FINDING MINIMAL LENGTH REPRESENTATIVES IN THOMPSON'S GROUP F

MICAH ISRAEL MILLER

Abstract. Cleary and Taback devised a method called the nested traversal method to construct minimal length representatives for positive and negative elements in Thomspson's group. We show how to use the nested traversal method to construct minimal length representatives for a larger class of elements of this group.

1. Introduction

Thompson's group F is studied in diverse areas of mathematics for its many interesting properties. It has three common interpretations, (analytic, algebraic, and geometric), which give three different ways of studying its properties. Algebraically, Thomspon's group has two standard presentations. One is an infinite presentation, given by

$$\langle x_k, k \geq 0 | x_i^{-1} x_j x_i = x_{j+1}, i < j \rangle.$$

The second is a finite presentation given by

$$\langle x_0, x_1 | [x_0 x_1^{-1}, x_0^{-1} x_1 x_0], [x_0 x_1^{-1}, x_0^{-2} x_1 x_0^{-2}] \rangle.$$

In Thompson's group F, we have an example of a group that has an infinite presentation and a finite presentation. It also has the property that $F \times F$ sits inside F itself.

In this paper, we consider the Cayley graph of Thompson's group with respect to its standard finite presentation. This Cayley graph is a metric space, where distance is defined to be the shortest path between two elements. This gives rise to the word metric. It is not known what this Cayley graph looks like. However, we begin to explore some of its properties below.

A natural question to ask is how to find a minimal length path in the Cayley graph from the identity element to any other element in F. Using purely geometric methods, Fordham showed how to obtain the length of a minimal path in [6]. However, to find a minimal length path using this method involves checking many subcases and is better suited for a computer program. Cleary and Taback have shown how to construct a minimal length representative for a particular class of elements using a method they called the *nested traversal method* [5]. We will give necessary and sufficient conditions for when this method produces minimal length representatives for a larger class of elements in F.

2. Cayley Graphs

The Cayley graph of a group G with finite generating set S is a geometric description of that group. We would like to ascertain properties of the Cayley graph

The author acknowledges support from NSF grant DMS-0437481.

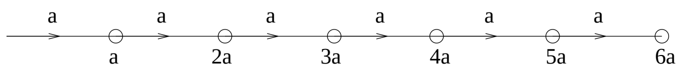

FIGURE 2.1. The Cayley graph of $\mathbb{Z} = \langle a \rangle$, where a is a generator of $\mathbb{Z}$. The circles represent the elements $\{\cdots, -2a, -a, 0, a, 2a \cdots\}$ of $\mathbb{Z}$ and the edges are directed, and labeled with the generator a.

.

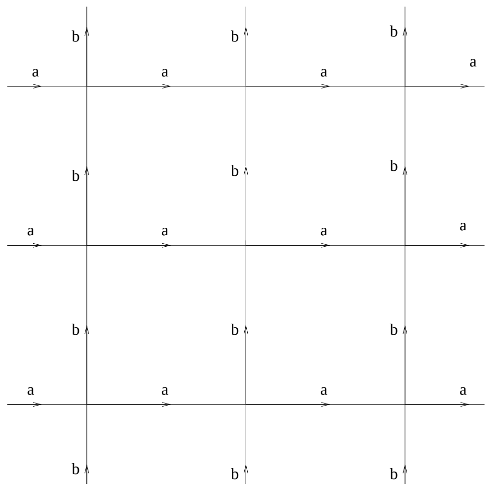

FIGURE 2.2. The Cayley graph of $\mathbb{Z} \times \mathbb{Z} = \langle a, b | ab = ba \rangle$, where $a = (1, 0)$ and $b = (0, 1)$. Each edge is labeled either by generator a or b.

of $(F, \{x_0, x_1\})$, so we can better understand F. The vertices of the Cayley graph are exactly the group elements of G, and so we will label each vertex as a group element. Two elements, x and y, are connected by an edge if $x\alpha = y$ for some generator $\alpha \in S$. We will label each directed edge by the generator α that sends x to y under group multiplication. Going against the direction of the edge can be thought of as multiplying by the inverse, i.e., $y\alpha^{-1} = x$. We can define the distance between elements in the Cayley graph to be the length of a minimal path between those elements, where each edge is a distance of one. This path need not be unique. The reader my be familiar with the Cayley graphs of $\mathbb{Z}$ and $\mathbb{Z} \times \mathbb{Z}$ shown in Figures 2.1 and 2.2, respectively. Since we connect two vertices when multiplication by a generator sends one element to the other, different generating sets will give different Cayley graphs. As an example, we show the Cayley graphs of S_3, the symmetric group on three letters, with respect to two different presentations in Figure 2.3 and 2.4.

When we refer to the Cayley graph of Thompson's group F, we are referring to the Cayley graph with respect to its finite generating set $\{x_0, x_1\}$. We can classify elements in Thompson's group according to which subset of the generators, $\{x_0, x_0^{-1}, x_1, x_1^{-1}\}$, reduce the element's word length under right multiplication. A *dead end element* is defined to be an element whose word length decreases under multiplication by all four generators. This means that in the Cayley graph there are geodesic rays that cannot be extended past dead end elements. Cleary and Taback showed that

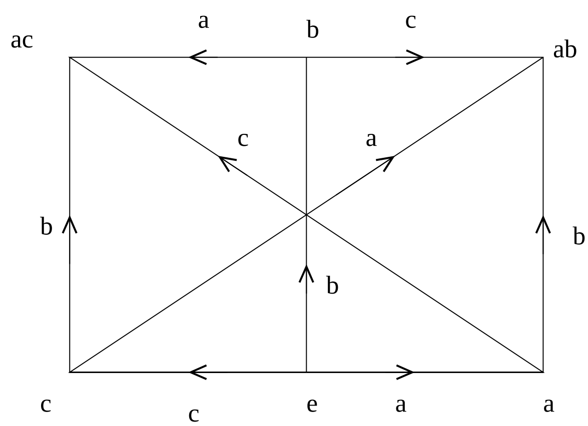

FIGURE 2.3. The Cayley graph of S_3 with the presentation $\langle a, b, c | ab = bc = ca, ac = ba = cb, a^2 = b^2 = c^2 = 1 \rangle$, with edges labelled by generators, the vertices labelled by the group elements they represent, and e representing the identity.

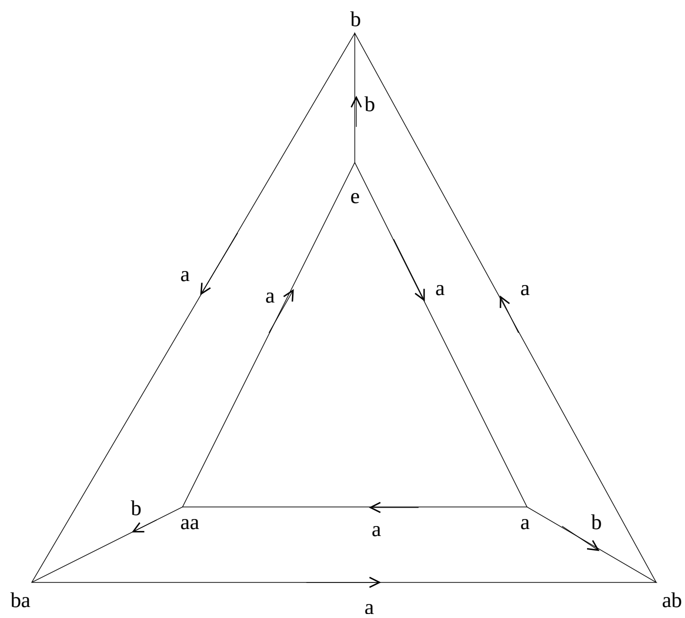

FIGURE 2.4. The Cayley graph of S_3 with the presentation $\langle a, b | a^3 = 1, b^2 = 1, ba = a^2 b \rangle$, with edges labelled by generators, vertices labelled by the group elements they represent, and e representing the identity.

Thompson's group has infinitely many dead end elements and describe their forms [5]. As mentioned earlier, no one knows what the Cayley graph of Thompson's group looks like. Still, it is known that the Cayley graph has infinitely many dead end elements, whose forms are given in [5]. It has also been shown that the Cayley graph is not almost convex with respect to the finite presentation [4]. A useful tool for investigating properties of this Cayley graph will be Fordham's method for calculating word length, described in the next section.

3. Background on Thompson's Group

Thompson's group F has three common interpretations. Analytically, we can define F to be the group of orientation preserving, piecewise-linear homeomorphisms from the unit interval to itself, where each homeomorphism has a finite number singularities whose coordinates lie in $\mathbb{Z}[\frac{1}{2}]$, and away from the singularities, the slopes are powers of two. We can define these homeomorphisms uniquely by their sets of singularities.

Algebraically, Thompson's group F is commonly presented via a finite and an infinite presentation, given above. In the infinite presentation, which has generators $\{x_0, x_1, x_2, \cdots\}$, a normal form for an element $w \in F$ is given by

$$x_{i_1}^{r_1} x_{i_2}^{r_2} \cdots x_{i_k}^{r_k} x_{j_l}^{-s_l} \cdots x_{j_2}^{-s_2} x_{j_1}^{s_1}$$

where $r_i, s_j > 0$, $0 \le i_1 < i_2 \cdots < i_k$, and $0 \le j_1 < j_2 \cdots < j_l$. This normal form is unique if we add the condition that when x_i and x_i^{-1} occur, so does x_{i+1} or x_{i+1}^{-1} (discussed in Brown and Geoghegan [1]). In the finite presentation, there is no convenient set of normal forms for elements in F.

There is also a geometric description of F in terms of pairs of finite, rooted binary trees that have the same number of leaves, which we will call tree pair diagrams. We denote each tree pair diagram by (T_-, T_+), where T_- and T_+ are trees of this type. We define a *caret* of a tree to be a vertex together with the two downward-directed edges from the node and an *exposed leaf* to be an edge which ends in a vertex of valence 1. We label the exposed leaves in a tree from left to right as we move along the tree starting with 0. We say that the tree pair diagram (T_-, T_+) is *unreduced* if both T_- and T_+ contain carets with exposed leaves numbered m and $m + 1$, and *reduced* otherwise. To reduce a tree pair diagram, we remove the carets from T_- and T_+ that have leaves numbered m and $m + 1$ and renumber the remaining exposed leaves. We refer the reader to Figure 3.1, for a picture of an unreduced tree pair diagram and a reduced tree pair diagram. While there are infinitely many tree pair diagrams representing the same element of F, there is only one reduced tree pair diagram for any given element. When we write $w = (T_-, T_+)$ for an element of F we are assuming that the tree pair diagram is reduced.

For a more detailed introduction to Thompson's group F, we refer the reader to [3]. We now discuss the equivalence of these three interpretations of Thompson's group F.

3.1. **Equivalence of algebraic and geometric interpretations of F.** We now give the correspondence between tree pair diagrams (T_-, T_+) and elements $w \in F$ written in the normal form arising from the infinite presentation. First we begin with a few definitions. The *right side* of the tree is the maximal path of right edges beginning at the root caret of the tree. We define the exponent of an exposed leaf numbered k, denoted $E(k)$, to be the length of the maximal path consisting entirely of left edges from k that does not reach the right side of the tree (each edge has length one). In the tree pair diagram $w = (T_-, T_+)$, we use T_- to determine the part of the normal form for w consisting of generators with negative exponents, and T_+ to determine the part of the normal form for w consisting of generators with positive exponents. If the leaf numbered k is in T_+, then it corresponds to the term $x_k^{E(k)}$ in

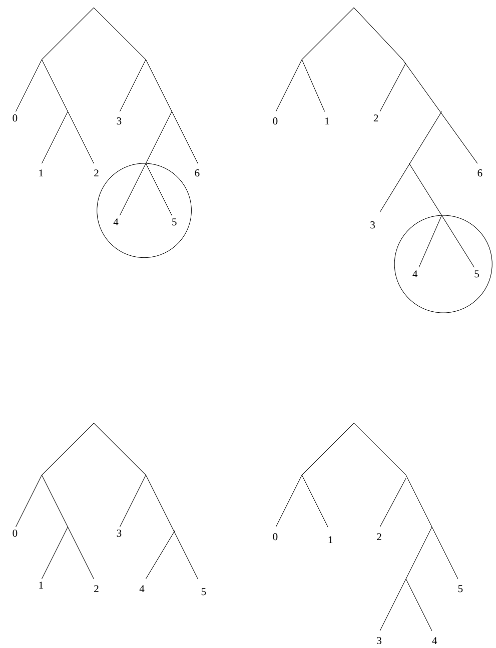

FIGURE 3.1. Top: An unreduced tree pair diagram. Bottom: A reduced tree pair diagram representing the same element.

the normal form of w. If the leaf k is in T_-, then it corresponds to the term $x_k^{-E(k)}$ in the normal form of w. When $E(k) = 0$, the term x_k does not appear in the normal form of w. If there are m exposed leaves in T_- and n exposed leaves in T_+, the tree pair diagram corresponds to the element

$$x_0^{E(0)} x_1^{E(1)} \cdots x_n^{E(n)} x_m^{-E(m)} \cdots x_1^{-E(1)} x_0^{-E(0)}$$

It is clear that given an element w that we can construct a tree pair diagram whose leaves have the correct exponents, representing w.

We refer the reader to Figure 3.2, for a picture of a tree pair diagram. Consider the leaf numbered zero in T_-. There is path of two left edges starting at the leaf, but the second left edge reaches the right side of the tree, so the exponent of the leaf is 1. Any path starting at the leaf numbered one in T_- will begin with a right edge, so $E(1) = 0$. A path starting at the leaf numbered two in T_- begins with a left edge, but this edge does reach the right side of the tree, and so $E(2) = 0$. The exponents of the leaves in T_- in increasing order are 1, 0, 0, 2, 0, 0, 0. And the exponents of the leaves in T_+ are 2, 1, 0, 0, 0, 0, 0. Thus, the tree pair diagram represents the element $x_0^2 x_1 x_3^{-2} x_0^{-1}$.

Note that an exposed leaf of a caret with an edge on the right side of the tree will always have exponent equal to zero. When T_- and T_+ do not have the same number of carets, we can add these types of carets to the tree with fewer carets until T_- and T_+ do have the same number of carets, without changing the element. From now on, we will assume that T_- and T_+ have the same number of carets and exposed leaves.

Using tree pair diagrams, it is possible to construct an injective map from $F \times F$ to F, as mentioned in the introduction.

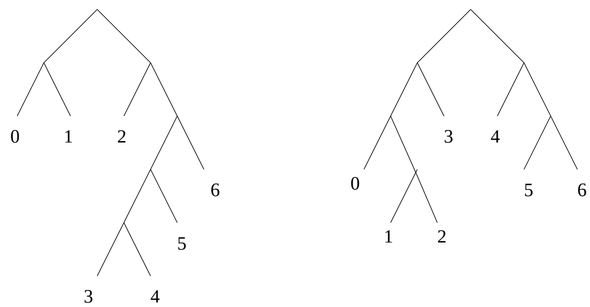

FIGURE 3.2. The reduced tree pair diagram corresponding to the element $x_0^2 x_1 x_3^{-2} x_0^{-1}$.

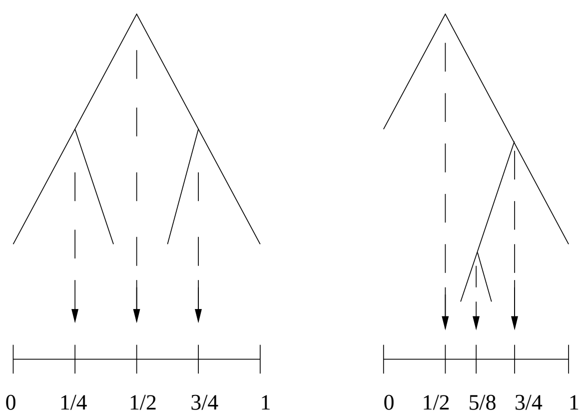

FIGURE 3.3. The tree pair diagram for $x_1 x_0^{-1}$ and its corresponding subdivision of the unit interval

3.2. **Equivalence of geometric and analytic interpretations of F.** To give the correspondence between the tree pair diagrams and the piecewise-linear homeomorphisms in the analytic interpretation of F, it is sufficient to show how a tree pair diagram (T_-, T_+) determines the set of singularities of the corresponding homeomorphism. This is done by viewing the tree pair diagram as a set of instructions for successive subdivisions of the unit interval. Each caret in the tree can be viewed as a break in a given interval into two equal parts. So, the root caret of the tree corresponds to a break of the unit interval into the intervals $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$. The left child of the root caret divides the left interval, $[0, \frac{1}{2}]$, to $[0, \frac{1}{4}]$ and $[\frac{1}{4}, \frac{1}{2}]$. Similarly, the right child of the root caret divides the right half of the interval $[\frac{1}{2}, 1]$ into $[\frac{1}{2}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$. We continue this process for all carets in T_- and T_+. The endpoints of the intervals determined by T_- are the x-coordinates of the singularities and the endpoints of the intervals determined by T_+ are the y-coordinates of the singularities of the element when viewed as a homeomorphism of the unit interval. We list the x-coordinates in increasing order and then the y-coordinates in increasing order. Pairing the x-coordinates to the y-coordinates in the same position gives us the singularities of the homeorphism corresponding to the element. We are able to perform this pairing because T_- and T_+ have the same number of carets. We work out an example in Figure 3.3. The tree pair diagram corresponds to the element $x_1x_0^{-1}$. The root carets in each tree divide the interval into $[0, \frac{1}{2}]$ and $[\frac{1}{2}, 1]$. In the T_- tree, the left child of the root caret breaks the left half of the unit interval into interval $[0, \frac{1}{4}]$ and $[\frac{1}{4}, \frac{1}{2}]$, while the right child of the root caret divides the right half of the unit interval into $[\frac{1}{2}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$. The right child of the root caret in T_+ divides the right half of the unit interval into $[\frac{1}{2}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$. Since this caret has a left child, the interval $[\frac{1}{2}, \frac{3}{4}]$ is divided into $[\frac{1}{2}, \frac{5}{8}]$ and $[\frac{5}{8}, \frac{3}{4}]$. So, the word $x_1x_0^{-1}$ corresponds to the homeomorphism with the singularities $\{(0,0), (\frac{1}{4}, \frac{1}{2}), (\frac{1}{2}, \frac{5}{8}), (\frac{3}{4}, \frac{3}{4}), (1,1)\}$. Also, given the singularities of a homeomorphism in F, we can determine the T_- and T_+ trees that divide the unit interval to get a tree pair diagram (T_-, T_+) that corresponds to the given homeomorphism.

3.3. **Group operations.** The group operations in the algebraic and analytic interpretations are straightforward. They are group multiplication as defined by the relators and function composition, respectively. However, it is not so obvious what the corresponding group operation should be for the geometric interpretation of F. It helps to make the connection between function composition and the group operation for tree pair diagrams. We described above how a tree pair diagram (T_-, T_+) determines a homeomorphism of the unit interval. With that in mind, we see that the T_- tree corresponds to the domain of the homeomorphism, since it gives the x-coordinates of the singularities of the homeomorphism, while the T_+ tree corresponds to the range, since it gives the y-coordinates of the singularities of the homeomorphism. In the analytic interpretation, the homeomorphisms are from the unit interval to itself, so the domain and range are the same, and we can compose any two homeomorphisms. When we have two tree pair diagrams (T_-, T_+) and (S_-, S_+), if T_+ and S$_-$ are equal (i.e. if the appropriate domain and range are equal), then group multipilication produces the tree pair diagram (T_-, S_+), which may be unreduced. However, T_+ and S_- may be different. In this case, we can add carets creating unreduced elements until T_+ and S_- are identical.

We add carets to the trees in the following way. We look to see if the T_+ tree has the same carets as S_-. When it does not have a caret in the same position as a caret in S_-, we add it in. This creates a caret with leaves numbered k and $k+1$ in T_+. To make sure we do not change the element, we add a caret in T_- to the leaf numbered k. This will create a new caret in T_- with leaves numbered k and $k+1$. This tree pair diagram will be unreduced, but when we reduce it, we will be left with (T_-, T_+). We follow this process for the rest of the carets in T_+, and then look to see if S_- has all the carets that T_+ has, adding carets when needed in a similar fashion. This process will yield unreduced elements (T'_-, T'_+) and (S'_-, S'_+), but they represent the same group elements as (T_-, T_+) and (S_-, S_+). Thus, multiplication of the elements $(T_-, T_+$ and (S_-, S_+) gives (T'_-, S'_+), which may be unreduced. We refer the reader to Figure 3.4 for an example.

4. Fordham's Method for 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 generating set $\{x_0, x_1\}$. That is, $|w|$ is equal to the least number of generators from the set $\{x_0^{\pm 1}, x_1^{\pm 1}\}$ needed to express w as a string of generators. We note that finding the length of a minimal path in the Cayley graph from the identity to w is the same as finding $|w|$. Although there is no obvious

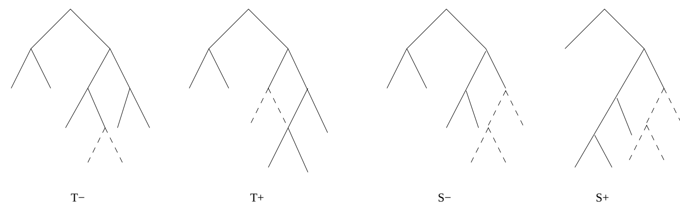

FIGURE 3.4. On the left is a tree pair diagram (T_-, T_+) and on the right is a tree pair diagram (S_-, S_+). The dotted carets are the carets added to make tree pair diagrams (T'_-, T'_+) and (S'_-, S'_+), in which T'_+ and S'_- are equal to each other. Multiplying these tree pair diagram yields (T'_-, S'_+).

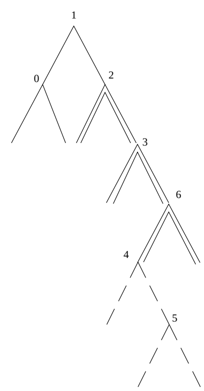

FIGURE 4.1. A tree pair diagram where left carets have single-lined edges, right carets have double-lined edges, and interior carets have dashed edges. The caret types in increasing order of caret numbering is L_0, L_L, R_{NI}, R_I, I_R, I_0, and R_0.

connection between $|w|$ and the tree pair diagram of w, Fordham presents a method to calculate word length based only on the carets in the tree pair diagram of w in [6].

First, we classify the carets in a tree pair diagram (T_-, T_+) into seven types. A caret in a tree is a *left caret* if its left edge is on the left side of the tree. A caret is a *right caret* if its right edge is on the right side of the tree, and its left edge is not on the left side of the tree. Thus the root caret is considered to be a left caret. If a caret is neither a right nor a left caret, then it is an *interior caret*. We refer the reader to Figure 4.1.

A *left child* of a caret c is the caret that is connected to the left edge of c. A *right child* is defined similarly. We number the carets using the infix ordering starting with 0. The infix ordering is done as follows: the left child of a caret is numbered first, then the caret, and then right child of the caret. The reader should look at the tree pair diagram in Figure 4.2, where the carets are numbered according to the infix ordering. We classify the carets into seven disjoint types:

(1) L_0: The first caret of the left side of the tree, with caret number 0.
(2) L_L: Any left caret other than L_0.

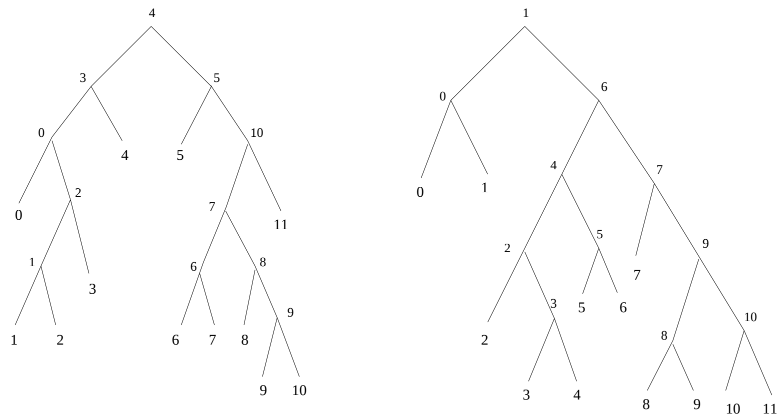

FIGURE 4.2. The reduced tree pair diagram corresponding to the element $x_0x_2^2x_3x_5x_8x_9^{-1}x_8^{-1}x_6^{-2}x_1^{-2}x_0^{-2}$.

(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: A right caret numbered k such that the caret numbered $k+1$ is an interior caret.
(6) R_{NI}: A right caret numbered k which is not an R_I caret but for which there is a higher numbered interior caret.
(7) R_0: A right caret with no higher-numbered interior carets.

Note that the root caret is considered an L_L caret if the root caret has left children, otherwise it is an L_0 caret. In Figure 4.2, the caret types in T_- in increasing order are L_0, I_0, I_0, L_L, L_L, R_I, I_0, I_R, I_R, I_0, R_0. The caret types in T_+ in increasing order are L_0, L_L, I_R, I_0, I_R, I_0, R_{NI}, R_I, I_0, R_0, R_0.

To calculate the word length of an element w in Thompson's group we need to consider its reduced tree pair diagram (T_-, T_+). We create caret pairs (τ_k, σ_k), where τ_k is the caret type of the caret numbered k in T_- and σ_k is the caret type of the caret numbered k in T_+. Each pair has a weight, which is determined from Table 4, below. The caret pair (L_0, L_0) is defined to have a weight of zero. Fordham's result is that the length of w in the word metric arising from the finite generating set is exactly the sum of the weights of the caret pairs.

	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

Table 4: The weights of all the caret pairs.

Fordham proves the following theorem.

Theorem 4.1. *(Fordham [6], Theorem 2.5.1) Given an element $w \in F$ described by the reduced tree pair diagram (T_-, T_+), the length $|w|$ of the word w with respect to* *the generating set* $\{x_0, x_1\}$ *is the sum of the weights of the caret pairings in* (T_-, T_+).

Consider the word $x_0x_2^2x_3x_5x_8x_9^{-1}x_8^{-1}x_6^{-2}x_1^{-2}x_0^{-2}$, whose tree pair diagram is given in Figure 4.2. As in any tree pair diagram, the caret pair (τ_0, σ_0) is of type (L_0, L_0), which has a weight of 0. The caret pair (τ_1, σ_1) is of type (I_0, L_L), which according to table 4 has weight 2. Continuing in this manner, we see that the sum of the weights of the weights of the caret types is $0+2+4+2+2+3+1+3+4+1+0 = 22$. According to Theorem 4.1, this is the word length of $x_0x_2^2x_3x_5x_8x_9^{-1}x_8^{-1}x_6^{-2}x_1^{-2}x_0^{-2}$.

The reason that we can ascertain properties about the Cayley graph of F with respect to the generating set $\{x_0, x_1\}$ is that the interpretation of Thompson's group through tree pair diagrams and Fordham's method for calculating word length give us powerful tools to study it. By deducing some properties of this graph, we hope to understand it better.

5. Effect of Muliplication by a Generator on an Element In Thompson's Group

We now describe the effect of multiplying an element in F by one of the generators x_0^{-1}, x_0, x_1^{-1}, and x_1 on a tree pair diagram. First, we state a lemma due to Fordham, which states that under certain conditions, multiplying by a generator will affect the caret type of exactly one caret pair.

Lemma 5.1 ([6], *Lemma 2.3.1*)**.** *Let* (T_-, T_+) *be a reduced pair of trees, each having* $m+1$ *carets, representing an element* $w \in F$*, and let* α *be any generator of* F*.*

(1) *If* $\alpha = x_0$*, we require that the left subtree of the root of* T_- *is nonempty.*
(2) *If* $\alpha = x_0^{-1}$*, we require that the right subtree of the root of* T_- *is nonempty.*
(3) *If* $\alpha = x_1$*, we require that the left subtree of the right child of the root of* T_- *is nonempty.*
(4) *If* $\alpha = x_1^{-1}$*, we require that the right subtree of the right child of the root of* T_- *is nonempty.*

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

We can easily determine which caret changes caret type when we multiply by a particular generator. First, we define C_R to denote the caret which is the right child of the root caret of T_-, and C_L to be the left child of the root of T_-.

Consider $w \in F$ written in normal form with tree pair diagram (T_-, T_+) satisfying the condition in Lemma 5.1 for $\alpha = x_0^{-1}$. Then, the right subtree of the root of T_- is nonempty. Let k be the caret number of C_R. Multiplying w by x_0^{-1} increases the exponent of the leaf numbered 0 in T_- by one, leaving the exponents of the other leaves unchanged. This means that we are adding a left caret to T_-, but otherwise keeping all the subtrees of T_- the same. This changes the caret numbered k in T_- from a right caret to a left caret. By Lemma 5.1, this is the only caret that changes type. Geometrically we can interpret this as a counter-clockwise rotation of T_- around the root caret. We can continue this reasoning for w with tree pair diagram (T_-, T_+) satisfying the condition in Lemma 5.1 for $\alpha = x_0$. In this case, the left subtree of the root of T_- is nonempty, and we let k denote the caret number of the root caret of T_-. Multiplying w by x_0 decreases the exponent of the leaf numbered 0

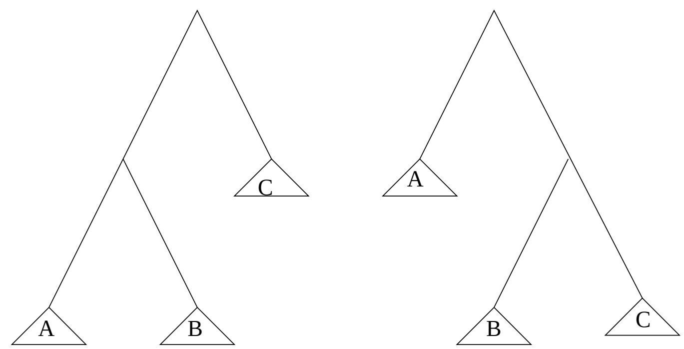

FIGURE 5.1. Let $w = (T_-, T_+)$. If the tree on the left represents T_-, where A,B, and C are possibly empty subtrees of T_-, then the tree on the right represents the negative tree of wx_0. If the tree on the right represents T_-, then the tree on the left represents the negative tree of wx_0^{-1}. In this case, the T_+ tree is not affected by multiplication by x_0 or x_0^{-1}.

by one, leaving the exponents of the other leaves unchanged. This means that we are removing a left caret of T_-. Geometrically, we interpret this as a clockwise rotation about the root caret. We refer the reader to Figure 5.1. When $w = (T_-, T_+)$ satisfies the condition of Lemma 5.1 for a generator α, then multiplying w by α does not affect T_+.

Similar reasoning can be used to describe the effect of x_1^{-1} and x_1. The difference is that instead of affecting $E(0)$, we are affecting $E(k)$ where k is the smallest numbered exposed left leaf on the right subtree of the root. This is not obvious and is proved in [4]. If w satisfies the conditions of Lemma 5.1 for $\alpha = x_1$ or x_1^{-1}, then there is only one caret pair that changes type. Applying these two observations, we see that multiplying by x_1^{-1} increases the exponent of $E(k)$ by one. Thus, if C_R is a caret numbered m, the caret numbered m in $w\alpha$ is an interior caret. This is the only change in T_-, so we can describe the effect of multiplying by x_1^{-1} as a counter-clockwise rotation around C_R. Because multiplying by x_1 decreases $E(k)$ by one, the left child of C_R changes from an interior caret to a right caret. We can describe this as a clockwise rotation around C_R, thus changing the left child of C_R from an interior caret to a right caret. We refer the reader to Figure 5.2. These observations are summarized in the following lemma.

Lemma 5.2. *([5], Lemma 2.4) If $w = (T_-, T_+) \in F$ satisfies the appropriate condition of Lemma 5.1, 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 5.1. In addition, x_1 and x_1^{-1} perform analogous operations on the subtrees of C_R as depicted in Figure 5.2.*

The trees in Figures 5.1 and 5.2 have that form because we have assumed the conditions in Lemma 5.1 for the appropriate generators. In this case, the T_+ tree is not affected when the generator is applied. When T_- does not satisfy the conditions in Lemma 5.1, we need to add in carets to both T_- and T_+, creating an unreduced tree equal to (T_-, T_+) that satisfies the conditions of Lemma 5.1. Then the rotation can be performed. The added carets have weights greater than zero, so the new word $w\alpha$ has word length strictly greater than $|w|$. Since we are most interested when

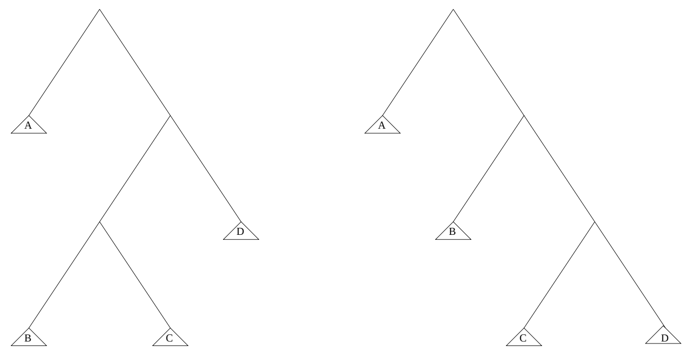

FIGURE 5.2. Let $w = (T_-, T_+)$. If the tree on the left represents T_-, where A,B,C, and D are possibly empty subtrees of T_-, then the tree on the right represents the negative tree of wx_1. If the tree on the right represents T_-, then the tree on the left represents the negative tree of wx_1^{-1}. In this case, the tree T_+ is not affected by multiplication by x_1 or x_1^{-1}.

multiplying by α decreases word length, we assume from now on that w satisfies the conditions of Lemma 5.1 for the appropriate generator.

6. Nested Traversal Method

Suppose G is a group with finite generating set S. A *minimal length representative* of an element $w \in G$ with respect to S is a string of generators $\gamma = w$ such that the number of generators in γ is equal to the length of w. Finding a minimal length representative for an element is equivalent to finding a minimal path in the Cayley graph from the identity to the element. If we have a minimal length representative for an element, we can view this product of generators as a set of instructions for which edge to take in the Cayley graph to get to the vertex representing w. So, if the first generator in the minimal length representative is α_1, we start at the identity in the Cayley graph and go over the edge that represents the multiplication by α_1 to get to the vertex α_1. If the second generator in the minimal length representative is α_2, then we start at the vertex α_1 and go over the edge representing multiplication by α_2. When we follow the directions for each generator in the minimal length representative in order, we will arrive at w. Conversely, if we had a minimal path to w, we can travel the path and write down the generators used to get to w to obtain a minimal length representative. For example, a minimal length representative for the element $ab \in S_3$ with respect to the generating set $\{a, b\}$, is ab. On the Cayley graph shown in Figure 2.3, we see that if we start at the identity and travel by edge a and then b, we get to the element ab. Furthermore, looking at the Cayley graph, we see that there is another path of length two from the identity to ab. This path travels by b and a^{-1}, and so we can conclude that ba^{-1} is a minimal length representative for ab.

Using Fordham's method for calculating word length, we can find a minimal length representative for an element $w \in F$ in the following way. We first find a generator α_1 such that $|w\alpha_1| = |w| - 1$, trying all four generators if necessary. Then find another generator α_2, such that $|w\alpha_1\alpha_2| = |w| - 2$. Continue this process for $\alpha_1\alpha_2 \cdots \alpha_n$, until $|w\alpha_1 \cdots \alpha_n| = 0$. Then inverting the string of generators will produce a minimal length representative for w. However, there is no efficient way of choosing generators, so this method is better suited for a computer program. Fordham uses this method in a LISP program to construct minimal length representatives.

While a method for constructing minimal length elements for a generic element in F based on its tree pair diagram is not yet known, there is a method which constructs minimal length representatives for certain types of elements. An element $w \in F$ is *negative* if the exponents of all the terms in the normal form of w are negative. A positive element can be defined in similar fashion. If $w = (T_-, T_+)$ is a negative word, then T_+ is the tree consisting of the root caret followed by R_0 carets, i.e, the unique tree pair diagram where $E(k) = 0$ for all leaves k. We will denote this tree by $*$. Cleary and Taback devised a method which will produce a minimal length representative for a negative word or positive word directly from the tree pair diagram [5]. They call it the nested traversal method.

The nested traversal method begins by considering the tree pair diagram $(T_-, *)$ of a negative element w. Creating each caret type requires a certain sequence of generators, which is listed in Table 6. The generators x_0 and x_0^{-1} change the caret types of the root caret and the right child of the root caret C_R, respectively, in T_-. The generators x_1^{-1} and x_1 changes the caret types of C_R and the left child of C_R, respectively, in T_-. This means that there are only a few positions in the tree pair diagram that the generators affect. With this in mind, we describe a method of producing minimal length representatives for elements in F, one caret at a time, in increasing order of caret numbers. We start with the $*$ tree and will manipulate it via multiplication by generators x_0, x_0^{-1}, and x_1^{-1} until it looks like T_-.

We first determine the string of generators needed to create each caret type. Consider the tree pair diagram $(T_-, *)$ and let τ be the right child of the root caret. Suppose we wanted τ to be of type L_L. Multiplying by x_0^{-1} will move τ into the root position, making it a left caret. If we wanted τ to be of type I_0, then multiplying by x_1^{-1} will make τ an I_0 type caret. If we wanted τ to be of type R_0, no generators are necessary, because τ is of type R_0. Suppose τ is to be of type R_I or R_{NI}. The τ caret is in the correct position, but higher numbered interior carets need to be created. In this case, we apply $x_0^{-1} \cdots x_0$, where multiplying by x_0^{-1} changes the tree so the higher numbered carets can be affected by multiplication by a generator and multiplying by x_0 makes τ a right caret again. The ellipses in this string represent the generators necessary to create the rest of the right side of T_-, including interior carets. If we wanted τ to be of type I_R, we run into a similar situation we had when trying to create R_{NI} and R_I type carets, since I_R is also defined by a higher numbered caret. So we use $x_0^{-1} \cdots x_0$, where the ellipses represent the generators necessary to create the right subtree of I_R. Then, we multiply by x_1^{-1}, making τ an I_R type caret. As an example, suppose we start with the $*$ tree and wish to get the tree in Figure 6.1. We could begin by multiplying by x_1^{-1}, which will make the caret numbered one into an interior caret, as desired. We can then multiply by x_1^{-1} to make the caret numbered two an interior caret, however it will not be the right child of the caret numbered one. So instead, we first multiply by x_0^{-1}, moving the caret numbered two into the C_R position. Now, we can multiply by x_1^{-1}, which makes the caret numbered two into an interior caret. Then we multiply by x_0, which will move the caret numbered one into the C_R position. Multiplying by x_1^{-1} will finish the sequence of generators needed to create the desired tree. We note that no generators

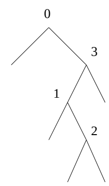

Figure 6.1. The T_- tree for the negative element $x_2^{-1}x_1^{-1}$.

are necessary to create an L_0 type caret because the caret numbered zero will always be a left caret. This gives us the generators necessary to create each caret type. The results are summarized in Table 6.

Now that we have the sequence of generators to create each caret type, we need to give the order in which to use them. We begin with the tree $*$ as the negative tree. Suppose we are looking for the sequence of generators to produce a given tree T_-. We denote by τ the right child of the root caret. The caret numbered one is τ, and we multiply by the appropriate string of generators depending on the desired caret type for τ. After multiplying by the first generator in this string, we will have a new T_- tree and the caret numbered two will be τ. We multiply by the next generator, depending on the desired caret type for τ. This gives another T_- that satisfies the condition above, and we proceed accordingly, until all the carets in T_- are of the correct type. The following theorem proved by Cleary and Taback states that this process will produce a minimal length representative for a negative element.

Theorem 6.1 (Cleary and Taback, [5]). *The nested traversal method constructs a minimal length representative for a negative element in F.*

To obtain a minimal length representative for a positive word $(*, T_+)$, we take advantage of the fact that the inverse of $(*, T_+)$ is the negative word $(T_+, *)$. So, the minimal length representative of $(*, T_+)$ is the inverse of the minimal length representative for $(T_+, *)$.

Looking at the negative word $x_9^{-1}x_8^{-1}x_6^{-2}x_1^{-2}x_0^{-2}$, whose negative tree is the negative tree in Figure 4.2, we have caret types L_0, I_0, I_0, L_L, L_L, R_I, I_0, I_R, I_R, I_0, R_0. Following the nested traversal method, we start with a tree with one L_0 caret and nine R_0 carets. We first multiply by x_1^{-2} to make carets numbered one and two into I_0 type carets. Carets numbered three and four are to be L_L type carets, so we multiply by x_0^{-2}. This brings the caret numbered five into the C_R position. Since this caret is to be an R_I caret, we need to multiply by x_0^{-1}, which brings the caret numbered six into the C_R position. Multiplying this by x_1^{-1} makes the caret numbered six into an I_0 caret. This will bring the caret numbered seven into the C_R position, and continuing on with the nested traversal method will produce the minimal length representative $x_1^{-2}x_0^{-3}x_1^{-1}x_0^{-2}x_1^{-1}x_0x_1^{-1}x_0x_1^{-1}x_0$. The minimal length representative for the positive word $x_0x_2^2x_3x_5x_8$, whose positive tree is T_+ in Figure 4.2, is $x_0^{-2}x_1x_0^2x_1x_0^{-1}x_1x_0x_1x_0^{-1}x_1x_0^3$.

Caret type	Generators involved in creation of caret
L_0	none
L_L	x_0^{-1}
I_0	x_1^{-1}
I_R	$x_0^{-1}...x_0x_1^{-1}$
R_0	none
R_{NI}	$x_0^{-1}...x_0$
R_I	$x_0^{-1}...x_0$

Table 6: The sequence of generators necessary to create each caret type in T_- for the Nested Traversal Method [5].

7. An Extension Of The Nested Traversal Method

The nested traversal method can be applied to create a sequence of generators equal to any element $w \in F$ with tree pair diagram (T_-, T_+) by using the nested traversal method first on $(*, T_+)$, and on $(T_-, *)$, and finally combining the two strings of generators. This will give a string of generators equal to w, but it will not necessarily be a minimal length representative for w. We define $|(T_-, *)|$ to be the word length of the negative word whose tree pair diagram is $(T_-, *)$. Similarly, $|(*, T_+)|$ is defined to be the length of the positive word whose tree pair diagram is $(*, T_+)$. By the triangle inequality, $|(T_-, T_+)| \leq |(*, T_+)| + |(T_-, *)|$. When $|(T_-, T_+)| = |(*, T_+)| + |(T_-, *)|$, doing the nested traversal method on w as above will yield a minimal length representative for w. We will give a necessary and sufficient condition for when a word w with tree pair diagram.

First, note that certain caret pairs in $(T_-, T_{+)}$ satisfy the condition $wt(\tau_i, \sigma_i) = wt(\tau_i, R_0) + wt(\sigma_i, R_0)$, where $wt(\tau_i, \sigma_i)$ is the weight of the pair of caret types (τ_i, σ_i) in Fordham's method for calculating word length. When we perform the nested traversal method on w as above, we use $wt(R_0, \sigma_i)$ generators to create a σ_i type caret in T_+ and use $wt(\tau_i, R_0)$ generators to create a τ_i type caret in T_-. If the sum of these two totals is $wt(\tau_i, \sigma_i)$ for every caret pair in (T_-, T_+), then we have used the correct number of generators to create each caret pair, meaning we have minimal length representative for w. Our claim is that applying the nested traversal method as above works for words whose tree pair diagrams contain only these caret pairs. We refer the reader to Table 7 for a list of all the caret pairs relating $wt(\tau, \sigma)$ to $wt(\tau, R_0) + wt(\sigma, R_0)$.

Caret pair (τ_i, σ_i)	
$(R_0, *)$	$wt(R_0, *) = wt(R_0, R_0) + wt(*, R_0)$
(I_0, R_I)	$wt(I_0, R_I) = wt(I_0, R_0) + wt(R_I, R_0)$
(L_L, L_L)	$wt(L_L, L_L) = wt(L_L, R_0) + wt(L_L, R_0)$
(L_L, I_0)	$wt(L_L, I_0) = wt(L_L, R_0) + wt(I_0, R_0)$
(I_0, I_*)	$wt(I_0, I_*) = wt(I_0, R_0) + wt(I_*, R_0)$
(R_{NI}, R_{NI})	$wt(R_{NI}, R_{NI}) < wt(R_{NI}, R_0) + wt(R_{NI}, R_0)$
(R_I, R_{NI})	$wt(R_I, R_{NI}) < wt(R_I, R_0) + wt(R_I, R_0)$
(L_L, R_{NI})	$wt(L_L, R_{NI}) < wt(L_L, R_0) + wt(R_{NI}, R_0)$
(I_*, R_{NI})	$wt(I_*, R_{NI}) < wt(I_*, R_0) + wt(R_{NI}, R_0)$
(R_I, R_I)	$wt(R_I, R_I) < wt(R_I, R_0) + wt(R_I, R_0)$
(L_L, R_I)	$wt(L_L, R_I) < wt(L_L, R_0) + wt(R_I, R_0)$
(I_R, R_I)	$wt(I_R, R_I) < wt(I_R, R_0) + wt(R_I, R_0)$
(L_L, I_R)	$wt(L_L, I_R) < wt(L_L, R_0) + wt(I_R, R_0)$
(I_R, I_R)	$wt(I_R, I_R) < wt(I_R, R_0) + wt(I_R, R_0)$

Table 7: The list of all caret pairs and the relation between the weight of the caret pairs to the weight of each caret type paired with R_0. Since $wt(\tau, \sigma) = wt(\sigma, \tau)$, the relation between the weight of (τ, σ) to the weight of each caret type paired with R_0 is the same as the corresponding relation with (σ, τ).

Theorem 7.1. *Let w be an element in Thompson's group with tree pair diagram (T_-, T_+), let β be the string of generators created by performing the nested traversal on $(*, T_+)$, and let α be the string of generators created by performing the nested traversal method on $(T_-, *)$. Then $\beta\alpha$ is a minimal length representative for w if and only if $wt(\tau, \sigma) = wt(\tau, R_0) + w(\sigma, R_0)$ for all caret pairs (τ, σ) in (T_-, T_+).*

Proof. Suppose that all the carets in (T_-, T_+) satisfy the equation

$$wt(\tau_i, \sigma_i) = wt(\tau_i, R_0) + wt(\sigma_i, R_0)$$

Since the caret pair numbered zero (L_0, L_0) always has weight zero, we can in fact start our indexing at one. Let n be the highest numbered caret pair in (T_-, T_+). From Fordham's method of calculating word length, $|w| = \sum_1^n wt(\tau_i, \sigma_i)$, $|(T_-, *)| = \sum_1^n wt(\tau_i, R_0)$, and $|(*, T_+)| = \sum_1^n wt(R_0, \tau_i)$. With the condition on the weights of the carets, we have that $|w| = |(T_-, *)| + |(*, T_+)|$. Since β creates the positive part of w and α creates the negative part of w, $\beta\alpha = w$. However, the number of generators in $\beta\alpha$ is equal to $|w|$. So, $\beta\alpha$ is a minimal length representative for w.

For the other direction, we prove the contrapositive. So, suppose $w = (T_-, T_+)$ contains a caret (τ, σ) such that $wt(\tau, \sigma) \neq wt(\tau, R_0) + w(\sigma, R_0)$. Then, we see from Table 7, that $wt(\tau, \sigma) < wt(\tau, R_0) + w(\sigma, R_0)$. Using Fordham's method for calculating word length, we see that $|w| < |(T_-, R_0)| + |(T_+, R_0)|$. So, if we apply the nested traversal method to each tree and combine the string of generators from each application, then we have a string of generators equal to w, where the number of generators in the string is greater than $|w|$. Hence, the nested traversal method used on the positive and negative parts of the word independently did not produce a minimal length representative. ☐

We would like to point out that, using the word metric for Cayley graphs, an element with a tree pair diagram (T_-, T_+) always satisfies the inequality $|w| \leq |(T_-, *)| + |(*, T_+)|$ because of the triangle inequality. Our theorem deals with case

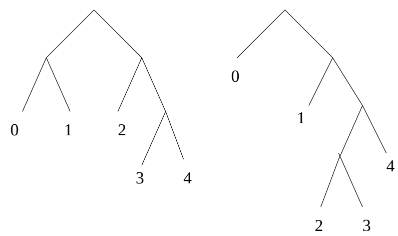

FIGURE 7.1. The T_- tree for the element $x_2x_0^{-1}$. Doing the extension of the nested traversal method for this element will not yield a minimal length representative.

when we have equality. As we see from the table, all positive and negative words satisfy the condition of Theorem 7.1.

We are now able to construct minimal length representatives for a larger class of words than before. The method constructs a minimal length representative for $w \in F$ with a tree pair diagram (T_-, T_+) by first creating $(*, T_+)$, then $(T_-, *)$, and finally combining those strings of generators to get (T_-, T_+). However, this method does not construct a minimal length representative for every element. For a simple example, consider the element $x_2x_0^{-1}$, whose tree pair diagram is shown in Figure 7.1. Notice that the first caret pair the tree pair diagram is of the type (L_L, R_I), which does not satisfy the hypothesis in Theorem 7.1. Using the extension of the nested traversal method we get that the string of generators $x_0^{-1}x_1x_0x_0^{-1}$ is equal to $x_2x_0^{-1}$. This clearly cannot be a minimal length representative. In other words, creating each tree separately is not the most efficient way to construct (T_-, T_+) for every element. There must be a way to construct T_- and T_+ at the same time.

REFERENCES

[1] K. S. Brown and R. Geoghegan. An infinite-dimensional torsion-free FP_∞ group. *Inventiones mathematicae*, 77:267-381, 1984
[2] J. Burillo, S. Cleary, and M. I. Stein. Metrics and embeddings of generalizations of Thompson's group F. *Trans. Amer. Math. Soc.*, 353(4):1677-1689 (electronic), 2001
[3] J. W. Cannon, W. J. Floyd, and W. R. Parry. Introductory notes on Richard Thompson's groups. *L'Ens. Math*, 42:215-256, 1996
[4] S. Cleary and J. Taback. Thompson's group F is not almost convex *J. Algebra*, Vol. 270 No. 1 (2004), pp 133-149
[5] S. Cleary and J. Taback. Combinatorial Properties of Thompson's group F. *Trans. Amer. Math. Soc.*, 356(7):2825-2849 (electronic), 2004
[6] Blake Fordham. *Minimal Length Elements of Thompson's group F*. *Geom. Dedicata* 99 (2003), 179–220