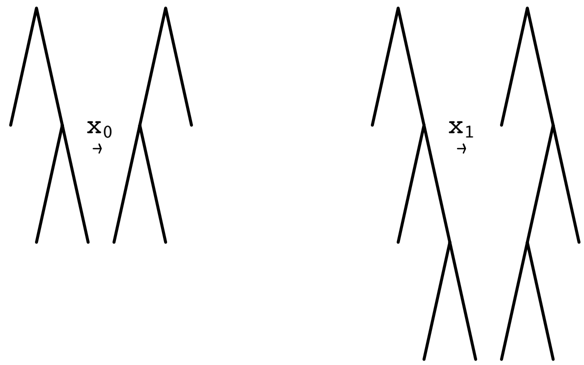


FIGURE 1. The tree pair diagrams for the generators $x_0$ and $x_1$ of $\mathcal{F}$.

In a tree pair $(T\ \ , T_+)$, the tree $T$ is called the *negative tree* and $T_+$ the *positive tree*. This terminology is explained further in §2.2 below. The equivalence between tree pair diagrams and homeomorphisms of $[0, 1]$ is described in [4]. In Figure 1 we give the tree pair diagrams for the generators $x_0$ and $x_1$ of $\mathcal{F}$. In §2.2 below the correspondence between the trees and the elements is explained.

A tree pair diagram is *unreduced* if both $T$ and $T_+$ contain a caret with two exposed leaves numbered $m$ and $m + 1$. There are many tree pair diagrams representing the same element of $F$, but each element has a unique reduced tree pair diagram representing it. When we write $(T\ \ , T_+)$ to represent an element of $F$, we are assuming that the tree pair is reduced.

We refer the reader to Cannon, Floyd and Parry [4] for an excellent introduction to Thompson's group $F$, and to Cleary and Taback [5] for more details on understanding geometrically the elements of $F$ as reduced tree pairs. All of the geometric facts used below are justified in [5].

2.2. **Exponents in tree pair diagrams.** There is a bijective correspondence between the tree pair diagram of $w = (T\ \ , T_+)$ and the normal form of $w$, described in [4]. In the tree pair $(T\ \ , T_+)$, number the exposed leaves of $T$ and $T_+$ from left to right, beginning with 0. The *exponent* of the leaf labelled $k$, written $E(k)$, is defined as the length of the maximal path consisting entirely of left edges from $k$ which does not reach the right side of the tree. Note that $E(k) = 0$ for an exposed leaf labelled $k$ which is a right leaf of a caret, as there is no path consisting entirely of left edges originating from $k$. In Figure 1, number the exposed leaves of the trees in the pair representing $x_1$ from left to right, beginning with 0. Then the exponents of the leaves of $T$ are all 0, and the exponents of the leaves of $T_+$ are $0, 1, 0, 0$, in order. We refer the reader to [5] for a more detailed example of computing exponents in a tree.

Once the exponents of the leaves in $T$ and $T_+$ have been computed, the normal form of the element $w = (T\ \ , T_+)$ is easily obtained. The positive part of the normal form of $w$ is

$$x_0^{E(0)} x_1^{E(1)} \cdots x_m^{E(m)},$$

where $m$ is the number of exposed leaves in either tree, and the exponents are obtained from the leaves of $T_+$. The negative part of the normal form of $w$ is similarly found to be

$$x_m^{-E(m)} x_{m-1}^{-E(m-1)} \cdots x_0^{-E(0)},$$

where the exponents are now computed from the leaves of $T_-$. Note that many of the exponents in the normal form as given above may be zero.

Similarly, given an element $x$ in normal form with respect to the infinite generating set, it is possible to construct a tree pair diagram $(T_-, T_+)$ so that each leaf has the correct exponent. If $R$ is a right caret with a single exposed left leaf labelled $k$, then $E(k) = 0$ by definition. Thus, arbitrarily many right carets with no left subtrees can be added to either $T_-$ or $T_+$ without affecting the normal form to ensure that both trees have the same number of carets, and thus equivalently the same number of exposed leaves.

2.3. **Fordham's method of calculating word length.** For an element $w$ of $F$, we let $|w|$ denote the word length of $w$ with respect to the word metric arising from the finite presentation $\mathcal{F}$. Fordham [6] presents a method of calculating $|w|$ based on the trees $T_-$ and $T_+$ in the tree pair diagram representing $w$. He defines seven types of carets that can be found in a rooted binary tree, and an intricate system of weights assigned to different pairs of caret types, which sum to $|w|$. A detailed example of calculating $|w|$ in this way can be found in [5].

Let $T$ be a finite rooted binary tree. The *left side* of $T$ is the maximal path of left edges beginning at the root of $T$. Similarly, we have the *right side* of $T$. A caret in $T$ is a *left caret* if its left edge is on the left side of the tree, a *right caret* if it is not the root and its right edge is on the right side of the tree, and an *interior caret* otherwise. The carets and the exposed leaves of $T$ are numbered independently, according to different methods. As above, the exposed leaves are numbered from left to right, beginning with 0. The carets in $T$ are numbered according to the infix ordering of nodes. Caret 0 is a left caret with an exposed left leaf numbered 0 in the leaf numbering. According to the infix scheme, we number the left children of a caret before the caret itself, and number the right children after numbering the caret.

In Figure 2 we give an example of a tree whose carets are numbered according to the infix numbering method. More examples of trees whose carets are numbered in this way can be found in [5].

Fordham classifies carets into seven disjoint types, as follows:

1. $L_0$. The first caret on the left side of the tree, with caret number 0. Every tree has exactly one caret of type $L_0$.
2. $L_L$. Any left caret other than the one numbered 0.
3. $I_0$. An interior caret which has no right child.
4. $I_R$. An interior caret which has a right child.
5. $R_I$. Any right caret numbered $k$ with the property that caret $k+1$ is an interior caret.
6. $R_{NI}$. A right caret which is not an $R_I$ but for which there is a higher-numbered interior caret.
7. $R_0$. A right caret with no higher-numbered interior carets.

The root caret is always considered to be a left caret and will be of type $L_L$ unless it has no left children, in which case it would be the single caret of type $L_0$.

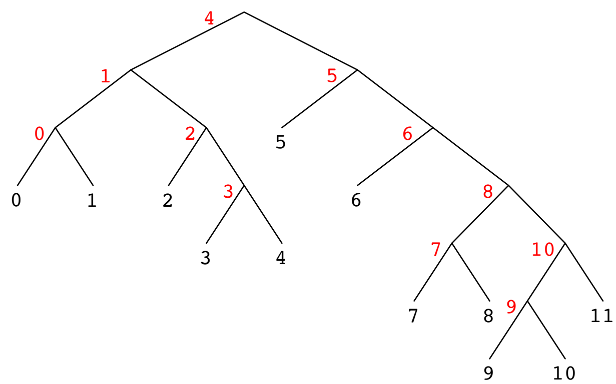


FIGURE 2. A tree whose leaves are numbered left to right and whose nodes are numbered according to the infix method.

The main result of Fordham [6] is that the word length $|w|$ of $w = (T\ ,T_+)$ can be computed from knowing the caret types of the carets in the two trees, as long as they form a reduced pair, via the following process. We number the $k+1$ carets according to the infix method described above, and for each $i$ with $0 \leq i \leq k$ we form the pair of caret types consisting of the type of caret number $i$ in $T$ and the type of caret number $i$ in $T_+$. The single caret of type $L_0$ in $T$ will be paired with the single caret of type $L_0$ in $T_+$, and for that pairing we assign a weight of 0. For all other caret pairings, we assign weights according to the following table:

| | $R_0$ | $R_{NI}$ | $R_I$ | $L_L$ | $I_0$ | $I_R$ |
|---|---|---|---|---|---|---|
| $R_0$ | 0 | 2 | 2 | 1 | 1 | 3 |
| $R_{NI}$ | 2 | 2 | 2 | 1 | 1 | 3 |
| $R_I$ | 2 | 2 | 2 | 1 | 3 | 3 |
| $L_L$ | 1 | 1 | 1 | 2 | 2 | 2 |
| $I_0$ | 1 | 1 | 3 | 2 | 2 | 4 |
| $I_R$ | 3 | 3 | 3 | 2 | 4 | 4 |

The main result of Fordham [6] is the following theorem.

**Theorem 2.1** (Fordham [6], 2.5.1)**.** *Given a word $w \in F$ described by the reduced tree pair diagram $(T\ ,T_+)$, the length $|w|_{\mathcal{F}}$ of the word with respect to the generating set $\mathcal{F}$ is the sum of the weights of the caret pairings in $(T\ ,T_+)$.*

2.4. **How generators a ect a tree pair diagram.** The strength of Fordham's method is that it requires only geometric information about the pair of trees representing an element $w$ to determine $|w|$. Beginning with an element $w = (T\ ,T_+)$, if we knew the reduced pair of trees which represented $w\ $ for $\ \in \{x_0^{\pm 1},\ x_1^{\pm 1}\}$, we could deduce the word length of $w\ $. We now discuss how the tree pair diagrams for $w$ and $w\ $ are related.

We begin with a lemma from Fordham [6] which states under fairly broad conditions that when applying a generator to a tree pair $(T\ ,T_+)$, exactly one pair of caret types will change.

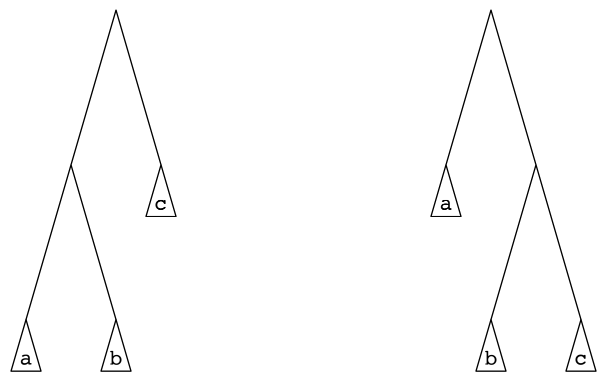


FIGURE 3. The action of $x_0$ transforms the right tree to the left one, while the action of $x_0^{\ 1}$ transforms the left tree to the right one. Each tree represents only the negative trees in their respective tree pairs.

**Lemma 2.2** (Fordham [6], Lemma 2.3.1)**.** *Let* $(T\ \ , T_+)$ *be a reduced pair of trees, each having* $m+1$ *carets, representing an element* $x \in F$*, and* *any generator of* $\mathcal{F}$ *which can be applied to* $x$ *without the addition of a new caret pair to* $(T\ \ , T_+)$*. That is,*

1. *If* $\ \ = x_0$*, the left subtree of the root of* $T$ *must be nonempty.*
2. *If* $\ \ = x_0^{\ 1}$*, the right subtree of the root of* $T$ *must be nonempty.*
3. *If* $\ \ = x_1$*, the left subtree of the right child of the root of* $T$ *must be nonempty.*
4. *If* $\ \ = x_1^{\ 1}$*, the right subtree of the right child of the root of* $T$ *must be nonempty.*

*Then if the reduced tree pair diagram for* $x$ *also has* $m+1$ *carets, there is exactly one* $i$ *with* $0 \leq i \leq m$ *so that the pair of caret types of caret* $i$ *changes when* *is applied to* $x$*.*

We now begin to understand geometrically the action of a generator of $\mathcal{F}$ on a reduced tree pair $(T\ \ , T_+)$, and the corresponding change in normal form. In this section we will assume that the conditions of Lemma 2.2 are met by the generic elements with which we begin. The following geometric lemma describing the action of the generators in $\mathcal{F}$ on an element $w$ is proven in [5]. Let $C_R$ denote the caret which is the right child of the root caret of $T$ , and $C_L$ the left child of the root.

**Lemma 2.3** ([5], Lemmas 2.6 and 2.7)**.** *If* $w = (T\ \ , T_+) \in F$ *satis es the appropriate condition of Lemma 2.2, then* $x_0$ *(resp.* $x_0^{\ 1}$*) alters the position of the right subtree of* $C_L$ *in* $T$ *(resp. the left subtree of* $C_R$*) as depicted in Figure 3. In addition,* $x_1$ *and* $x_1^{\ 1}$ *perform analogous operations on the subtrees of* $C_R$*, as depicted in Figure 4.*

Notice that in all of the descriptions above, the tree $T_+$ from the pair $w = (T\ \ , T_+)$ is not affected when a generator is applied to $w$. This is not true in general for reduced tree pair diagrams not satisfying the conditions of Lemma 2.2. In general, $T_+$ can be affected in exactly three ways:

(1) when $T$ has a single left edge, and the generator is $x_0$,

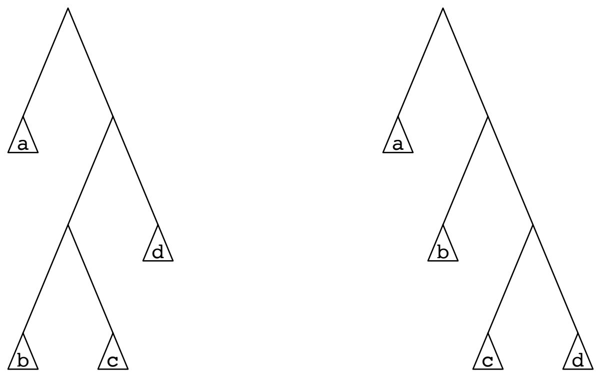


FIGURE 4. The action of $x_1$ transforms the right tree to the left one, while the action of $x_1{}^{1}$ transforms the left tree to the right one. Again, each tree represents only the negative trees in their respective tree pairs.

(2) when the left subtree of the right child of the root caret of $T$   is empty, and the generator is $x_1$, or
(3) if the generator is   and the pair of trees corresponding to $x$   is not reduced.

When the generators $x_0$ and $x_0{}^{1}$ are applied to an element $w \in F$, the change in normal form is straightforward. Namely, $wx_0{}^{1}$ remains in normal form. If $w = w'x_0{}^{1}$ in normal form, then $wx_0 = w'$ in normal form. Otherwise, $w = x_0^m w''$ in normal form, where $m \geq 0$. In this case, $wx_0 = x_0^{m+1}\phi(w'')$, where $\phi : F \to F$ is the shift map which increases the index of each generator in the normal form of $w$.

We now determine the change in normal form when a generator $x_1^{\pm 1}$ is applied to an element $w$ in normal form. The following lemmas are proven in [5].

**Lemma 2.4** (The normal form of $wx_1{}^{1}$, [5], Lemma 2.4)**.** *Let $w \in F$ be represented by the tree pair $(T\ \ , T_+)$, and have normal form $x_{i_1}^{r_1} \cdots x_{i_n}^{r_n} x_{j_m}{}^{s_m} \cdots x_{j_1}{}^{s_1}$. Then $wx_1{}^{1}$ has normal form*

$$x_{i_1}^{r_1} \cdots x_{i_n}^{r_n} x_{j_m}{}^{s_m} \cdots x_{j_{q\ 1}}{}^{s_q\ 1} x\ \ {}^{1} x_{j_q}{}^{s_q} \cdots x_{j_1}{}^{s_1}, \tag{1}$$

*where we might have   $= j_{q+1}$. If the root caret of $T$   has right and left subtrees $S_R$ and $S_L$ respectively, then   is smallest leaf number in $S_R$.*

**Lemma 2.5** (The normal form of $wx_1$, [5], Lemma 2.5)**.** *Let $w$ satisfy the conditions of Lemma 2.2 and have normal form $x_1^{r_1} \cdots x_{i_n}^{r_n} x_{j_m}{}^{s_m} \cdots x_{j_1}{}^{s_1}$. Then $wx_1$ has normal form*

$$x_1^{r_1} \cdots x_{i_n}^{r_n} x_{j_m}{}^{s_m} \cdots x_{j_l}{}^{(s_l\ \ 1)} \cdots x_{j_1}{}^{s_1}, \tag{2}$$

*for some index $j_l$, which is the smallest leaf number in the right subtree of $T$  .*

2.5. **Calculating distance between elements using tree pair diagrams.** When viewing elements of $F$ as homeomorphisms of $[0, 1]$, it is clear that inversion and group multiplication correspond to inversion and composition of homeomorphisms. We now interpret inversion and group multiplication in terms of tree pair diagrams.

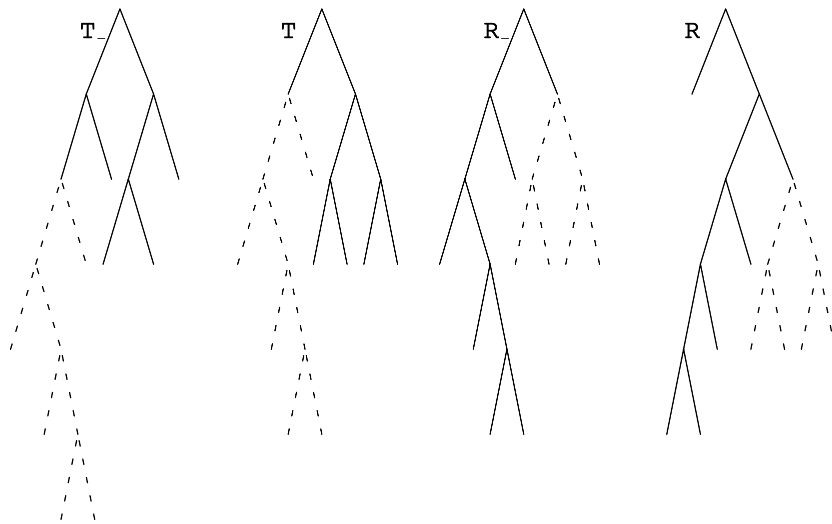


FIGURE 5. Composing $x_1^3x_2^{-1}x_1^{-1}x_0^{-2}$ with $x_1x_2^{-1}x_0^{-1}$ to get $x_1^3x_5x_6^{-1}x_2^{-1}x_1^{-1}x_0^{-3}$ by adding the dashed carets.

Inversion of a group element $f$ given by a tree pair diagram $(T_-, T_+)$ is simply the tree pair diagram $(T_+, T_-)$.

Given two group elements $f, g \in F$ with tree pair diagrams $f = (T_-, T_+)$ and $g = (R_-, R_+)$, we would like to form their product $fg$ by a process consistent with composition of homeomorphisms. When $T_+$ and $R_-$ are identical, we see that $g \circ f$ is represented by the (possibly unreduced) tree pair diagram $(T_-, R_+)$. This corresponds to composition of the piecewise linear homeomorphisms represented by $f$ and $g$, where $Range(f) = Domain(g)$.

When $T_+$ and $R_-$ differ, we create temporary, unreduced representatives of $f$ and $g$ in which the new trees $T_+$ and $R_-$ are identical. Then the composition is carried out in the same manner as described above.

Figure 5 gives an example of the composition of two elements of $F$. The solid lines indicate the original carets and the dashed lines indicate carets added to perform the composition which create unreduced representatives of both elements. To measure the distance between two elements of $F$, we use the word metric on the product $f^{-1}g$ to obtain the metric $d(f, g) = |f^{-1}g|$.

## 3. ESTIMATING THE WORD METRIC $d_{\mathcal{F}}$

It follows immediately from the chart in §2.3 that the word length $|w|_{\mathcal{F}}$ of $w = (T_-, T_+) \in F$ can be estimated in terms of the number of carets $N(w)$ in either tree. This estimate is analogous to the one given by Burillo, Cleary and Stein [2] which has multiplicative constant 12 for the upper bound.

**Theorem 3.1.** *Let $w \in F$ be represented by a tree pair $(T_-, T_+)$ in which each tree has $N(w)$ carets. Then*

$$N(w) - 2 \leq |w|_{\mathcal{F}} \leq 4N(w) - 4.$$

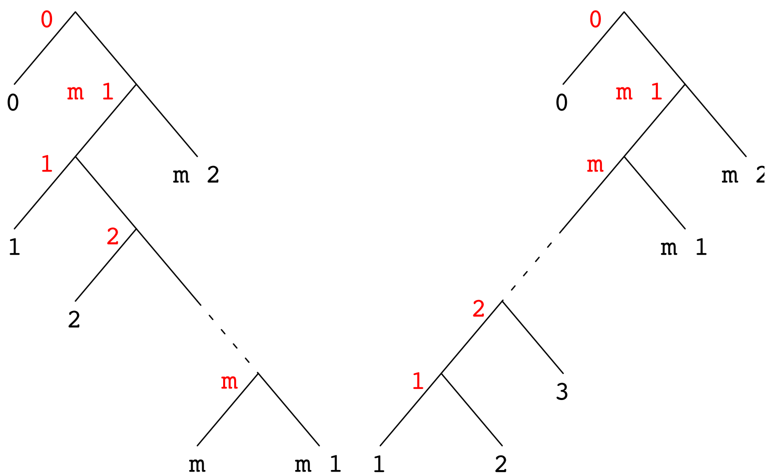


FIGURE 6. Tree pair diagram $(T\ \ ,T_+)$ for the word $x_1^m x_m^{\ 1} x_{m\ \ 1}^{\ \ 1} \cdots x_1^{\ 1}$ with the carets and the leaves numbered.

*Proof.* We first note that every reduced tree pair has a caret type pair $(L_0, L_0)$ of weight 0. Also, it is possible to have the last caret type pair be $(R_0, R_0)$ which also has weight 0. (The only instance in which this does not happen is when the root caret of $T$ or $T_+$ has no right subtree.) For the upper bound, we can ignore the first $(L_0, L_0)$ caret pair, and looking at the chart in §2.3, we see that the maximum weight of any other pair of caret types is 4. Thus the word length of $w$ is at most $4(N(w)\ \ \ 1)$.

To compute the lower bound, we ignore pairs of carets of type $(L_0, L_0)$ and $(R_0, R_0)$. Any other pair of caret types has weight at least 1, and the lower bound is easily obtained.

It is natural to ask if the multiplicative coefficient of 4 in Theorem 3.1 can be improved to 3, since looking at the chart in §2.3 we see that there are very few entries which are 4; that is, very few caret type pairs actually have weight 4. The answer is no; one can produce words which get extremely close to the bound of 4 by pairing $I_R$ and $I_0$ carets in a particular way.

**Example 3.2.** Words of the form $x_1^m x_m^{\ 1} x_{m\ \ 1}^{\ \ 1} \cdots x_1^{\ 1}$, where $m > 1$ is a positive integer, realize the upper bound of 4 in Theorem 3.1.

Words of the above form are represented by the tree pair diagram given in Figure 6. They are constructed so that most carets in $T$ are of type $I_R$ and are paired with carets of type $I_0$ in $T_+$, to give a weight of 4 per pair for most caret pairs. The weights of the different caret pairs are summarized in the following table:

| Caret numbers | Caret types | Weight per caret | Total weight |
|---|---|---|---|
| 0 | $(L_0, L_0)$ | 0 | 0 |
| $1, 2, \cdots, m\ \ \ 1$ | $(I_R, I_0)$ | 4 | $4(m\ \ \ 1)$ |
| m | $(I_0, I_0)$ | 2 | 2 |
| m+1 | $(R_0, R_0)$ | 0 | 0 |

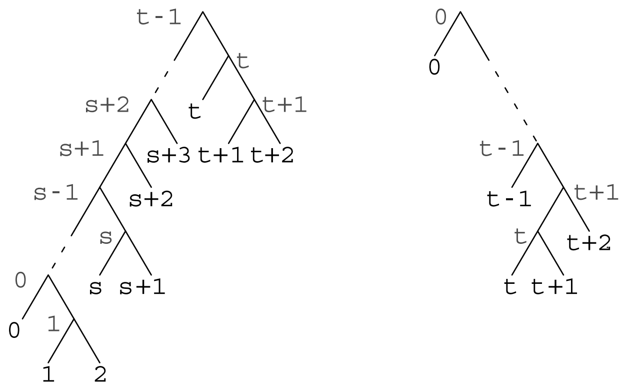


FIGURE 7. The tree pair diagram $(T_-, T_+)$ for words $x_t x_s^{-1} x_{s-2}^{-1} \cdots x_5^{-1} x_3^{-1} x_1^{-1} x_0^{-m}$, with both the carets and the leaves numbered.

The total weight of a word $w$ of this form is $4(m-1)+2 = 4m-2$. The total number of carets $N(w)$ is $m+2$, so these examples, which have weight $4m-2 = 4N(w)-10$, show that for large $N(w)$ the multiplicative coefficient of 4 in Theorem 3.1 is optimal.

It is also natural to wonder if the lower bound can be realized; that is, are there examples of words $w$ where the number of carets is exactly two more than the word length $|w|_{\mathcal{F}}$? This will always be true for words of the form $x_1^{\pm n}$ (but not for $x_0^{\pm n}$). In the following example, we show that it can also be true for more complicated words.

**Example 3.3.** Words of the form $x_t x_s^{-1} x_{s-2}^{-1} \cdots x_5^{-1} x_3^{-1} x_1^{-1} x_0^{-m}$, where $t > s+2$, $s$ is odd, and $m$ is chosen so that the root caret of $T_-$ is caret number $t-1$, realize the lower bound of Theorem 3.1.

These words are represented by tree pairs of the form given in Figure 7. The weights of the caret type pairs are summarized in the following table:

| Caret numbers | Caret types | Weight per caret |
|---|---|---|
| $0$ | $(L_0, L_0)$ | 0 |
| $2, 4, \cdots, s-1$, even numbers | $(L_L, R_{NI})$ | 1 |
| $1, 3, \cdots, s$, odd numbers | $(I_0, R_{NI})$ | 1 |
| $s+1, \cdots t-2$ | $(L_L, R_{NI})$ | 1 |
| $t-1$ | $(L_L, R_I)$ | 1 |
| $t$ | $(R_0, I_0)$ | 1 |
| $t+1$ | $(R_0, R_0)$ | 0 |

It is clear from the table that the total weight of the word is two less than the number of carets, realizing the lower bound of Theorem 3.1.

Theorem 3.1 has an immediate improvement for strictly positive or negative words.

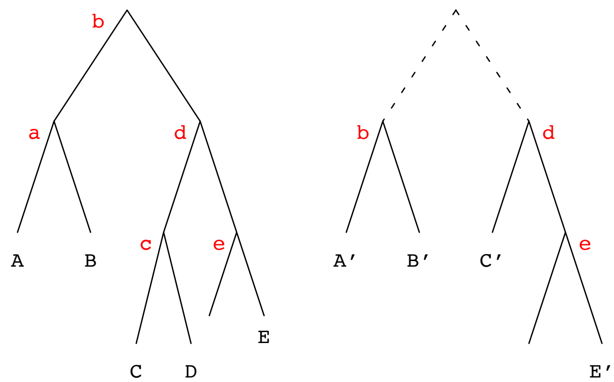


FIGURE 8. The general form of dead end elements of $F$. Capital letters represent (possibly empty) subtrees, while lowercase letters label the carets. Since E and E' are nonempty, caret d is of type $R_{NI}$ in at least one tree.

**Corollary 3.4.** *Let $w$ be a strictly positive or negative word, represented by a tree $T$ having $N(w)$ carets. Then*

$$N(w) \quad 2 \leq |w|_{\mathcal{F}} \leq 3N(w) \quad 3.$$

*Proof.* Since $w$ is strictly positive or negative, one of the trees in the tree pair diagram for $w$ consists entirely of $R_0$ carets (excepting the root caret which is of type $L_0$ and does not contribute to the total weight of the word). Thus, we only need to look at the first column of the chart in §2.3 to assign weights to the different carets. We notice that the maximum weight in the first column of this chart is 3, and the corollary follows.

## 4. Dead end elements

We now consider the Cayley graph  of $F$ with respect to the finite generating set $\mathcal{F}$. Fordham describes a family of elements $w \in F$, which we call *dead end elements*, that have the property that all four generators $x_0^{\pm 1}$ and $x_1^{\pm 1}$ decrease the word length of $w$. They are "dead ends" in the sense that a geodesic ray in  from the identity cannot pass through them; that is, a ray through a dead end element $w$ can no longer be geodesic past $w$. We prove that all dead end elements have a particular form, which is slightly more general than the examples given by Fordham [6], and we discuss other possible dead end behavior in .

### 4.1. The form of dead end elements in $F$.

**Theorem 4.1.** *All dead end elements in $F$ are given by tree pairs of the form in Figure 8 where the subtrees $E, A'$ and $E'$ are nonempty.*

The key step in the proof of Theorem 4.1 is enumerating the conditions under which a specific generator  $\in \{x_0^{\pm 1}, x_1^{\pm 1}\}$ decreases the word length of an element. This requires a detailed understanding of caret pairings as well as how a generator can affect a tree pair. We begin with a lemma of Fordham [6] which states that if $w$ does not satisfy the appropriate condition of Lemma 2.2, then $|w\ \ | = |w| + 1$. This allows us to consider only words satisfying all the conditions of Lemma 2.2 as possible dead end words.

**Lemma 4.2** (Fordham [6], Lemma 2.4.2)**.** *Let   be a generator of $\mathcal{F}$, and $w = (T\ \ , T_+)$ a word not satisfying the condition in Lemma 2.2 corresponding to  . Then $|w\ \ | = |w| + 1$.*

Thus, if $w \in F$ does not satisfy all the conditions of Lemma 2.2, then $w$ is not a candidate for a dead end word.

We let $C_R$ and $C_L$ denote the left and right carets, respectively, of the root caret of $T\ \ $. Similarly, let $C_{RR}$ and $C_{RL}$ denote the right and left children, respectively, of the caret $C_R$. Let $S_L$ and $S_R$ denote the left and right subtrees, respectively, of the root caret of $T\ \ $. Continue this notation to let $S_{RL}$ denote the left subtree of $C_R$, etc. In general, if $\overline{D}$ is a string of entries from the set $\{R,\ L\}$, then $S_{\overline{D}L}$ is the left subtree of $C_{\overline{D}}$. We analogously define $S_{\overline{D}R}$. We use the notation $R_*$ to refer to a right caret of any type, and $I_*$ to refer to an interior caret.

We now rewrite the chart in §2.3 from a different perspective. Assume that $w = (T\ \ , T_+)$ satisfies the conditions of Lemma 2.2. We are interested in the conditions on the pairings of certain carets in $T\ \ $ which determine whether $|w\ \ | = |x| + 1$ or $|w\ \ | = |x|\ \ \ 1$ for a given generator $\ \ \in \{x_0^{\pm 1}, x_1^{\pm 1}\}$.

These conditions are summarized in the charts below. Since we are only considering elements $w = (T\ \ , T_+) \in F$ satisfying Lemma 2.2, we know that only the type of a single caret $C$ in $T\ \ $ will change when a generator of $\mathcal{F}$ is applied. We list the initial type of this caret $C$ in the second column of the charts below, and the new type of caret $C$ in the third column. Column 4, titled "Increase", lists the types of carets in $T_+$ which can be paired with $C$ in order for $|w\ \ | = |w|+1$, and in column 5, titled "Decrease", we put the pairings of $C$ which yield $|w\ \ | = |w|\ \ \ 1$. These pairings are determined by whether certain subtrees of $T\ \ $ are empty or not. These conditions are summarized in the following table.

**Changes in word length when a specific generator is applied to** $w = (T\ \ , T_+)$**.**
Consider the elements $w = (T\ \ , T_+)$ and $wx_0$. Caret $C$ is the root caret of $T\ \ $.

| Condition on $T$ | Initial type of caret $C$ | New type of caret $C$ | Increase if $C$ paired with | Decrease if $C$ paired with |
|---|---|---|---|---|
| $S_{RL} \neq \emptyset$ | $L_L$ | $R_I$ | $R_*, I_*$ | $L_L$ |
| $S_{RL} = \emptyset, S_{RR} \neq \emptyset$ | $L_L$ | $R_{NI}$ | $R_*, I_R$ | $L_L, I_0$ |
| $S_{RL} = \emptyset, S_{RR} = \emptyset$ | $L_L$ | $R_0$ | $R_{NI}, R_I, I_R$ | $R_0, L_L, I_0$ |

Consider the elements $w = (T\ \ , T_+)$ and $wx_0{}^{\ 1}$. Caret $C$ is the caret $C_R$ of $T\ \ $.

| Condition on $T$ | Initial type of caret $C$ | New type of caret $C$ | Increase if $C$ paired with | Decrease if $C$ paired with |
|---|---|---|---|---|
| $S_{RL} \neq \emptyset$ | $R_I$ | $L_L$ | $L_L$ | $R_*, I_*$ |
| $S_{RL} = \emptyset, S_{RR} \neq \emptyset$ | $R_{NI}$ | $L_L$ | $L_L, I_0$ | $R_*, I_R$ |
| $S_{RL} = \emptyset, S_{RR} = \emptyset$ | $R_0$ | $L_L$ | $R_0, L_L, I_0$ | $R_{NI}, R_I, I_R$ |

Consider the elements $w = (T \ \ , T_+)$ and $wx_1$. Caret $C$ is the caret $C_{RL}$ of $T$ .

| Condition on $T$ | Initial type of caret $C$ | New type of caret $C$ | Increase if $C$ paired with | Decrease if $C$ paired with |
|---|---|---|---|---|
| $S_{RLR} \neq \emptyset$ | $I_R$ | $R_I$ | none | any |
| $S_{RLR} = \emptyset, S_{RR} \neq \emptyset$ | $I_0$ | $R_{NI}$ | $R_0, R_{NI}$ | $L_L, I_*, R_I$ |
| $S_{RLR} = \emptyset, S_{RR} = \emptyset$ | $I_0$ | $R_0$ | $R_{NI}$ | $L_L, I_*, R_I, R_0$ |

Consider the elements $w = (T \ \ , T_+)$ and $wx_1^{\ \ 1}$. Caret $C$ is the caret $C_R$ of $T$ .

| Condition on $T$ | Initial type of caret $C$ | New type of caret $C$ | Increase if $C$ paired with | Decrease if $C$ paired with |
|---|---|---|---|---|
| $S_{RRL} \neq \emptyset$ | $R_I$ | $I_R$ | any | none |
| $S_{RRL} = \emptyset, S_{RRR} \neq \emptyset$ | $R_{NI}$ | $I_0$ | $L_L, I_*, R_I$ | $R_0, R_{NI}$ |
| $S_{RRL} = \emptyset, S_{RRR} = \emptyset$ | $R_0$ | $I_0$ | $L_L, I_*, R_I, R_0$ | $R_{NI}$ |

*Proof of Theorem 4.1.* From Lemma 4.2 we may assume that the initial word $w$ satisfies all the conditions of Lemma 2.2. Combining the above four charts, we see that for all four generators to reduce the word length of $w$, the element $w$ must be represented by a tree pair $(T \ \ , T_+)$, where $T$ is given in Figure 8. We now determine which of the following subtrees $A$, $B$, $C$, $D$, and $E$ may be empty.

The possible pairings of carets $a, b, c$ and $d$ are also determined by the above four charts. The combination of these conditions restricts the pairings further, as follows. It is now clear that $x_0$ causes $b$ to become a caret of type $R_I$, forcing $b$ to be paired with a caret of type $L_L$. Since $a < b$, caret $a$ must be paired with a caret appearing before $b$, thus the left subtree $A'$ of caret $b$ in $T_+$ is nonempty.

When $x_1^{\ \ 1}$ reduces the word length of $w$, we now see that caret $d$ becomes a caret of type $I_0$ rather than of type $I_R$, and thus $d$ must be paired with a caret of type $R_0$ or $R_{NI}$. It follows that the left subtree of caret $d + 1 = e$ in $T_+$ is empty (otherwise caret $d$ in $T_+$ would be of type $R_I$).

We are now left with showing that the subtrees $E$ of $T$ and $E'$ of $T_+$ are nonempty. Recall that the two trees $T$ and $T_+$ have the same number of carets. If $E$ was empty, then $E'$ must also be empty, given the placement of caret $d$ in both trees. If this is the case, then the pair $(T \ \ , T_+)$ is not reduced, contradicting initial assumptions. Similarly, if $E'$ is empty, so is $E$, and the pair is again not reduced. Thus $E$ and $E'$ are both nonempty, and we have shown that all dead end elements have the claimed form.

4.2. **Pockets in the Cayley graph of $F$.** A natural question to ask is whether there are more severe forms of dead end phenomena in $F$. Many researchers have wondered if there might be *pockets* in the Cayley graph of $F$. For $k > 0$, the element $w \in F$ defines a $k$-pocket if $w \in B_{id}(n)$, where $n = |w|$ and $B_w(k) \subset B_{id}(n)$; that is, if all paths of length $k$ emanating from $w$ remain in the ball of radius $n$ centered at the identity. A dead end element of the form described above defines a 2-pocket. Although there are dead end elements, we now show that there are no deeper pockets.

**Theorem 4.3.** *There are no $k$-pockets in the Cayley graph of $F$ with respect to the nite generating set $\mathcal{F}$ for $k \geq 3$.*

*Proof.* A word $w$ which defines a $k$-pocket must be a dead end word, otherwise there would immediately be a path of length 1 from $w$ which would leave $B_{id}(n)$. We will produce a path of length 3 emanating from any dead end word $w$ which leaves $B_{id}(n)$. The key fact in constructing this path is that according to Theorem 4.1, in a dead end word $w$, the left subtree of $C_{RR}$ is empty. In Figure 8 the caret $C_{RR}$ is labelled $e$. We label the exposed left leaf of caret $C_{RR}$ by $m$, and it follows that the caret number of $C_{RR}$ is also $m$.

Let $w = (T\ \ , T_+) \in B_{id}(n)$ be a dead end word. Then $|wx_0^{\ 1}| = |w| \quad 1$ by construction. In the tree pair diagram $(R\ \ , R_+)$ representing $wx_0^{\ 1}$, caret $m$ is the right child of the root, and leaf $m$ is still its exposed left leaf. We notice that $wx_0^{\ 1}$ does not satisfy the condition of Lemma 2.2 corresponding to $x_1$, so an application of $x_1$ would increase the length of $wx_0$ by one to give $|wx_0^{\ 1}x_1| = |w| = n$. To construct this path of length 3 which leaves the ball, we look at the resulting pair of trees for $wx_0^{\ 1}x_1$.

If $w = (T\ \ , T_+)$ is a word not satisfying the condition of Lemma 2.2 corresponding to the generator  , then  acts on $(T\ \ , T_+)$ in a way that adds an additional caret. To see this, we create an unreduced representative of $w$ by adding carets to both trees so that the unreduced representative satisfies the condition of Lemma 2.2 corresponding to  . Then, we allow  to act on the unreduced pair; the resulting tree pair will represent $w$  and be reduced.

An alternate way to determine the resulting pair of trees is to simplify the normal form for $wx$  and draw the corresponding trees.

Consider the tree pair $(R\ \ , R_+)$ representing the element $wx_0^{\ 1}$. By adding an extra caret attached to leaf $m$ to both trees in the pair, we obtain an unreduced representative of $wx_0^{\ 1}$ which satisfies the condition of Lemma 2.2 corresponding to $x_1$. Figure 4 in §2 exhibits the change in $R$  when $x_1$ is applied to this unreduced tree pair. We see that in the negative tree of the tree pair representing $wx_0^{\ 1}x_1$, the right caret of the root again has an exposed left leaf. Hence the tree pair diagram of $wx_0^{\ 1}x_1$ again does not satisfy the conditions of Lemma 2.2 corresponding to $x_1$. Thus, by Lemma 4.2, $|wx_0^{\ 1}x_1^2| = |w| + 1$ and is not in $B_{id}(n)$, and there can be no $k$-pockets in the Cayley graph of $F$ for $k \geq 3$.

## 5. Counting carets

Given the beautiful relationship between the normal form of elements of $F$ and their representation as pairs of finite binary rooted trees, it is natural to ask what information about the trees can be readily determined from the normal form. We show that the total number of carets in each tree can be determined, as well as the number of right, interior and left carets in each tree. As an application, this count is used to give a more accurate estimation of $|w|_{\mathcal{F}}$ than the one given in Theorem 3.1. We note that Burillo [3] also estimates $|w|_{\mathcal{F}}$ from the normal form of $w$.

We temporarily alter our notion of the normal form to make our computations and formulae easier to understand. Namely, let $w$ have normal form

$$x_0^{r_0} x_{i_1}^{r_1} x_{i_2}^{r_2} \dots x_{i}^{r}\ \ x_{j_l}^{\ \ s_l} \dots x_{j_2}^{\ \ s_2} x_{j_1}^{\ \ s_1} x_0^{\ \ s_0}$$

allowing for the possibility that $s_0 = 0$ and $r_0 = 0$ if the generator $x_0$ does not appear in the normal form. We still retain the conditions for uniqueness of the normal form.

We first show that it is easy to detect from the normal form whether the right subtree of the root caret of $T_\pm$ is empty.

**Lemma 5.1** (Seeing the right side of a tree)**.** *Let the element $w = (T_-, T_+)$ have normal form $x_0^{r_0}x_{i_1}^{r_1}x_{i_2}^{r_2}\dots x_{i_k}^{r_k}x_{j_l}^{-s_l}\dots x_{j_2}^{-s_2}x_{j_1}^{-s_1}x_0^{-s_0}$.*

(1) *If $i_k < \sum_{m=0}^{k-1} r_m$, then the right subtree of the root caret of $T_+$ is empty.*
(2) *If $j_l < \sum_{m=0}^{l-1} s_m$, then the right subtree of the root caret of $T_-$ is empty.*

*Proof.* We work through the proof in the case of $T_-$. The proof is identical for $T_+$. We begin to build the tree $T_-$ using the fact that the exponent of $x_j$ in the normal form is the leaf exponent of the leaf labelled $j$ in the tree. Let $S_L$ and $S_R$ denote, respectively, the left and right subtrees of the root caret of $T_-$.

In the tree $T_-$, we build the subtrees $S_L$ and $S_R$, beginning with $s_0+1$ left carets in $S_L$, with highest leaf number $s_0$. If the index $j_1$ is greater than $s_0$, then we must add $s_1$ interior carets added to the right subtree of the root of $T_-$. If not, we add $s_1$ interior carets to the left subtree of the root of $T_-$. Assume that these interior carets are added to the left subtree of the root. We then ask the question again, with the next index. The highest leaf number in $S_L$ is now $s_0 + s_1$. If $j_2 > s_0 + s_1$, then we must add $j_2$ interior carets to the right subtree $S_R$, otherwise we add them to $S_L$. As soon as $j_n > \sum_{m=1}^{n-1} s_m$ for some value of $n$, $S_R$ is nonempty, and the analogous equation is still true for higher indices. Thus it is sufficient to test the highest index to see if $S_R$ is empty.

In the next proposition, we drop the requirement that the trees $T_-$ and $T_+$ in a tree pair have the same number of carets. In §2 we saw that if two trees did not have the same number of carets, this was easily achieved by adding extra carets of type $R_0$ to one of the trees in the pair. The addition of carets had no affect on the normal form. The next proposition shows that one can compute the number of carets in each tree *which are not these extra $R_0$ carets* directly from the normal form.

**Proposition 5.2** (Counting caret types from the normal form)**.** *Let $w = (T_-, T_+)$ have normal form $x_0^{r_0}x_{i_1}^{r_1}x_{i_2}^{r_2}\dots x_{i_k}^{r_k}x_{j_l}^{-s_l}\dots x_{j_2}^{-s_2}x_{j_1}^{-s_1}x_0^{-s_0}$, where we do not require that $T_-$ and $T_+$ have the same number of carets.*

(1) *If the subtree $S_R$ of $T_-$ is not empty, i.e. $w$ does not satisfy condition* (2) *of Lemma 5.1, then the tree $T_-$ then has*
  (a) *$j_l + s_l + 1$ total carets,*
  (b) *$s_0 + 1$ left carets,*
  (c) *$\sum_{m=1}^{l} s_m$ interior carets, and*
  (d) *$j_l + s_l - \sum_{m=0}^{l} s_m$ right carets.*
(2) *If the subtree $S_R$ of $T_-$ is empty, i.e. $w$ satisfies condition* (2) *of Lemma 5.1, then the tree $T_-$ has*
  (a) *$\sum_{m=0}^{l} s_m + 1$ total carets,*
  (b) *$s_0 + 1$ left carets,*
  (c) *$\sum_{m=1}^{l} s_m$ interior carets, and*
  (d) *no right carets.*

*Proof.* We first prove case (1), using the fact that the right subtree of $T_-$ is not empty. To see that the total number of carets of $T_-$ is $j_l + s_l + 1$, notice that there must be a left leaf in $T_-$ labelled $j_l$, but no higher numbered left leaves. Since all the remaining leaves are right, we need $s_l+1$ of them so that all carets are complete (that is, have two leaves).

It follows from the definition of leaf exponent in §2 that there are $s_0+1$ left carets in $T_-$. Every interior caret contributes 1 to the exponent of an exposed leaf numbered $i$, for $i \neq 0$. Thus the number of interior carets is given by $\sum_{m=1}^{l} s_m$. It then follows that the number of right carets of $T_-$ is $(j_l+s_m) - \sum_{m=0}^{l} s_m$; that is, the total number of carets less the number of left and interior carets.

To prove case (2), we follow the proof for case (1), omitting the right carets.

Note that we have the identical theorem for the positive tree $T_+$, replacing any instance of $j_l$ with $i_l$ and $s_m$ with $r_m$, and using condition (1) of Lemma 5.1.

When we reinstate the requirement that the trees in the pair have the same number of carets $N$, the number $N$ will be the maximum of the number of carets obtained below for $T_-$ and $T_+$.

We can now improve the bound in Theorem 3.1 slightly using Proposition 5.2. Theorem 3.1 is imprecise when it must assume, for example, in the upper bound, that each caret has weight 4. In reality there are only a few caret type pairs which carry weight 4.

The theorem below has the assumption that $T_-$ has at most as many carets of type $R_0$ as $T_+$. This is equivalent to saying that if we do not require $T_-$ and $T_+$ to have the same number of carets, then $T_-$ has more carets than $T_+$. This assumption allows us to use the estimates in Proposition 5.2 for the number of carets of either tree in the pair $(T_-, T_+)$ when we reinstate the requirement that the two trees have the same number of carets.

**Theorem 5.3.** *Let $w = x_0^{r_0}x_{i_1}^{r_1}x_{i_2}^{r_2}\dots x_{i_k}^{r_k}x_{j_l}^{-s_l}\dots x_{j_2}^{-s_2}x_{j_1}^{-s_1}x_0^{-s_0}$ be in normal form, and let $(T_-, T_+)$ be the tree pair diagram for $w$. Without loss of generality we assume that $T_-$ has at most as many carets of type $R_0$ as $T_+$.*

(1) *If the subtree $S_R$ of $T_-$ is nonempty, then*

$$j_l + s_l - 1 \leq |w|_{\mathcal{F}} \leq 3(j_l+s_l) + \sum_{m=1}^{l} s_m + 2s_0.$$

(2) *If the subtree $S_R$ of $T_-$ is empty, then*

$$\sum_{m=0}^{l} s_m - 1 \leq |w|_{\mathcal{F}} \leq 2s_0 + 4\sum_{m=1}^{l} s_m.$$

*Proof.* We begin with the proof of case (1). The lower bound is immediate, since from the chart in §2, we know that each caret except the first and possibly the last has weight at least 1. To obtain the upper bound, we use the expressions in Proposition 5.2, part (1), for the number of each type of caret. Looking at the chart in §2.3, we note the maximum weight for a caret of each of the three types: right, left and interior. Namely, the maximum weight for a right caret is 3, for a left caret is 2 and for an interior caret is 4. The upper bound is then 3(# right carets) + 2(# left carets − 1) + 4(# interior carets). We omit a single left caret since there will always be a caret of type $L_0$ in a pair of the form $L_0, L_0)$, which has weight 0.

In case (2), the proof is identical, substituting the caret counts from part (2) of Proposition 5.2.

We note that for strictly positive or negative words, this method may improve the bound significantly.

**Corollary 5.4.** *Let* $w = x_0^{s_0} x_{j_1}^{s_1} x_{j_2}^{s_2} \ldots x_j^{s}$ *or* $w = x_j{}^{s} \ \ldots x_{j_2}{}^{s_2} x_{j_1}{}^{s_1} x_0{}^{s_0}$ *be a strictly positive or negative word.*

(1) *If* $w$ *is a negative word which does not satisfy condition* (2) *of Lemma 5.1, or a positive word which does not satisfy condition* (1) *of Lemma 5.1, then*

$$2(j_k + s_k) \quad \sum_{m=0}^{k} s_m \quad 2 \leq |w|_{\mathcal{F}} \leq 2(j_k + s_k) + \sum_{m=0}^{k} s_m \quad 2s_0.$$

(2) *If* $w$ *is a negative word which satis es condition* (2) *of Lemma 5.1, or a positive word which satis es condition* (1) *of Lemma 5.1, then*

$$\sum_{m=0}^{k} s_m \leq |w|_{\mathcal{F}} \leq s_0 + 3 \sum_{m=1}^{k} s_m.$$

*Proof.* We work through the proof in the case that $w$ is a strictly negative word, so $w = x_j{}^{s} \ \ldots x_{j_2}{}^{s_2} x_{j_1}{}^{s_1}$. The case when $w$ is a positive word is completely analogous.

Let $w = (T \ , T_+)$, where $T_+$ is a tree consisting entirely of the root and $R_0$ carets. The proof of the upper bound in either case is identical to the proof in Theorem 5.3 in the analogous case, except that when we look up on the chart to see the maximum weight of each type of caret we get different results. Namely, the maximum weight for a right caret, when paired with an $R_0$ caret, is 2 rather than the 3 obtained in the proof of Theorem 5.3. Similarly, the maximum weight of a left caret when paired with an $R_0$ caret is 1, and of an interior caret is 3. Using these new values, we compute the upper bound in the corollary.

To obtain the lower bound, in either case, we know that each caret has weight at least 1, but any right carets paired with a caret of type $R_0$ must have weight 2. The exception is a pair of types $(R_0, R_0)$, which has weight 0. But there can be at most one pair of these types in a reduced word, and we account for it by subtracting 1. Thus (# carets)+(# right carets) $\ 2 = (i_k + r_k + 1) + (i_k + r_k + 1) \quad {}^{k}_{m=1} r_m \quad 2 = 2(i_k + r_k) \quad {}^{k}_{m=1} r_m$ is the lower bound in case (1), as desired. In case (2) we obtain $s_1 + \ {}^{k}_{m=1} r_m + 1 + 0 \quad 2 = s_1 + \ {}^{k}_{m=1} r_m \quad 1$ as desired.

## 6. Construction of short paths

We now address the following question. Given a word $w \in F$ written in normal form, how do we find a minimal length representative of $w$ in the finite presentation $\mathcal{F}$, that is, a string representing $w$ which contains only $x_0^{\pm 1}$ and $x_1^{\pm 1}$, whose length is equal to $|w|_{\mathcal{F}}$?

While Fordham's methods present a simple way to calculate the word length of a word presented in the normal form, he does not present an easy way of obtaining minimal length representatives in the presentation $\mathcal{F}$. Fordham's algorithms, including those implemented as LISP programs in his thesis [6], do produce a minimal length representative for a general word in normal form, but the method requires substantial checking of cases, and is more suitable for computer than human execution.

There is a simple method for constructing a (usually nonminimal) representative for a word given in normal form in terms of $x_0^{\pm 1}$ and $x_1^{\pm 1}$, which we call the *replacement method*. We simply use the relators of $\mathcal{P}$ to replace each occurrence of $x_n^{\epsilon}$ in the normal form of $w$ by $x_0^{\ (n\ \ 1)} x_1^{\epsilon} x_0^{n\ \ 1}$. It is easy to see that this method gives a string of generators which is not generally a minimal representative for the initial word. In general, this path differs from the minimal length representative by no more than a factor of 4.

Below, we present a simple method to obtain a minimal length representative for a strictly positive or negative word in $F$; that is, a word whose normal form consists entirely of generators with positive or negative exponents. We call this method the *nested traversal method*. For a general word $w$, the nested traversal method could be applied separately to the negative and positive parts of $w$, but in general that would produce a representative of $w$ which is not minimal in the finite presentation.

6.1. **The nested traversal method.** Assume that $w = (T\ \ , *)$ is a tree pair diagram for a negative word with $m$ carets. The tree $*$ consists almost entirely of $m$ 1 carets of type $R_0$; the only other caret in $*$ is the root caret which is of type $L_0$. Every leaf in the tree $*$ has leaf exponent 0. Viewing the identity as the nonreduced pair $(*, *)$, we detail a sequence of generators which transform it to the pair $(T\ \ , *)$. We note that it is never necessary to apply the generator $x_1$ as part of this process. Also, the sequence of generators produced by this method does not give (in general) a unique representative of the pair $(T\ \ , *)$.

For each caret type $C$, the nested traversal method produces a sequence of generators necessary to transform a caret of type $R_0$ into a caret of type $C$. For each caret type, this can be accomplished in a fixed number of steps, described below, which is exactly the weight of the pair $(C, R_0)$ in the chart in §2.3. Specifically, the following chart gives the number of steps (applications of a generator) required to transform an $R_0$ caret into a caret of type $C$. Note that the single $L_0$ caret will come from the single existing $L_0$ root caret in $*$.

| Type of caret $C$ | $L_0$ | $L_L$ | $I_0$ | $I_R$ | $R_0$ | $R_{NI}$ | $R_I$ |
|---|---|---|---|---|---|---|---|
| Number of steps | 0 | 1 | 1 | 3 | 0 | 2 | 2 |

Thus it is clear that the length of the path given by the nested traversal method (described below), once it is proven that the method produces the appropriate tree, will be the word length of $w = (T\ \ , *)$.

We now describe the nested traversal method, and in the theorem below, prove that it yields the desired tree $T\ \ $. The goal of this method is to identify a sequence of generators of $\mathcal{F}$ which transform a caret of type $R_0$ into a caret of type $C$. In §2.3 we saw that applying $\ \ \in \{x_0^{\pm1},\ x_1^{\ 1}\}$ to a word $w$ satisfying the appropriate condition of Lemma 2.2 will either change the caret type of the root caret or the right child of the root caret of $T\ \ $.

First, we note that to transform an $R_0$ caret which is the right child of the root caret of $T\ \ $ into an $L_L$ caret, we only need apply the generator $x_0^{\ 1}$. Similarly, the generator $x_1^{\ 1}$ transforms an $R_0$ caret which is the right child of the root caret of $T\ \ $ into an $I_0$ caret. This behavior was exhibited in Figures 3 and 4.

The simplest example of a caret of type $I_R$ is an interior caret with a single right child of type $I_0$. We do not view creating the $I_0$ caret (or whatever the right subtree of the $I_R$ caret may contain) as part of the transformation of the $R_0$ caret into the $I_R$ caret. Thus, to create an $I_R$ caret, we must do the following: we apply

$x_0{}^{1}$, moving the $R_0$ caret, which begins as the right child of the root caret of $T$ and which we denote $R$, to the root position of $T$ . Then we create the $I_0$ caret which will be the right child of the completed $I_R$ caret (or create the appropriate right subtree of the completed $I_R$ caret) using the string of generators specified by the nested traversal method for those caret or carets. We do not count these steps toward the creation the $I_R$ caret, but rather as the steps required to create the caret or carets in the right subtree of the final $I_R$ caret. Finally, we apply the generator $x_0$, which moves $R$ back to the position of right child of the root caret, and apply $x_1{}^{1}$, which moves $R$ to an interior caret while moving the newly created $I_0$ caret (or the carets in the right subtree of the $I_R$ caret) to the right subtree of $R$. Thus, the sequence of generators needed to create the $I_R$ caret is $x_0{}^{1} \cdots x_0 x_1{}^{1}$, where the $\cdots$ represents a sequence of generators which create the carets in the right subtree of the final $I_R$ caret, and must occur before the $I_R$ caret can be completed.

The type of a right caret $C$ is determined by the left subtree of the right caret "beneath" $C$ on the right side of the tree, that is, the next right caret further from root than $C$. Keeping in mind that only one of two positions in the tree can be affected by $\in \{x_0^{\pm 1},\ x_1{}^{1}\}$, namely the root position and the right child of the root, it is clear how to transform an $R_0$ caret, denoted $R$, which begins as the right child of the root caret of $T$ , into a caret of type $R_{NI}$ or $R_I$. Namely, we apply $x_0{}^{1}$ to move $R$ to the root position of the tree. Then we use the nested traversal method to create the subtree of the right caret beneath $R$, remembering that these generators are all involved in the creation of *other* carets. Finally, we apply $x_0$, moving $R$ back to the right side of the tree, now with its right subtree correct and thus the caret type is now of the appropriate type.

It is the nesting of these sequences of generators which gives the method its name. To summarize this method, consider the following chart, which lists the sequences of generators required to create each type of caret:

| Caret type | Generators involved in creation of caret |
|---|---|
| $L_0$ | none |
| $L_L$ | $x_0{}^{1}$ |
| $I_0$ | $x_1{}^{1}$ |
| $I_R$ | $x_0{}^{1} \ldots x_0 x_1{}^{1}$ |
| $R_0$ | none |
| $R_{NI}$ | $x_0{}^{1} \ldots x_0$ |
| $R_I$ | $x_0{}^{1} \ldots x_0$ |

To use this method to produce a minimal length representative of a positive or negative word, we traverse the tree in infix order. When we encounter the caret types of $L_0$, $L_L$, $I_0$ and $R_0$, we record at most a single appropriate generator. When we encounter the other caret types $I_R$, $R_{NI}$ and $R_I$, we record the initial generator of the sequence given in the chart above, then apply the process recursively on the right subtree (if present) of the caret; after completion of the process on the right subtree, we record the one or two additional generators specified in the chart, and continue with the infix traversal of the tree $T$ .

We will prove the following theorem.

**Theorem 6.1.** *Let $w$ be a strictly positive or negative word. Then the nested traversal method produces a minimal length representative for $w$.*

We begin with an example of the nested traversal method.

**Example 6.2.** To understand the traversal construction method, we use it to construct a minimal length representative of the strictly negative word $w$ with normal form $x_{10}^{1}x_7^{\ 1}x_6^{\ 1}x_4^{\ 1}x_2^{\ 2}x_0^{\ 2}$.

If $w = (T\ \ , *)$, where $w$ is the word given above, then the carets of $T$ , in infix order, have caret types given in the following table:

| Caret number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Caret type | $L_0$ | $L_L$ | $I_0$ | $I_R$ | $I_0$ | $L_L$ | $I_R$ | $I_0$ | $R_{NI}$ | $R_I$ | $I_0$ | $R_0$ |

Since a tree pair diagram for $w$ will require 11 carets in each tree, we begin with a tree $*$ consisting of one caret of type $L_0$ and 10 carets of type $R_0$. Caret 0 is of type $L_0$ and is identical to the $L_0$ caret of $*$. To create caret 1, we apply $x_0^{\ 1}$. To create caret 2, we apply $x_1^{\ 1}$. Caret 3 is more complicated. We begin with $x_0^{\ 1}$, and then must traverse the right subtree of caret 3, which in this case is the single caret numbered 4. Since caret 4 is of type $I_0$, it requires only the generator $x_1^{\ 1}$. We now return to complete the sequence of generators necessary to create caret 3, namely $x_0x_1^{\ 1}$. Caret 5, the root caret, is of type $L_L$ and requires only the generator $x_0^{\ 1}$. Caret 6 is again more complicated, since it is of type $I_R$. We begin with an $x_0^{\ 1}$ before descending into the right subtree of caret 6, i.e. caret 7, which is created via an $x_1^{\ 1}$. We then finish the creation of caret 6 with the string $x_0x_1^{\ 1}$. To create caret 8, we apply $x_0^{\ 1}$ and begin to traverse the right subtree of caret 8. For caret 9, we apply the generator $x_0^{\ 1}$ and begin to traverse the right subtree of caret 9. Caret 10 is created with a single $x_1^{\ 1}$ generator, and caret 11 is simply the last $R_0$ caret from the initial tree $*$ and requires no additional generators. We then return to finish the creation of caret 9 by applying $x_0$ to make it once again a right caret, and we apply another generator $x_0$ to finish creating caret 8. Thus we have produced the tree $T$   representing $w$ and found the string

$$x_0^{\ 1}x_1^{\ 1}x_0^{\ 1}x_1^{\ 1}x_0x_1^{\ 1}x_0^{\ 1}x_0^{\ 1}x_1^{\ 1}x_0x_1^{\ 1}x_0^{\ 1}x_0^{\ 1}x_1^{\ 1}x_0x_0$$

to be a minimal representative for $w$.

This entire process is illustrated in Figure 9, and described in more detail in Figures 10 through 13.

We now prove Theorem 6.1.

*Proof.* We first prove that the nested traversal method produces the correct tree for strictly negative words in $F$. Once this is proven, it is clear that the string of generators produced by the method is a minimal length representative, because the number of carets needed to create a caret of type $C$ is exactly the weight of the pair $(C, R_0)$. We prove that the correct tree is produced via induction on the length of the normal form of a negative word $w \in F$. Note that the length of the normal form of $w$ is exactly $|w|_{\mathcal{P}}$, the word length in the infinite presentation $\mathcal{P}$.

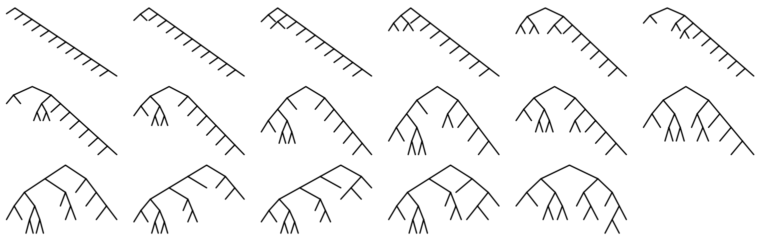

FIGURE 9. The negative trees for the complete construction of a minimal length representative for $w = x_{10}^{1} x_7^{1} x_6^{1} x_4^{1} x_2^{2} x_0^{2}$ via the nested traversal method. The generators needed for each step are exhibited below. The positive trees $*$ are omitted.

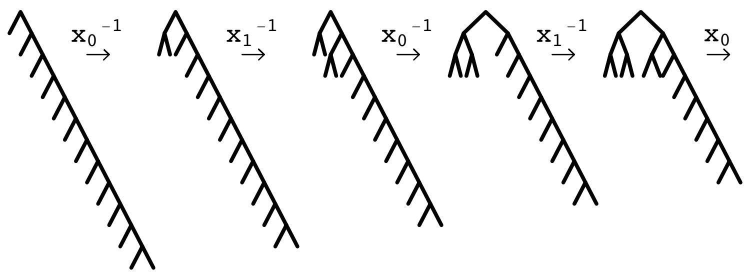


FIGURE 10. The negative trees for the first four steps of the nested traversal construction of a minimal length representative for $w = x_{10}^{1} x_7^{1} x_6^{1} x_4^{1} x_2^{2} x_0^{2}$, listing the generators applied at each stage. The positive trees $*$ are again omitted.

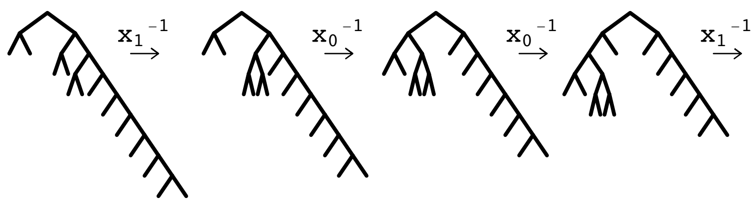


FIGURE 11. The negative trees for the next four steps of the construction.

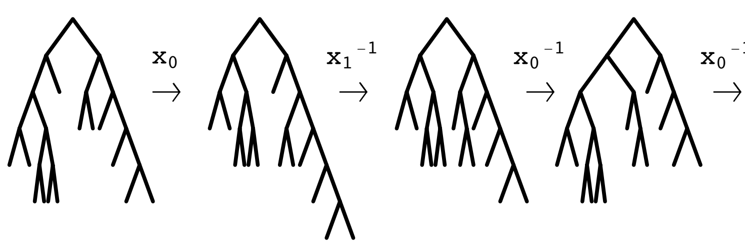


FIGURE 12. The negative trees for the next four steps of the construction.

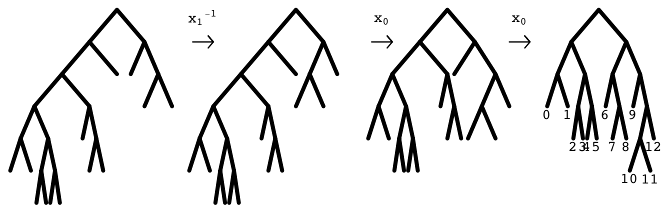


FIGURE 13. The negative trees for the final four steps of the construction of a minimal length representative for $w = x_{10}^{1}x_7{}^{1}x_6{}^{1}x_4{}^{1}x_2{}^{2}x_0{}^{2}$ via the nested traversal method. It is easily checked that the final tree represents $w$.

We begin with the base case of the induction and $w = x_k{}^{1}$. We must prove that the nested traversal method produces the correct tree pair for $w$. If $w = x_0{}^{1}$, then it is clear that the nested traversal method yields the correct pair of trees.

If $w = x_k{}^{1}$ with $k \neq 0$, it is easily checked that the tree $T$ for $w$ is of the following form: caret 0 is of type $L_0$ and is followed by $k \quad 2$ carets of type $R_{NI}$, a single caret of type $R_I$, a single caret of type $I_0$ and a final $R_0$ caret. Creating these carets in infix order, following the nested traversal method, we begin with the tree $*$ containing a single $L_0$ caret and $k+1$ carets of type $R_0$.

According to the nested traversal method, we apply $x_0{}^{(k \quad 1)}$, which moves the first $k \quad 1$ carets of type $R_0$ to the left side of the tree. Each of these generators is the first half of the pair of generators needed for creating an $R_I$ or an $R_{NI}$ caret. Now, we create the $I_0$ caret by applying the generator $x_1{}^{1}$; this creates an $I_0$ caret with exposed leaves numbered $k$ and $k+1$. We finish with $x_0^k$, which moves all but the initial $L_0$ caret on the left side of the tree back to the right side of the tree. Thus, the nested traversal method, with the correct nesting, produced the tree $T$ representing the element $x_k{}^{1}$.

We complete the proof by induction on the size of the negative word. We assume inductively that the nested traversal method produces the correct tree for a negative word $w = x_{j_l}{}^{s_l} \dots x_{j_2}{}^{s_2} x_{j_1}{}^{s_1}$ with tree pair diagram $(T \;, *)$. We consider a new word $v = x_k{}^{1}w$ in normal form whose length (in the infinite presentation) is one more than the length of $w$. If $v = (S \;, *)$, we prove by induction that $S$ can be constructed via the nested traversal method. We divide the proof into two cases:

(1) in case 1, we assume that the normal form of $w$ begins with $x_k{}^{1}$, and
(2) in case 2, we assume that the normal form of $w$ does not begin with $x_k{}^{1}$.

In both cases, the tree $S$ has at least one more caret than the tree $T$ .

**Case 1.** Let $w = x_k{}^{m}w'$ in normal form, and thus $v = x_k{}^{(m+1)}w'$ in normal form. In terms of tree pair diagrams, let $w = (T \;, *)$ and $v = (S \;, *)$. Let $\gamma$ be the string of letters obtained via the nested traversal method which creates the tree $T$ , according to the induction hypothesis. We will obtain a string $\gamma'$ which is a minimal representative of $v$ and creates the tree $S$ .

(1) If $k = 0$, it is clear that the nested traversal method produces the correct tree $S$ and gives a minimal length representative for $v$.
(2) Assume that $k \neq 0$. If the right subtree of the root caret of $T$ is empty (which we note can be detected via Lemma 5.1), then except for the initial

$L_0$ caret, $T$ consists entirely of left and interior carets. Since $k \neq 0$, there must be at least one interior caret in $T$ . Since $\gamma$ constructs $T$ according to the nested traversal method, we know that $\gamma$ can be written as $\gamma = \eta_1 x_1^{\ 1}\eta_2$, where the $x_1^{\ 1}$ creates the final interior caret, which has caret number $k$. In addition, $\eta_2$ is a sequence of letters containing at least $l$ repetitions of $x_0^{\ 1}$, which create a series of left carets in $T$ : the left carets with caret numbers greater than $k$ as well as the left caret with the highest caret number less than $k$. The string $\eta_2$ may contain other letters prescribed by the nested traversal method.

From the normal form of $w$, we know that the highest numbered left leaf in $T$ is leaf number $k$, and it must be the exposed left leaf of a string of $m$ carets of type $I_0$. None of the carets in this string can be of type $I_R$; if they were, then there would be an exposed left leaf with positive leaf exponent whose leaf number is greater than $k$. Since $k$ is the largest index of a generator in the normal form of $w$, such an $I_R$ caret cannot occur in this string of carets.

Such a string of $I_0$ carets is created according to the nested traversal method via a string of $x_1^{\ 1}$ letters. Thus we can further enumerate $\gamma$ as $\gamma = \eta_1 x_1^{\ m}\eta_2$. Since the normal form of $v$ simply contains one more occurrence of the generator $x_k$, we know that in $S$ the exposed leaf numbered $k$ is the left exposed leaf of a sequence of $m+1$ carets of type $I_0$. Thus the tree $S$ is created by the string of letters $\gamma' = \eta_1 x_1^{\ (m+1)}\eta_2$. Since the generator $x_1^{\ 1}$ is what the nested traversal method determines will create an $I_0$ caret, we see that $\gamma'$ is the string produced by the nested traversal method which generates $S$ .

(3) Now suppose that the right caret of the root of $T$ is not empty. In this case, $\gamma$ can be written as $\gamma = \xi_1 x_1^{\ m}\xi_2$, where $\xi_2$ accounts for the right carets that are created in a nested manner. It is easy to see that as above, the tree $S$ is produced from the string $\gamma = \xi_1 x_1^{\ (m+1)}\xi_2$, which is exactly the string of letters produced by the nested traversal method to generate $S$ .

**Case 2.** We now assume that $v = x_k^{\ 1} w$ in normal form, and that the generator $x_k$ does not appear in the normal form of $w$. In all of the subcases below, we begin with a tree $*$ containing at least one more $R_0$ caret than the tree used to create $T$ . Since the generator $x_k^{\ 1}$ does not appear in the normal form of $w$, in the tree $T$ there is a caret $C$ with an exposed leaf numbered $k$, which has leaf exponent 0. It may be the case that one must add carets of type $R_0$ to the right side of $T$ to obtain the caret $C$ with exposed leaf $k$. However, adding these carets does not affect the normal form of the element. Again, let $\gamma$ be the string of letters generated by the nested traversal method which creates the tree $T$ .

(1) First, suppose that $C$ is an interior caret, which must be of type $I_0$ and have an exposed right leaf numbered $k$ in order for the leaf exponent of $k$ to be 0. Since $k$ is larger than any index of a generator in the normal form of $w$, it also follows that the highest index appearing in the normal form of $w$ is at most $k$ 1.

We can write $\gamma = \eta_1 x_1^{\ 1}\eta_2$, where $x_1^{\ 1}$ is the letter in $\gamma$ which creates the caret $C$ from a right caret. It is easily seen that in $\gamma' = \eta_1 x_0^{\ 1} x_1^{\ 1} x_0 x_1^{\ 1}\eta_2$

the caret $C$ becomes an $I_R$ caret, the leaf exponent of $k$ is now 1, and no new carets with higher numbered leaves and positive leaf exponent are added, which might cause additional generators to appear in the normal form for $v$. We have exactly added the sequence of generators corresponding to an $I_R$ caret in the chart above describing the nested traversal method.

(2) Now suppose that $C$ is a left caret with an exposed right leaf numbered $k$. Write $\gamma = \eta_1 x_0{}^{1}\eta_2$, where $x_0{}^{1}$ is the letter in $\gamma$ which creates the caret $C$ according to the nested traversal method. Then in the tree created from $*$ by the string $\eta_1 x_1{}^{1}$, the right child of the root caret has a left subtree consisting of a single $I_0$ caret with exposed leaves numbered $k$ and $k+1$. Then, in the tree corresponding to the string $\eta_1 x_1{}^{1} x_0{}^{1}$, we have created the left caret $C$ whose right subtree contains a single $I_0$ caret, with exposed leaves numbered $k$ and $k+1$. Thus the leaf exponent of $k$ is now 1, and so $x_k{}^{1}$ appears in the normal form of the element. Since the trees $T$ and $S$ differ only in this one place, we see that the string $\gamma' = \eta_1 x_1{}^{1} x_0{}^{1}\eta_2$ creates the tree $S$ . The trees $S$ and $T$ differ in a single $I_0$ caret, and $\gamma'$ and $\gamma$ differ only in the generator $x_1{}^{1}$, which is the letter that the nested traversal method uses to create an $I_0$ caret. We see that $\gamma'$ is the string produced by this method for the tree $S$ .

(3) Finally, suppose that $C$ is a right caret in $T$ with exposed left leaf $k$, with leaf exponent 0. The $C$ must be a caret of type $R_0$; if it is not, then $k$ would be smaller than the index of some generator appearing in the normal form of $w$, contradicting initial assumptions.

Since $w$ is a negative word, it contains a single caret of type $R_0$ which has a single right exposed leaf numbered $l$. If there was a caret of type $R_0$ with two exposed leaves, the pair $(T\ ,*)$ could be reduced. It is necessary to add a string of additional $R_0$ carets to the original tree $*$ in order to obtain $S$ .

Let $m = k \quad t$, where $t$ is the highest index of a generator appearing in the normal form of $w$, so in particular $t < k$. It now follows that the string $\gamma' = \gamma x_0{}^{m} x_1{}^{1} x_0^{m}$ creates the tree $S$ . The first $x_0{}^{m}$ letters move the new $R_0$ carets to the left side of the tree, the $x_1{}^{1}$ creates an interior caret with exposed leaves labelled $k$ and $k+1$, and the $x_0^{m}$ letters move the $m$ carets back to the right side of the tree where they become of type $R_{NI}$ or $R_I$. So again we see that the additional letters needed exactly coincide with the letters prescribed by the nested traversal method for constructing $S$ .

If we begin with a strictly positive word $w$, we use the nested traversal method to construct a minimal length representative for the strictly negative word $w\ {}^{1}$. The inverse of this path will then produce a minimal length representative for $w$.

The authors would like to thank Azer Akhmedov for comments on an earlier draft and the anonymous referee for helpful comments.

## References

[1] K. S. Brown and R. Geoghegan. An in nite-dimensional torsion-free $P_\infty$ group. *Inventiones mathematicae*, 77:367–381, 1984. MR **85m:**20073

[2] J. Burillo, S. Cleary, and M. I. Stein. Metrics and embeddings of generalizations of Thompson's group . *Trans. Amer. Math. Soc.*, 353(4):1677–1689 (electronic), 2001. MR **2 1k:**20087

[3] José Burillo. Quasi-isometrically embedded subgroups of Thompson's group . *J. Algebra*, 212(1):65–78, 1999. MR **99m:**20051

[4] J. W. Cannon, W. J. Floyd, and W. R. Parry. Introductory notes on Richard Thompson's groups. *L'Ens. Math.*, 42:215–256, 1996. MR **98g:**20058

[5] Sean Cleary and Jennifer Taback. Thompson's group is not almost convex. *J. Algebra*. to appear.

[6] Blake Fordham. *Minimal Length Elements of Thompson's group* . Ph.D. thesis, Brigham Young Univ, 1995.

Department of Mathematics, City College of New York, City University of New York, New York, New York 10031

*E-mail address*: `cleary@sci ccny cuny edu`

Department of Mathematics and Statistics, University at Albany, Albany, New York 12222

*E-mail address*: `jtaback@math albany edu`