# Metric Properties of Thompson's Groups $F(n)$ and $F(n, m)$

by

Claire W. Wladis

A Dissertation submitted to the Graduate Faculty
in Mathematics in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2007

UMI Number: 3283170

UMI Microform 3283170



ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the
Graduate Faculty in Mathematics in satisfaction of the
dissertation requirement for the degree of Doctor of Philosophy.

Sean Cleary

Date

Chair of Examining Committee

Josef Dodziuk

Date

Executive Officer

Gilbert Baumslag

Vladimir Shpilrain

Sean Cleary

Supervisory Committee

The City University of New York

Abstract

Metric Properties of Thompson's Groups $F(n)$ and $F(n,m)$

by

Claire W. Wladis

Adviser: Professor Sean Cleary

We prove several metric properties of Thompson's group $F(n)$ with respect to the standard finite generating set $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$. We prove that seesaw words exist in $F(n)$ with respect to $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$and therefore that $F(n)$ is not combable by geodesics and that there does not exist a regular language of geodesics for $F(n)$. We also prove that dead ends exist in $F(n)$ with respect to $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$and that these dead ends all have depth 2. Both these results were proven for the case $n = 2$ (i.e. when $F(n) = F$) by Cleary and Taback; our paper extends these results to $F(n)$ for $n = 2, 3, 4, ....$ We then prove that $F(n)$ is not minimally almost convex with respect to $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$. Belk and Bux proved this for $n = 1$. We follow the outline of their proof, generalizing it to $F(n)$ for all $n \in \{2, 3, 4, ...\}$ by representing elements of $F(n)$ as tree-pair diagrams rather than forest diagrams in order to use Fordham's metric on $F(n)$.

We also prove several metric properties of Thompson's group $F(n,m)$ where $n, m \in \{2, 3, 4, ...\}$. We highlight several differences between $F(n)$ and $F(n,m)$, including the fact that minimal tree-pair diagram representatives of elements of $F(n,m)$ may not be unique. We establish how to find minimal tree-pair diagram representatives of elements of $F(n,m)$ and give two different ways to find unique minimal representatives for elements of the group. We establish how tree-pair diagrams can be composed in order to perform group multiplication, and we prove several theorems describing the equivalence of trees and tree-pair diagrams. We introduce a normal form for elements of $F(2,3)$ with respect to the standard infinite generating set. We also give bounds for the metric of $F(2,3)$ with respect to the standard finite generating set and show that the order of these bounds is sharp, showing that the metric on $F(2,3)$ is not quasi-isometric to the number of leaves or carets in the minimal tree-pair diagram representative, as is the case in $F(2)$.

INDEX WORDS: Thompson's group, Higman group, Stein group

# Acknowledgments

I would like to thank my adviser Sean Cleary for taking me seriously as a mathematician, for mentoring me through my first mathematics research, and for taking the time to give advice and to critique my work, even when he was traveling all over the world on his sabbatical. I'd like to thank the other members of my supervisory committee for their generosity in contributing their time and energy to my work: Vladimir Shpilrain and Gilbert Baumslag, whose group theory class first inspired my love of the subject. I'd like to thank Jose Burillo in particular for his support, and also Melanie Stein and Jen Taback for giving me feedback and ideas on my work and for sharing their research with me. I'd like to thank my colleagues at BMCC for recognizing and appreciating my strengths while I was still a graduate student and for giving me the resources and support required for me to pursue my own research interests. And finally, I'd like to thank my family: my mother Carol Woods and father John Wladis, my sister Carrie Wladis, my "daughter" Miss L. B. Jones, and my husband Nicholas Rennie for their continual support and encouragement.

# Table of Contents

Page

Acknowledgments . . . . . . . . . . vi

List of Tables . . . . . . . . . . x

List of Figures . . . . . . . . . . xi

I Metric Properties of Thompson's Group $F(n)$ 1

Chapter

1 Introduction to Thompson's Group $F(n)$ . . . . . . . . . . 2

1.1 A Description of $F(n)$ . . . . . . . . . . 2

1.2 Representation of $F(p+1)$ by tree-pair diagrams . 3

1.2.1 Definitions . . . . . . . . . . 3

1.2.2 Equivalence of Tree-Pair Diagrams and Piecewise-linear Orientation-preserving Homeomorphisms of the Closed Unit Interval . . . . . . . . . . 4

1.2.3 Leaf Ordering in a Tree-pair Diagram . . 6

1.2.4 Further Classification of Carets of Type $\mathcal{M}$ . . . . . . . . . . 6

1.2.5 Fordham's node order . . . . . . . . . . 7

1.2.6 Fordham's caret types: . . . . . . . . . . 8

1.2.7 Simplifying tree-pair diagrams . . . . . . 9

1.2.8 Multiplying tree-pair diagrams . . . . . . 10

1.3 Presentations of $F(p+1)$ . . . . . . . . . . . . . . . . 11

1.4 Normal Forms of Elements of $F(p+1)$ . . . . . . . 13

2 The Metric on $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . 17

2.1 Fordham's method for computing word length in $F(p+1)$. . . . . . . . . . . . . . . . . . . . . . . . . 17

3 Seesaw words . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3.1 Definitions and Background . . . . . . . . . . . . . 20

3.2 Seesaw words exist in $F(p+1)$ . . . . . . . . . . . . . 21

3.3 Regular languages of geodesics in $F(p+1)$ . . . . 31

4 Dead Ends in Thompson's group $F(p+1)$ . . . . . . . . . . . 33

4.1 Dead Ends and K-pockets . . . . . . . . . . . . . . . 33

4.2 Dead Ends in $F(p+1)$ . . . . . . . . . . . . . . . . . . . 34

4.3 Depth of Dead Ends in $F(p+1)$ . . . . . . . . . . . . 42

5 Thompson's group F(p+1) is not Minimally Almost Convex 45

5.1 Almost Convexity . . . . . . . . . . . . . . . . . . . . . . 45

5.2 Thompson's group $F(p+1)$ is not minimally almost convex . . . . . . . . . . . . . . . . . . . . . . . . . . 46

5.2.1 Any Path from $l$ to $r$ which passes through vertices with distance at least $2n+3$ apart must leave $B_{2n+2}$ . . . . . . . . . . . . 50

5.2.2 Finding the vertex $h_r$ . . . . . . . . . . . . . 51

5.2.3 Finding the vertex $h_l$ . . . . . . . . . . . . . 61

II METRIC PROPERTIES OF THOMPSON'S GROUP $F(n, m)$ 83

6 INTRODUCTION TO THOMPSON'S GROUP $F(n, m)$ . . . . . . . . . 84

7 REPRESENTING ELEMENTS OF $F(p+1, q+1)$ USING TREE-PAIR DIAGRAMS . . . . . . . . . . . . . . . . . . . . . . . . . . 87

7.1 EQUIVALENCE OF TREES AND TREE-PAIR DIAGRAMS . . 95

7.2 FINDING MINIMAL LEAF-PATH CHARTS AND TREE-PAIR DIAGRAMS . . . . . . . . . . . . . . . . . . . . . . . . 128

7.3 COMPOSITION OF TREE-PAIR DIAGRAMS . . . . . . . . . 131

8 ACTIONS OF GENERATORS ON TREE-PAIR-DIAGRAMS . . . . . . 138

8.1 PRESENTATIONS OF $F(2, 3)$ . . . . . . . . . . . . . . . . 138

8.2 ACTIONS OF FINITE GENERATORS ON $(2, 3)$–ARY TREE-PAIR DIAGRAMS . . . . . . . . . . . . . . . . . . . . . . 141

9 A NORMAL FORM FOR ELEMENTS OF $F(2, 3)$ . . . . . . . . . . . 142

10 THE METRIC ON $F(2, 3)$ . . . . . . . . . . . . . . . . . . . . . . 180

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . . . . . . . 211

# List of Tables


2.1 Weight of Types of Caret Pairs in the Tree-Pair Diagram of Elements in $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

7.1 Leaf-path chart for tree-pair diagram given in Figure 7.17 . . . . . . 110

7.2 Leaf-path chart for tree-pair diagrams given in Figure 7.18 . . . . . 111

7.3 Leaf-path chart for tree-pair diagram given in Figure 7.19 . . . . . . 112

7.4 Leaf-path chart for tree-pair diagrams given in Figure 7.4 . . . . . . 114

10.1 Leaf-path chart for minimal tree-pair diagram representative of $y_0$ in $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182

10.2 Leaf-path chart for minimal tree-pair diagram representative of $y_0^n$ in $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182

# List of Figures


1.1 The homeomorphism of the closed unit interval and the tree-pair diagram representing $x_0$ in $F(p+1)$ . . . . . 5

1.2 Types of child carets . . . . . 7

1.3 Ordering of child nodes with respect to parent caret type . . . . . 8

1.4 Multiplication of tree-pair diagrams for the product $x_0 x_p$ in $F(p+1)$ 12

1.5 The generators $\{x_0, x_1, x_2, ...\}$ for the standard infinite presentation of $F(p+1)$ . . . . . 13

1.6 The generators $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$ for the standard finite presentation of $F(p+1)$ . . . . . 14

1.7 $x_1 x_3^5 x_4^{-1} x_0^{-3}$ in $F(p+1)$ with all carets and leaves numbered . . . . 16

3.1 Seesaw elements in $F(p+1)$ . . . . . 23

3.2 $wx_0^{-q}$ (when $0 \leq q < r$) in $F(p+1)$ . . . . . 25

3.3 $wx_0^{q}$ (when $0 \leq q < l$) in $F(p+1)$ . . . . . 26

3.4 Symmetric Seesaw Words in $F(p+1)$ . . . . . 29

4.1 Form of All Dead Ends in $F(p+1)$ . . . . . 35

4.2 Form of Negative Tree of Dead End in $F(p+1)$ . . . . . 36

4.3 $wx_0^{-1}$ for $w$ a dead end in $F(p+1)$ . . . . . 43

4.4 $wx_0^{-1}x_i$ for $i = 1, 2, ..., p$ and $w$ a dead end in $F(p+1)$ . . . . . 44

5.1 $r$ in $F(p+1)$ . . . . . 48

5.2 $l$ in $F(p+1)$ . . . . . 49

5.3 $rx_p x_0^{-1}$ in $F(p+1)$ . . . . . 55

5.4 $rx_p x_i$ in $F(p+1)$ . . . . . 56

5.5 $rx_px_i^{-1}$ in $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . . . . 58

5.6 $rx_p^2$ in $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . . . . . . 60

5.7 $h_{ll}$ in $F(p+1)$ after carets have been added as needed so that $T^-$ satisfies the required conditions of Theorem 2.1.2 . . . . . . . . . . 66

5.8 $y$ in $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

5.9 General form of left-sided elements in $F(p+1)$ . . . . . . . . . . . . 71

5.10 $yf = fy$ for left-sided element $f$ in $F(p+1)$ . . . . . . . . . . . . 72

5.11 Rewritten version (see proof of Lemma 5.2.14) of $h_{ll}$ in $F(p+1)$ after carets have been added as needed so that $T^-$ satisfies the required conditions of Theorem 2.1.2 . . . . . . . . . . . . . . . . . . . . 74

5.12 $h_r^{-1}h_l'$ in $F(p+1)$ . . . . . . . . . . . . . . . . . . . . . . . . . 82

7.1 One tree-pair diagram for the identity element in $F(2,3)$ . . . . . . 87

7.2 Another tree-pair diagram for the identity element in $F(2,3)$ . . . . 88

7.3 Two equivalent but distinct minimal tree-pair diagrams representing an element of $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . 89

7.4 Two equivalent but distinct minimal tree-pair diagrams representing another element of $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . 90

7.5 $(T_-^1, T_+^1)$ is a (2,3)–ary tree-pair diagram which must have carets added to it in order to be transformed into its equivalent minimal tree-pair diagram $(T_-^4, T_+^4)$ . . . . . . . . . . . . . . . . . . . . . 91

7.6 $(T_{0-}, T_{0+})$ is a (2,3)–ary tree-pair diagram which must have carets added to it in order to be transformed into its equivalent minimal tree-pair diagram $(T_{7-}, T_{7+})$ . . . . . . . . . . . . . . . . . . . . 93

7.7 The step-by-step transformation using substitutions of the type given in Figure 7.16 taking the tree $T_1$ to the tree $T_6$ . . . . . . . . 94

7.8 Two equivalent but distinct tree-pair diagrams, each with a different number of leaves in each tree . . . . . . . . . . . . . . . . . . . . . 96

7.9 A (2,3)–ary tree . . . . . . . . . . . . . . . . . . . . . . . . . . . 98

7.10 The 2–ary leaf form of the tree given in Figure 7.9 . . . . . . . . . . 98

7.11 The 3–ary subtree of the tree given in Figure 7.9 . . . . . . . . . . . 99

7.12 Rewriting the tree given in Figure 7.9 as a tree with a 2–ary root caret by using the subtree substitutions given in Figure 7.16 in each step . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100

7.13 Two equivalent subtrees with 2 levels each. Here the carets composed of dotted lines are $(p + 1)$–ary and the carets composed of solid lines are $(q + 1)$–ary. . . . . . . . . . . . . . . . . . . . . . . 101

7.14 An example of a tree which satisfies the conditions of Lemma 7.1.6 but for which the two caret type still cannot be switched without producing a tree which is not equivalent to the original tree . . . . . 103

7.15 The two equivalent subtrees of Theorem 7.1.11 . . . . . . . . . . . . 107

7.16 The two equivalent subtrees of Theorem 7.1.11 when $p = 1$ and $q = 2$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108

7.17 A (2,3)–ary tree-pair diagram . . . . . . . . . . . . . . . . . . . . 109

7.18 Two (2,3)–ary tree-pair diagrams with the same leaf-path chart . . 110

7.19 A (2,3)–ary tree-pair diagram which is equivalent to the tree-pair diagram given in Figure 7.17 . . . . . . . . . . . . . . . . . . . . . 112

7.20 The (2,3)–ary tree-pair diagram which is equivalent to the leaf-path chart given in Table 7.4 . . . . . . . . . . . . . . . . . . . . . . . . 115

7.21 A (2,3)–ary tree-pair diagram . . . . . . . . . . . . . . . . . . . . 126

7.22 Composition of two elements of $F(2, 3)$ . . . . . . . . . . . . . . . . 132

8.1 The generators for the standard infinite presentation of $F(2, 3)$ . . . 138

8.2 The generators for the standard finite presentation of $F(2, 3)$ . . . . 140

8.3 The action of the generators on the negative tree of an arbitrary element of $F(2,3)$ once the tree-pair diagram for that element has had all necessary carets added so that multiplication can take place 141
9.1 General Form of Positive Element in $F(2,3)$ . . . . . . . . . . . . . 144
9.2 Minimal tree-pair diagram representative of $w_l$ in $F(2,3)$ . . . . . . 145
9.3 Minimal tree-pair diagram representative of $w_r$ in $F(2,3)$ . . . . . . 145
9.4 Minimal tree-pair diagram representative of $w_{lr}$ in $F(2,3)$ . . . . . . 146
9.5 Minimal tree-pair diagram representative of $w_i$ in $F(2,3)$ . . . . . . 147
9.6 Minimal tree-pair diagram representative of $\gamma_0^i w_i$ in $F(2,3)$ . . . . . 148
9.7 Minimal tree-pair diagram representative of $\gamma_0^i w_i$ in $F(2,3)$ . . . . . 148
9.8 Minimal tree-pair diagram representative of a positive element in $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
9.9 Minimal tree-pair diagram representative of a positive element in $F(2,3)$ where the root and the first $i+1$ right carets in the positive tree are 2–ary . . . . . . . . . . . . . . . . . . . . . . . . . . 153
9.10 Minimal tree-pair diagram representative of a positive element in $F(2,3)$ where the root and the first $i-1$ right carets in the positive tree are 2–ary . . . . . . . . . . . . . . . . . . . . . . . . . . 154
9.11 Minimal tree-pair diagram representative of an element of $F(2,3)$ where the root and all right carets in the positive tree are 2–ary and the root and the first $j-1$ right carets each have no left child . . . 156
9.12 Minimal tree-pair diagram representative of an element of $F(2,3)$ obtained by multiplying $x_j$ by an element with minimal tree-pair diagram represented by Figure 9.11 . . . . . . . . . . . . . . . . . . . 157
9.13 Minimal tree-pair diagram representative of an element of $F(2,3)$ obtained by multiplying $z_j$ by an element with minimal tree-pair diagram represented by Figure 9.11 . . . . . . . . . . . . . . . . . . . 158

9.14 Choice of the two equivalent subtrees given in Figure 7.16 for the normal tree-pair diagram representative of an element $x$ of $F(2,3)$ . 165
9.15 We replace all instances of this subtree in a minimal tree-pair diagram representative of an element of $F(2,3)$ with the equivalent subtree given in Figure 7.16 in order to obtain the normal tree-pair diagram representative . . . . . . . . . . . . . . . . . . . . . . . . . 165
9.16 Minimal tree-pair diagram representative of $y_i x_{i+1} x_i = y_i x_i x_{i+2}$ of $F(2,3)$, where the dotted caret pair shows a pair of carets that could be added to the diagram but which are not present in the minimal representative. . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
9.17 Minimal tree-pair diagram representative of $z_i x_{i+2} x_{i+1} x_i = z_i x_i x_{i+2} x_{i+4}$ of $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
9.18 Minimal tree-pair diagram representative of $y_{i+1} z_i$ of $F(2,3)$ . . . . 168
9.19 Minimal tree-pair diagram representative of $x_i z_{i+1} z_i = x_i z_i z_{i+3}$ of $F(2,3)$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
9.20 Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $y_i x_{i+1} x_i$ is present in the normal form . . . . . . . . . 170
9.21 Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $z_i x_{i+2} x_{i+1} x_i$ is present in the normal form . . . . . . 171
9.22 Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $y_{i+1} z_i$ is present in the normal form . . . . . . . . . . 171
9.23 Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $x_i z_{i+1} z_i$ is present in the normal form . . . . . . . . . 172
9.24 Tree-pair diagram representative of an element $w$ of $F(2,3)$ . . . . . 173
9.25 Minimal tree-pair diagram representative equivalent to the normal tree-pair diagram representative of the element of $F(2,3)$ given by Figure 9.24 . . . . . . . . . . . . . . . . . . . . . . . . . . . 176

10.1 The four possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_0$ . . . . 187

10.2 The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_1$ . . . . . . 189

10.3 The two possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xy_0$ . . . . . . 192

10.4 The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xy_1$ . . . . . . 193

10.5 The four possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_0^{-1}$ or $xy_0^{-1}$ 196

10.6 The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_1^{-1}$ or $xy_1^{-1}$ 196

# Part I

# Metric Properties of Thompson's Group $F(n)$

# Chapter 1

# Introduction to Thompson's Group $F(n)$

## 1.1 A Description of $F(n)$

Thompson's group $F(n)$ is a generalization of the group $F$, which R. Thompson introduced in the early 1960's (see [18]) while constructing the first example of finitely-presented infinitely simple group. $F$, which we will henceforward refer to as $F(2)$, represents the group of piecewise-linear orientation-preserving homeomorphisms of the closed unit interval with finitely-many breakpoints in $\mathbb{Z}[\frac{1}{2}]$ and slopes equal to powers of 2 in each linear piece. The generalization of $F$ to $F(n)$ is due to Brown in [2] as an extension of work done by Higman in [15]. Henceforward, we will use the convention that $n = p + 1$ where $p \in \mathbb{N}$ and write $F(p + 1)$ instead of $F(n)$. This is motivated by the fact that the use of $p + 1$ instead of $n$ simplifies some calculations later on.

Thompson's group $F(p + 1)$ (for $p \in \mathbb{N}$) can then be defined as the group of piecewise-linear orientation-preserving homeomorphisms of the closed unit interval with finitely-many breakpoints in $\mathbb{Z}[\frac{1}{p+1}]$ and slopes powers of $p + 1$ in each linear piece.

## 1.2 Representation of $F(p+1)$ by tree-pair diagrams

### 1.2.1 Definitions

An $(p+1)$–*ary caret* has $(p+2)$ vertices joined by $p+1$ edges, one with degree $p+1$ (the *parent*) and the rest with degree 1 (the *children*). Two children of the same parent are called *siblings*.

Another $(p+1)$-ary caret may then be attached to any of the $(p+1)$ children of the original caret so that the child of the original caret serves simultaneously as the parent of the new caret. A caret whose parent is also the child of a second caret will be referred to as a *child caret* of the second caret, and likewise, a caret whose child is also the parent of a second caret will be referred to as a *parent caret* of the second caret.

A graph formed by joining any number of $(p+1)$-ary carets by using the child vertex of one caret as the parent vertex of another caret is referred to as a $(p+1)$-*ary tree*. A $(p+1)$-ary tree is generally depicted so that directed edges point downward; when depicted in this way, the topmost caret is referred to as the *root caret* (or just the *root*) and it's parent is called the root node. When a $(p+1)$-ary caret is oriented in this way, its rightmost or leftmost edge is called the *right* or *left* edge, respectively. If a vertex in the tree has no children, it is referred to as a *leaf*; if it does have children, it is referred to as a *node*. For any two vertices $a$ and $b$ on a $(p+1)$-ary tree, vertex $a$ is the *ancestor* of vertex $b$ if it is on the directed path from the root node to vertex $b$. Similarly, vertex $b$ is the *descendent* of vertex $a$ if vertex $a$ is the ancestor of vertex $b$.

An element of $F(p+1)$ can be represented by a $(p+1)$-ary tree-pair diagram. A $(p+1)$-*ary tree-pair diagram* is a pair of $(p+1)$-ary trees containing the same number of nodes (or carets). The first tree in the pair is the *negative tree* and the second tree in the pair is the *positive tree*. This pair of trees is denoted $(T_-, T_+)$. The motivation for the choice of words "negative" and "positive" comes from the way the normal form of an element of $F(p+1)$ can be derived from its tree-pair diagram, which we will discuss below. The nodes of each tree are numbered (see node order, below), and the $n^{th}$ node of the negative tree is paired with the $n^{th}$ node of the positive tree.

To refer more easily to carets located in different regions of a tree, we classify the carets in each tree of a tree-pair diagram into one of 3 distinct categories or types; this is due to Fordham in [12].

1. $\mathcal{L}$. This is a left caret; a left caret is any caret that has one edge on the left side of the tree. The root caret is considered to be of this type.
2. $\mathcal{R}$. This is a right caret; a right caret is any caret (except the root caret) that has one edge on the right side of the tree.
3. $\mathcal{M}$. This is a middle caret; a middle caret is any caret that is neither a left nor a right caret.

### 1.2.2 Equivalence of Tree-Pair Diagrams and Piecewise-linear Orientation-preserving Homeomorphisms of the Closed Unit Interval

This representation of elements of $F(p+1)$ by $(p+1)$-ary tree-pair diagrams corresponds to the definition of $F(p+1)$ as the set of piecewise-linear orientation-preserving homeomorphisms of the unit interval in the following way: each leaf of each caret represents one of the linear subintervals of the closed unit interval. For example, when we consider a single root caret and number its leaves from 0 to $p$, the $i^{th}$ leaf corresponds to the subinterval of the closed unit interval $[\frac{i}{p+1}, \frac{i+1}{p+1}]$. If we then attach a caret to the leaf numbered $i$, and then number the leaves of this new caret from 0 to $p$ from left to right, the $j^{th}$ leaf corresponds to the subinterval $[\frac{j}{(p+1)^2} + \frac{i}{p+1}, \frac{j+1}{(p+1)^2} + \frac{i}{p+1}]$ of the closed interval $[\frac{i}{p+1}, \frac{i+1}{p+1}]$. This pattern can be continued with any further subdivisions by adding carets to the interval where the subdivisions are desired. For example, Figure 1.1 shows one example of how an element in $F(p+1)$ can be represented by a tree-pair diagram.

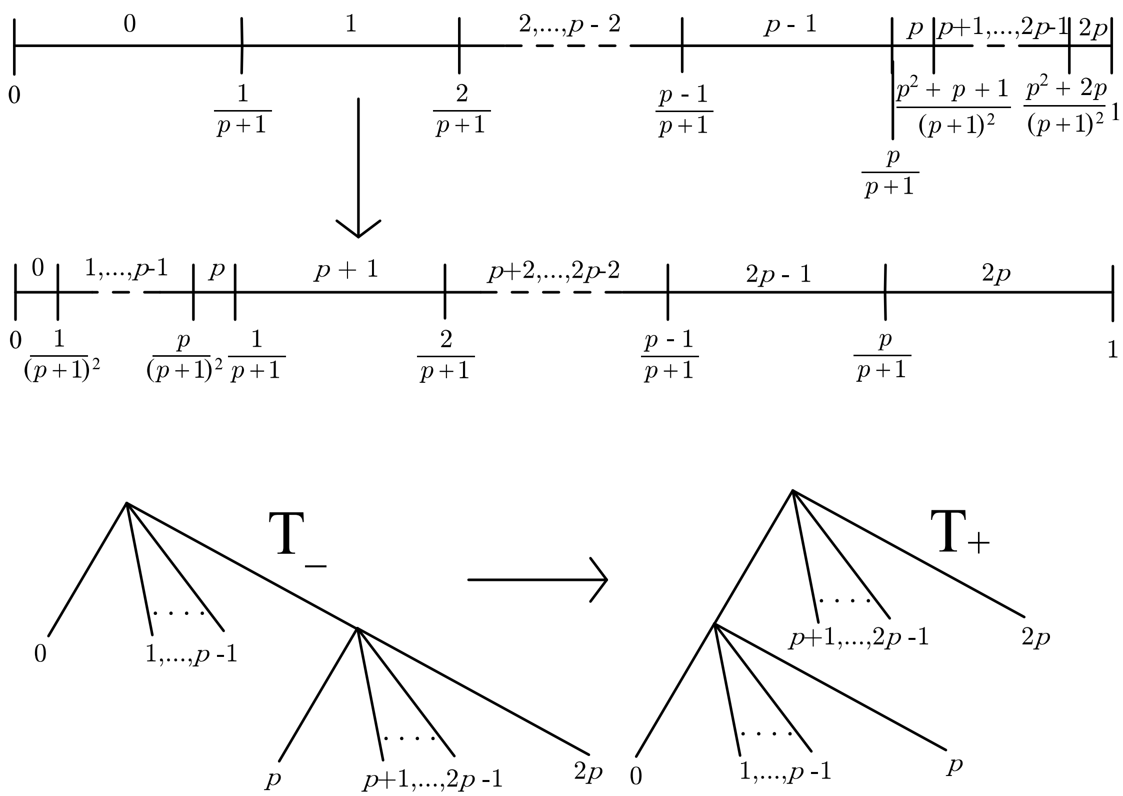


Figure 1.1: The homeomorphism of the closed unit interval and the tree-pair diagram representing $x_0$ in $F(p+1)$

### 1.2.3 Leaf Ordering in a Tree-pair Diagram

We can number the leaves of each of the trees in a tree-pair diagram of an element of $F(p+1)$ by thinking of each leaf as a subinterval of the closed unit interval; we number the leaves of the tree from left to right with respect to their position as subintervals of the closed unit interval. (See Figure 1.1)

To see an example of a tree-pair diagram with all of its leaves numbered, see Figure 1.7.

### 1.2.4 Further Classification of Carets of Type $\mathcal{M}$

Carets of type $\mathcal{M}$ can be further classified depending upon their placement with respect to other caret types in the tree. We will take all carets of type $\mathcal{M}$ and subdivide them into $p$ different subtypes each of which we will call type $\mathcal{M}^i$, for $i = 1, ..., p$. The value of $i$ depends upon the caret type of the middle caret's parent caret. To see how $i$ is determined for different parent caret types, see Figure 1.2. If the parent caret is of type $\mathcal{L}$, then the middle child carets are numbered from left to right so that the first middle child caret is type $\mathcal{M}^1$, the second is type $\mathcal{M}^2$, etc., and the last middle caret is type $\mathcal{M}^p$. If the parent caret is of type $\mathcal{R}$, then the middle child carets are numbered from left to right, but the numbering begins with $p$ and then goes to 1 so that the first middle child caret is type $\mathcal{M}^p$, the second middle child caret is type $\mathcal{M}^1$, the third middle child caret is type $\mathcal{M}^2$, etc., and the last middle child caret is type $\mathcal{M}^{p-1}$. If the parent caret is of type $\mathcal{M}^i$ for $i = 1, ..., p$, then all the child carets are of type $\mathcal{M}$; the leftmost and rightmost caret children will be the same types as the parent ($\mathcal{M}^i$) and the index on $\mathcal{M}$ will increase from left to right until it reaches $p$, after which the index will start over at 1 and continue increasing as we move to the right.

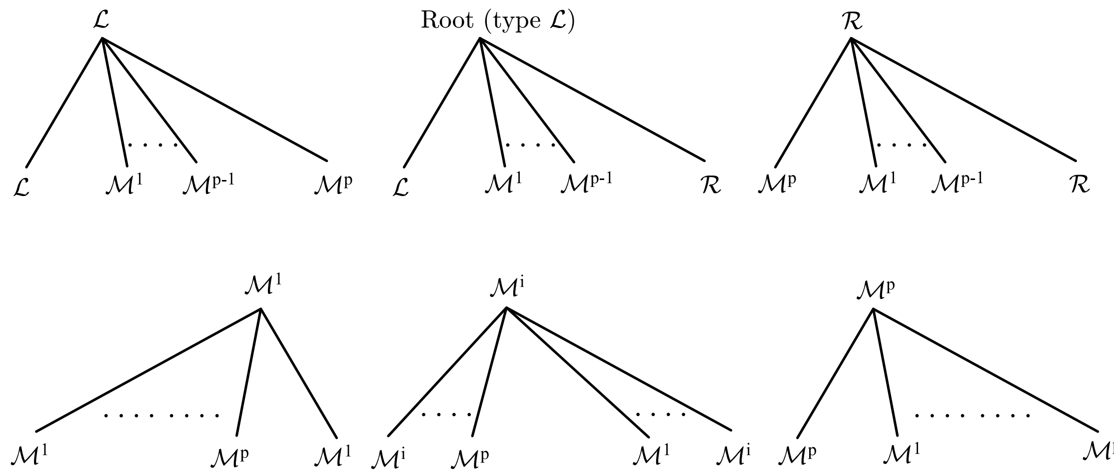


Figure 1.2: For each of the 6 parent caret types given above, the caret type listed below each child is the type of the child caret, if one exists.

### 1.2.5 Fordham's node order

The nodes of each tree are ordered in the following way. We will depict each caret in the form of one of the carets shown in Figure 1.2.

If we were to draw a vertical line through the parent node of a caret, the children drawn to the left of this line will be referred to as the *left children* of the parent, and all children drawn to the right of this line will be referred to as the *right children* of the parent. The caret type will determine which children of a caret are drawn to the left of the parent and which are drawn to the right. When numbering the nodes on the tree, we will number the left child nodes first from left to right, then number the parent, then number the right child nodes from left to right (see Figure 1.3). Because all carets in a $(p+1)$-ary tree are connected to each other by at least one node, the ordering of the nodes of a single caret induces an ordering of all the nodes in a tree.

The numbering of the nodes induces a numbering of the carets if we let a caret's number be the same as the number given to its parent node. We will denote the $n^{th}$-caret by $\wedge_n$. A caret $\wedge_i$ is a *successor* of the caret $\wedge_j$ if and only if $i > j$. For example, to see an element of $F(p+1)$ with all its nodes ordered, see Figure 1.7.

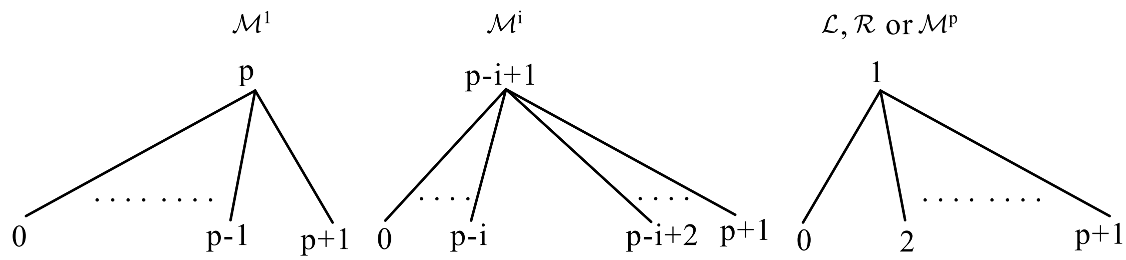


Figure 1.3: The ordering of child nodes with respect to the parent node in different caret types in tree-pair diagrams of elements of $F(p+1)$

### 1.2.6 Fordham's caret types:

Now we need to further subclassify the caret types in a $(p+1)$-ary tree. These more specific caret types are subtypes of the general $\mathcal{L}$, $\mathcal{R}$, and $\mathcal{M}^i$ types which we have already defined. This further subcategorization is necessary in order to proceed with Fordham's method for calculating word length.

1. $\mathcal{L}_\emptyset$. This is the first caret on the left side of the tree, which will therefore have no predecessor and always have index number 0. Every non-empty tree has one and only one caret of this type.

2. $\mathcal{L}_\mathrm{L}$. This is any left caret except the single $\mathcal{L}_\emptyset$ caret.

3. $\mathcal{R}_\emptyset$. This is any right caret with all successor carets of type $\mathcal{R}$.

4. $\mathcal{R}_\mathrm{R}$. This is a right caret whose immediate successor is of type $\mathcal{R}$, but which has at least one successor which is not of type $\mathcal{R}$.

5. $\mathcal{R}_j$. This is a right caret whose immediate successor is not a right caret and whose leftmost child successor is of type $\mathcal{M}^j$, where $j < p$. If the leftmost child successor is of type $\mathcal{R}$, we let $j = p$.

6. $\mathcal{M}^i_\emptyset$. This is a middle caret of type $\mathcal{M}^i$ that has no child successor carets.

7. $\mathcal{M}^i_j$. This is a middle caret of type $\mathcal{M}^i$ with leftmost child successor of type $\mathcal{M}^j$. (Note that $j \leq i$.)

### 1.2.7 Simplifying tree-pair diagrams

Before we can talk of multiplying tree-pair diagrams, we need to point out when tree-pair diagrams can be simplified. We say that a caret is *exposed* if it has no child carets (i.e. all its children are leaves). If there exist exposed carets with exactly the same number (and therefore whose leaves all have the same index numbers) on both the positive and negative trees then these two carets can be removed from their respective trees without changing the element that the tree-pair diagram represents. This removal of unnecessary carets is analogous to simplifying a word. This is possible because tree-pair diagrams can be rewritten as homeomorphisms of the closed unit interval: if there are carets in both the positive and negative trees with exactly the same leaf numbering, then this is akin to sending $p+1$ equal subdivisions of that subinterval represented by the parent node of the caret in the negative tree to exactly the same $p+1$ equal subdivisions of that subinterval represented by the parent node of the caret in the positive tree, all under the identity map; removing this pair of carets would then be equivalent to mapping the whole subinterval represented by the resulting leaf of the caret removal in the negative tree to the whole subinterval represented by the resulting leaf of the caret removal in the positive tree, under the identity map.

### 1.2.8 Multiplying tree-pair diagrams

For this paper all multiplication is on the right, where the multiplication convention is that of function composition. Multiplying x by y on the right will be denoted $xy$, which actually denotes $x \circ y$. To see what this looks like for the tree-pair diagram representations, let $x$ and $y$ be elements of $F(p+1)$ which are represented by the tree-pair diagrams $x = (T_-, T_+)$ and $y = (S_-, S_+)$; then $xy$ would be performed as multiplication on the tree-pair diagrams by doing the following:

We want to match up $S_+$ with $T_-$ so that they are the same tree. This is possible if we notice that we can add any carets we want to the leaves of the tree $S_+$ as long as we add carets at the leaves with the same numbers on the tree $S_-$. This is allowed because it is just the reverse process of that of simplification of the tree-pair diagrams, analogous to subdividing the same subinterval in a homeomorphism of the closed unit interval in exactly the same way in the domain and the range. Likewise, we can add any carets we want to add to the leaves of the tree $T_-$ as long as we add carets at the leaves with the same numbers on the tree $T_+$. By repeatedly adding carets to $S_+$ and $T_-$ (and therefore by extension to $S_-$ and $T_+$), we can eventually turn them into identical trees. If we let $S'_-$ denote $S_-$ with any extra carets that have been added to make up for any carets added to $S_+$ in the process of making it identical to $T_-$, and if we let $T'_+$ denote $T_+$ with any extra carets that have been added to make up for any carets added to $T_-$ in the process of making it identical to $S_+$, then the new tree-pair diagram for $xy$ will be $(S'_-, T'_+)$. We also may often denote the new tree-pair diagram for $xy$ by $((Ty)_-, (Ty)_+)$. To see an example of tree-pair multiplication, see Figure 1.4. We note also that when multiplying an element of $F(p+1)$ by another element on the right, the original element may be changed in the positive and/or the negative tree; however, the positive tree will only be changed if it is necessary to add carets to the tree-pair diagram in order to compute the product. So:

**Remark 1.2.1.** *Under multiplication on the right using function notation, if no carets need be added to an element to compute its product with another element, then the positive tree will remain unchanged under the multiplication, and all changes cause by the multiplication will affect only the negative tree.*

This fact will be used repeatedly in proofs of several of the theorems presented later in this paper.

## 1.3 Presentations of $F(p+1)$

Thompson's group $F(p+1)$ has the following infinite presentation:

$$F(p+1) = \{x_0, x_1, ...|x_j x_i = x_i x_{j+p} \text{ for } i < j\}$$

where the generators can be depicted by the tree-pair diagrams given in Figure 1.5.

The group $F(p+1)$ also has a finite presentation which will be needed to calculate length:
The generators are $\{x_0, x_1, ..., x_p\}$ and the relators are:

$$[x_0 x_i^{-1}, x_j] \text{ when } i < j, [x_0^2 x_i^{-1} x_0^{-1}, x_j] \text{ when } i \geq j-1, \text{ and } [x_0^3 x_p^{-1} x_0^{-2}, x_1]$$

where $j = 0, ..., p$ and $i = 1, ..., p$ (the generators for this finite presentation can be depicted by the tree-pair diagrams given in Figure 1.6). This finite presentation can be obtained from the infinite presentation by induction.

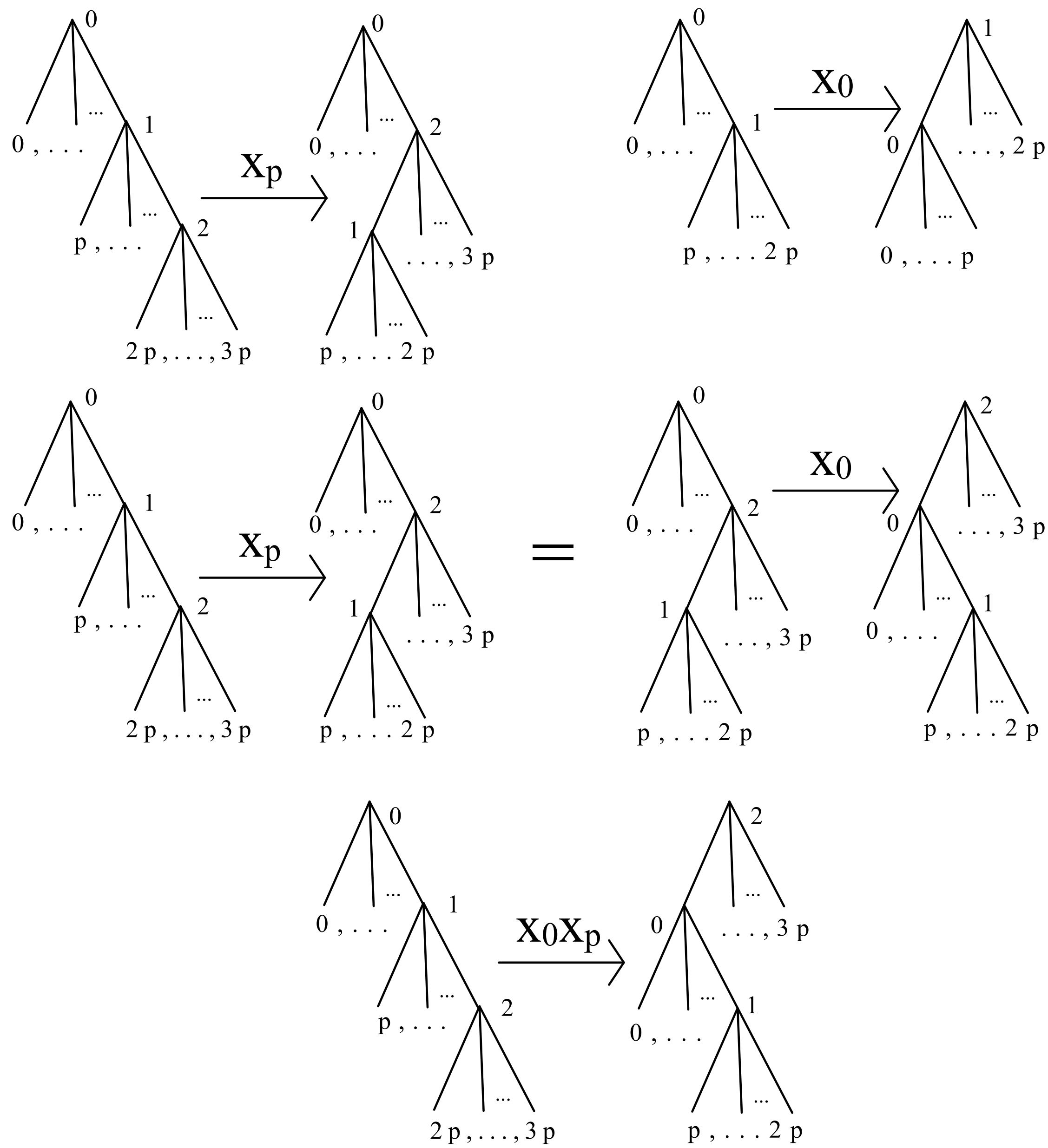


Figure 1.4: Multiplication of tree-pair diagrams for the product $x_0 x_p$ in $F(p+1)$ (to complete multiplication, one caret must be added to the leaf numbered $p$ in the tree-pair diagram for $x_0$)

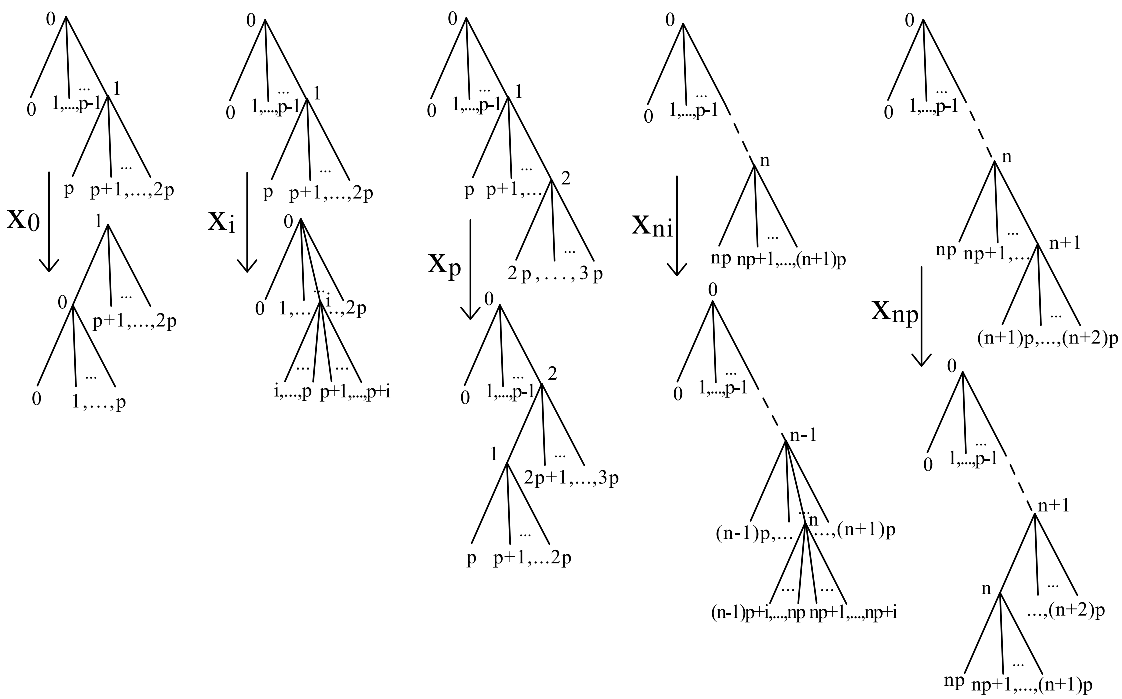

Figure 1.5: The generators $\{x_0, x_1, x_2, ...\}$ for the standard infinite presentation of $F(p+1)$ ($i = 1, 2, ...p-1$ and $n \in \mathbb{N}$)

## 1.4 Normal Forms of Elements of $F(p+1)$

By looking at the relators for the infinite presentation for $F(p+1)$, it becomes clear that all elements of $F(p+1)$ can be put into the form

$$x_{i_1}^{r_1} x_{i_2}^{r_2} \ldots\ x_{i_n}^{r_n} x_{j_m}^{-s_m} \ldots\ x_{j_2}^{-s_2} x_{j_1}^{-s_1}$$

where the generators are taken from the infinite presentation such that

$$i_1 < i_2 < \cdots < i_n \neq\ j_m > \cdots > j_2 > j_1$$

To ensure uniqueness and guarantee that this is the normal form of an element of $F(p+1)$ we need only add the condition that if both $x_i$ and $x_i^{-1}$ appear in the above expression, then a generator $x_j$ (or its inverse) where $i < j < i+p+1$, must also appear in the above expression; this normal form was first proved in [3].

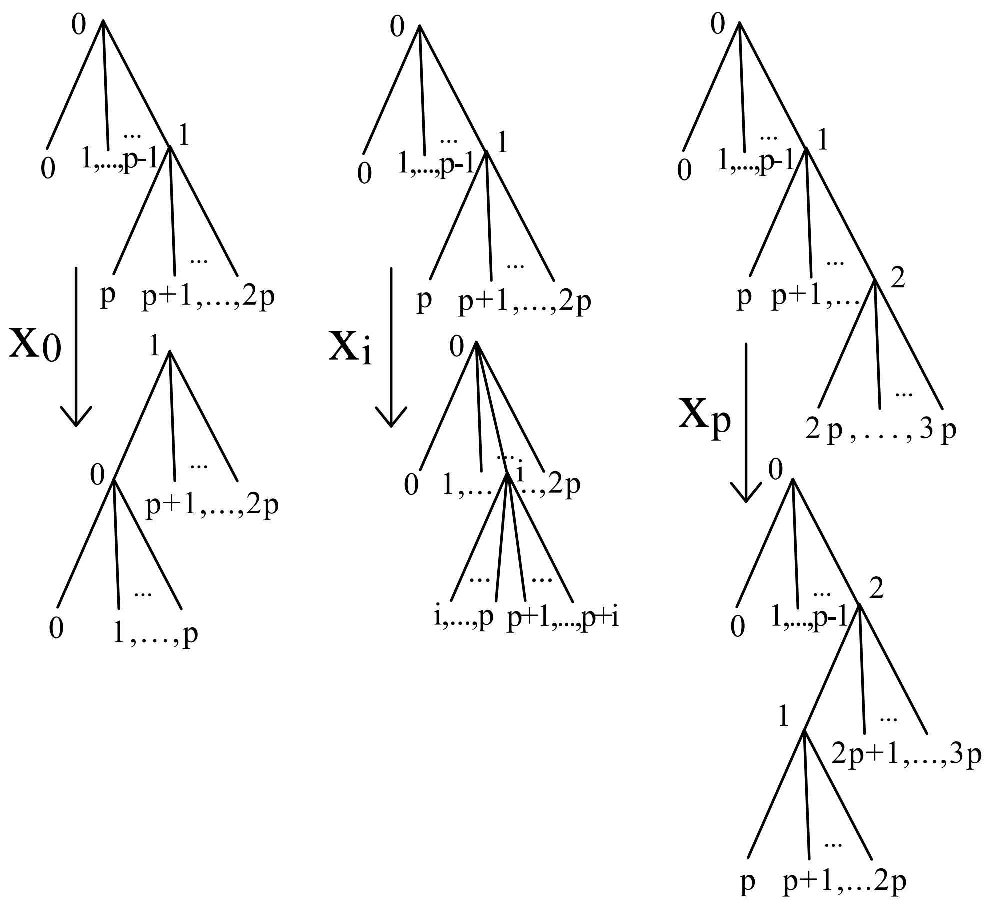


Figure 1.6: The generators $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$ for the standard finite presentation of $F(p+1)$ $(i = 1, 2, ...p-1)$

We can write down the normal form of a reduced tree-pair diagram in the following way:

1. To find all generators with positive exponents in the normal form, we look at the positive tree, and to find all generators with negative exponents in the normal form, we look at the negative tree.

2. To find out which generators are present in the positive or negative tree we are looking at, we will need to consider the leaf ordering of each tree. We determine the exponent of $x_i$ for $i = 0, 1, 2, ...$ in the following way. In order for the exponent of $x_i$ to be nonzero, the *ith* leaf in the tree must be at the end of a left edge of a caret. Then the nonzero exponent is determined by counting the number of left edges (edges which are the left edge of some caret in the tree) that appear in a row on the directed path from the *ith* leaf to the root node (any left edges that appear on this path after the appearance of a non-left edge will be excluded from the total); we also exclude from this total any left edges which appear in the root caret or a right caret. (If we are looking at the positive tree, this number would yield the positive exponent of $x_i$ in the normal form, and if we are looking at the negative tree, this number would yield the negative exponent of $x_i$ in the normal form). For example, the element $x_1 x_3^5 x_4^{-1} x_0^{-3}$ in $F(p+1)$ can be depicted by the tree-pair diagram given in Figure 1.7.

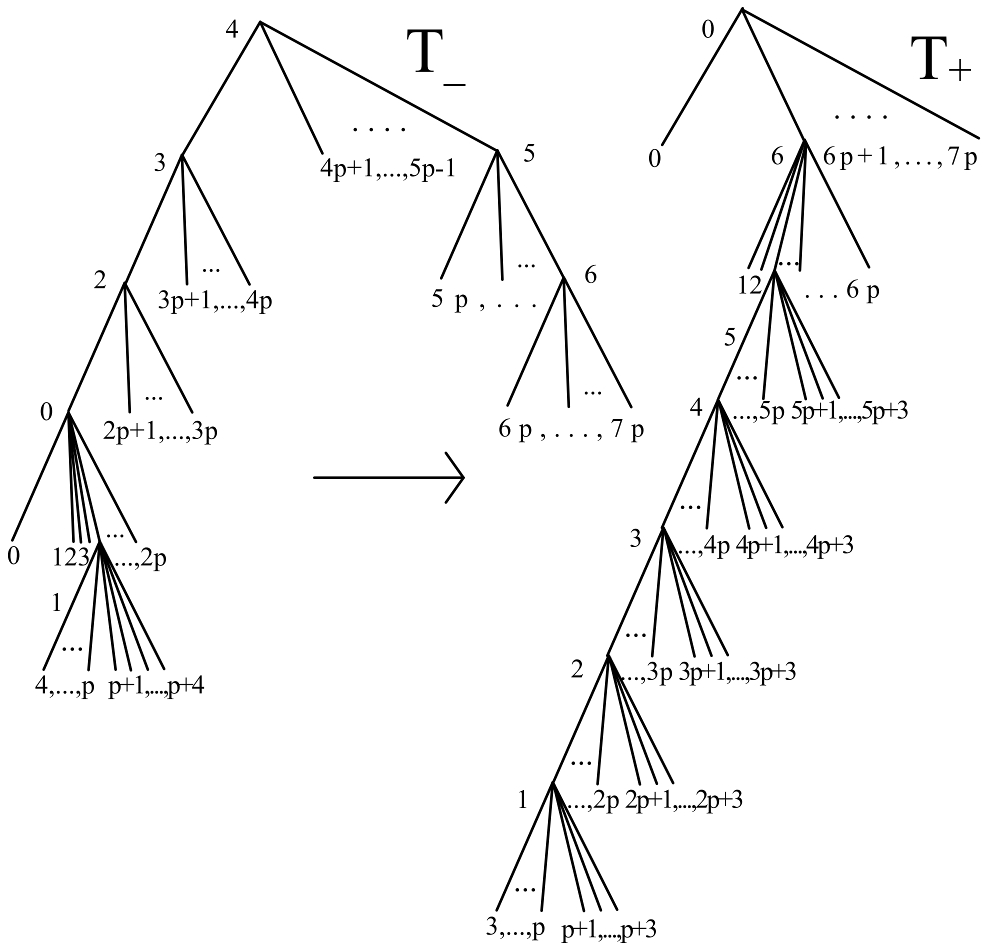


Figure 1.7: $x_1 x_3^5 x_4^{-1} x_0^{-3}$ in $F(p+1)$ with all carets and leaves numbered

# Chapter 2

# The Metric on $F(p+1)$

## 2.1 Fordham's method for computing word length in $F(p+1)$.

We use Fordham's method to compute the length of words in $F(p+1)$, which was introduced in [12] (see also [13] and [14] for Fordham's method for computing length in $F(2)$). Fordham's method for computing word length in $F(p+1)$ is based on the representation of the group by $(p+1)$-ary tree-pair diagrams.

To determine word length of elements in $F(p+1)$, Fordham creates a table that assigns a weight to each possible caret pairing that could be obtained in a $(p+1)$-ary tree-pair diagram. This table is displayed in Table 2.1. The weight which Fordham assigns to a caret pair comes from the cardinality of the set of generators which must be required to create such a caret pair. We will use the notation $w(\wedge_i)$ to denote the weight, given by Fordham's table, of the pair of carets in the tree-pair diagram numbered $i$. We note that carets of type $\mathcal{L}_\emptyset$ are not listed on the table; since there is only one caret of this type in both the positive and negative trees, the only pairing possible is $(\mathcal{L}_\emptyset,\mathcal{L}_\emptyset)$ and $w(\mathcal{L}_\emptyset,\mathcal{L}_\emptyset)=0$.

Fordham then proves the following theorem:

**Theorem 2.1.1.** *(Fordham [12], Theorem 2.0.11). Given an element $w$ in $F(p+1)$ described by the reduced tree-pair diagram $(T_-,T_+)$, the word length $|w|$ of the element with respect to the generating set $\{x_0,x_1,\dots,x_{p-1},x_p\}$ is the sum of the weights of each of the pairs of carets in the tree-pair diagram.*

Also in [12], Fordham proves the following theorem:

Table 2.1: Weight of Types of Caret Pairs in the Tree-Pair Diagram of Elements in $F(p+1)$ $(j_1 \leq i < j_2, i_1 < j \leq i_2)$

| ( , ) | $\mathcal{L}$ | $\mathcal{R}_\emptyset$ | $\mathcal{R}_{\mathrm{R}}$ | $\mathcal{R}_j$ | $\mathcal{M}_\emptyset^i$ | $\mathcal{M}_j^i$ |
|---|---|---|---|---|---|---|
| $\mathcal{L}$ | 2 | 1 | 1 | 1 | 2 | 2 |
| $\mathcal{R}_\emptyset$ | 1 | 0 | 2 | 2 | 1 | 3 |
| $\mathcal{R}_{\mathrm{R}}$ | 1 | 2 | 2 | 2 | 1 | 3 |
| $\mathcal{R}_{j_1}$ | 1 | 2 | 2 | 2 | 3 | 3 |
| $\mathcal{M}_\emptyset^{i_1}$ | 2 | 1 | 1 | 1 | 2 | 2 |
| $\mathcal{M}_{j_1}^{i_1}$ | 2 | 3 | 3 | 3 | 4 | 4 |
| $\mathcal{R}_{j_2}$ | 1 | 2 | 2 | 2 | 1 | 3 |
| $\mathcal{M}_\emptyset^{i_2}$ | 2 | 1 | 1 | 3 | 2 | 4 |
| $\mathcal{M}_{j_2}^{i_2}$ | 2 | 3 | 3 | 3 | 2 | 4 |

**Theorem 2.1.2.** *(Fordham [12], Theorem 2.1.1). If $x$ is a generator of $F(p+1)$ that can be applied to the reduced $n$-caret tree-pair diagram $w = (T_-, T_+) \in F(p+1)$ without adding any carets, and if $((Tx)_-, (Tx)_+)$ is also a reduced $n$-caret tree-pair diagram, then there is exactly one caret $\wedge_i$ (where $i < n$) that changes type; that is, if we let $\tau_{(T_-,T_+)}(\wedge)$ denote the caret type of $\wedge$ in the tree-pair diagram $(T_-, T_+)$, then $\exists\ i < n$ such that*

$$\tau_{(T_-,T_+)}(\wedge_i) \neq \tau_{((Tx)_-,(Tx)_+)}(\wedge_i) \text{ and } \tau_{(T_-,T_+)}(\wedge_j) = \tau_{((Tx)_-,(Tx)_+)}(\wedge_j) \forall\ j \neq i$$

When the conditions of Theorem 2.1.2 fail, Fordham provides two alternate theorems:

**Theorem 2.1.3.** *(Fordham [12], Theorem 2.1.3). If $x$ is a generator of $F(p+1)$ and $w \in F(p+1)$ is represented by the tree-pair diagram $(T_-, T_+)$, and $x$ cannot be applied to $T_-$ without first adding a caret, then $|wx| > |w|$.*

**Theorem 2.1.4.** *(Fordham [12], Theorem 2.1.4). If $x$ is a generator of $F(p+1)$ and $w \in F(p+1)$ is represented by the reduced tree-pair diagram $(T_-, T_+)$, and $((Tx)_-, (Tx)_+)$ is not reduced, then $|wx| = |w| - 1$.*

**Remark 2.1.5.** *We note that Theorems 2.1.2, 2.1.3, and 2.1.4 are all mutually exclusive. Only one of the three can apply in any given case.*

# Chapter 3

# Seesaw words

## 3.1 Definitions and Background

**Definition (seesaw word) 3.1.1.** *A word $w$ with length $|w|$ is a* **seesaw word with swing $k$ with respect to the generator** *$g$ in the generating set $X$ if the following conditions hold:*

1. $|wg^l| = |w| - l$ *for* $0 < |l| \leq k$

2. $|wg^l h| \geq |wg^l|$ *for all $h$ such that $h \neq g$ and when* $0 < |l| < k$

*In other words, for a seesaw word $w$, all geodesic representatives of $w$ end in either $g^k$ or $g^{-k}$ and there is at least one geodesic representative of each type.*

**Definition (k-fellow traveller property) 3.1.2.** *Let $\lambda$ and $\eta$ be geodesic paths in the Cayley graph $\Gamma(G, X)$ which begin at the identity and terminate at $w$ and $v$, respectively. Then $\lambda$ and $\eta$* **(synchronously) k-fellow travel** *if for some constant $k$:*

1. $d_\Gamma(w, v) = 1$ *and*

2. *For any 2 vertices $h$ and $g$ on $\lambda$ and $\eta$ respectively, if $|h| = |g|$, then $d_\Gamma(h, g) \leq k$.*

The existence of seesaw words with arbitrarily large swing in a group will show that no collection of geodesic paths in that group's Cayley graph can satisfy the k-fellow traveler property.

We say that a group is *(synchronously) combable* if it can be represented by a language of words satisfying the (synchronous) k-fellow traveller property.

## 3.2 Seesaw words exist in $F(p+1)$

Cleary and Taback [11] showed that seesaw words exist with arbitrarily large swing in $F(2)$. We now extend that result to $F(p+1)$ for all $p \in \mathbb{N}$.

**Theorem 3.2.1.** *Thompson's group $F(p+1)$ contains seesaw words of arbitrarily large swing with respect to the generator $x_0$ in the standard generating set $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$.*

*Proof.* To show the existence of seesaw words in $F(p+1)$, we need to construct $(p+1)$-ary tree-pair diagrams by choosing the pairs of caret types they contain so that the change in caret type induced by each successive multiplication of the prospective seesaw word $w$ by $x_0$ or $x_0^{-1}$ reduces word length by one each time. Likewise, we want to choose the pairs of caret types in the tree-pair diagram so that multiplying $wx_0^{\pm n}$ (where $n = 0, \ldots, k$ for some fixed $k$) by any generator $x_i^{\pm 1}$ (where $i \neq 0$) increases length by 1.

In this way, we can define one particular family of seesaw words in which each word in the family has the following normal form:

$$x_0^{m+l(p-1)-1} \cdot x_p \cdots x_{2p} \cdot x_{(l+r)p^2+(m+2n-l-r+2)p} \cdots x_{(l+r)p^2+(m+2n-l-r+3)p} \cdot$$
$$\cdot x^{-1}_{(l+r)p^2+(m+r-l+j+1)p} \cdots x^{-1}_{(l+r)p^2+(m+r-l+j)p} \cdots$$
$$\cdot x^{-1}_{(l+r)p^2+(m+r-l)p} \cdots x^{-1}_{lp^2+(m-l)p+1} \cdot$$
$$\cdot x^{-1}_{kp^2+(m-l)p-1} \cdots x^{-1}_{(k-1)p^2+(m-l)p+1} \cdots x_0^{-m}$$

where $j = 2n - 2r + 1, \ldots, 1$ and $k = l, \ldots, 1$

The tree-pair diagram of an element of this form can be seen in Figure 3.1.

We will denote the family of all words of this form by $\mathcal{S}$. For a given $w \in \mathcal{S}$ we will let $(S_-, S_+)$ denote the tree-pair diagram of $w$. The above word depends on four parameters: $l$, $r$, $m$, and $n$. The letter $l$ denotes the number of left carets, counting consecutively from the root but excluding the root, that have a child caret of type $\mathcal{M}_\emptyset^i$ for all $i = 1, \ldots, p-1$. The letter $r$ denotes the number of $\mathcal{R}_j$ carets, counting consecutively from the root, that have a child caret of type $\mathcal{M}_\emptyset^i$ for all $i = 1, \ldots, p-1$. The letter $m$ denotes the length of the left side of $S_-$, i.e. the number of carets of type $\mathcal{L}_\mathrm{L}$ in $S_-$ (and the number of carets of type $\mathcal{L}$ in $S_+$). The letter $n$ denotes the number of carets of type $\mathcal{R}$ along the right side of $S_-$ which are not of type $\mathcal{R}_\emptyset$ ($S_+$ will have $2n + (r+1)(p-1)$ carets of type $\mathcal{R}_\mathrm{R}$). We require $r \leq n$ and $l \leq m$.

Now we will consider an arbitrary $w \in \mathcal{S}$, and by repeatedly applying Theorem 2.1.2, analyze which single caret type changes and whether this change in caret type might increase or decrease the weight of the caret, and thus the length of the word. If at any point in this iteration of multiplication the conditions of Theorem 2.1.2 are not satisfied, then by Theorem 2.1.3, the length of the word is increased by the multiplication.

To see what happens with repeated multiplication by $x_0^{-1}$, we consider the tree pair diagram $(S_-^q, S_+^q)$ for $wx_0^{-q}$ where $0 \leq q < r$ (see Figure 3.2). When we multiply $w$ by a single $x_0^{-1}$, $\wedge_{m+(l+1)(p-1)+2}$ (the right child of the root) in

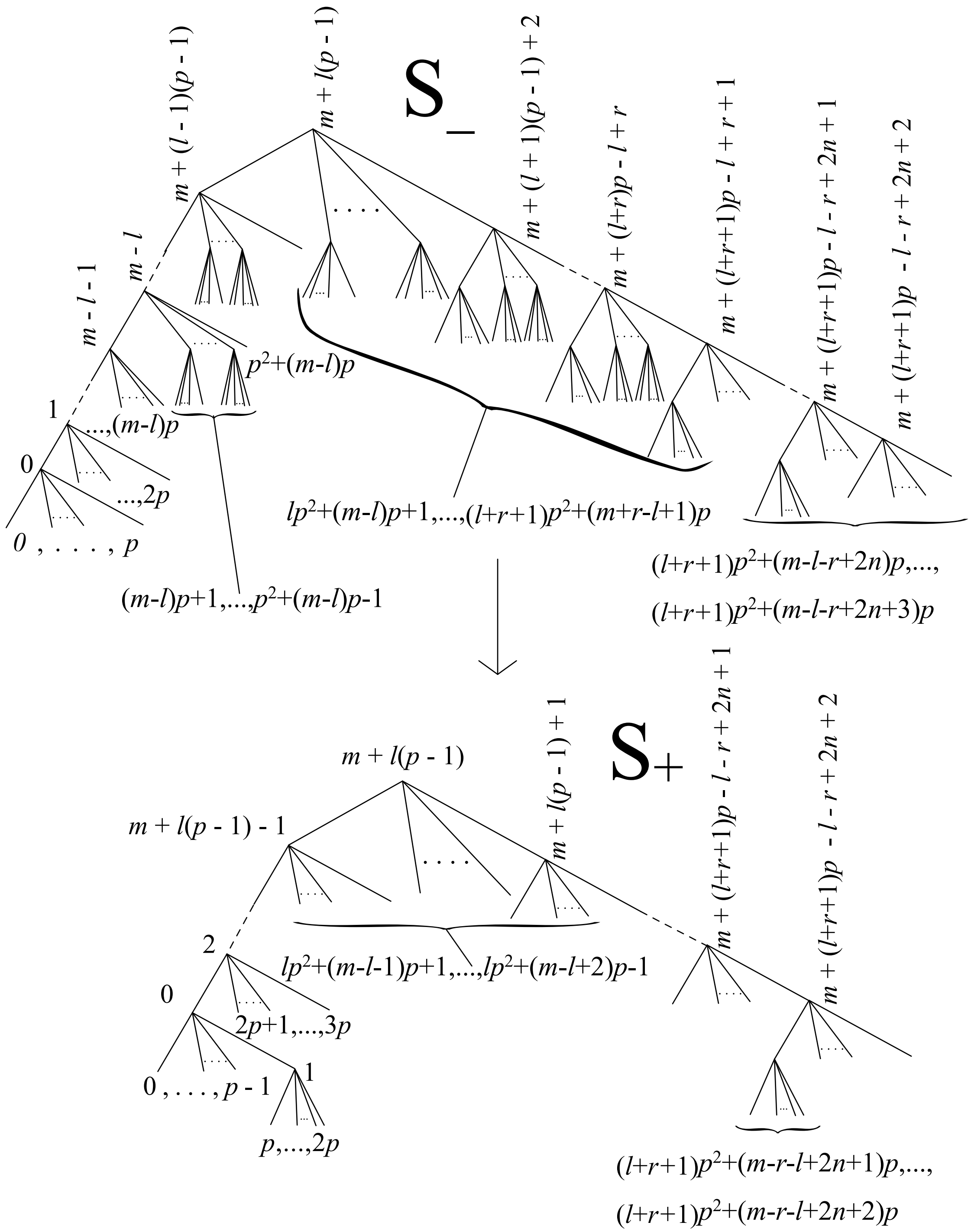


Figure 3.1: Seesaw elements in $F(p+1)$

$S_-^0$ $(i.e. S_-)$ changes from type $\mathcal{R}_j$ to type $\mathcal{L}_{\mathrm{L}}$ through a kind of counterclockwise rotation of the negative tree through its root. Through repeated iterations of multiplication by $x_0^{-1}$, we can see that multiplying $wx_0^{-(q-1)}$ by $x_0^{-1}$ changes $\wedge_{m+(l+q)(p-1)+2q}$ in $S_-^q$ (the right child of the root) from type $\mathcal{R}_j$ to type $\mathcal{L}_{\mathrm{L}}$. By looking at the tree-pair diagram we see that the caret $\wedge_{m+(l+i)(p-1)+2i}$ in $S_+^q$ will always be of type $\mathcal{R}_{\mathrm{R}}$ for any given $i = 0, 1, \ldots, q$ because the first successor of the root in $S_+^q$ which is not of type $\mathcal{R}_{\mathrm{R}}$ is $\wedge_{m+2n+p+(l+r+1)(p-1)}$. Because $q < r \leq n$, $m + (l + i)(p - 1) + 2i < m + 2n + p + (l + r + 1)(p - 1)$ for all $i = 0, 1, ..., q$. Therefore, this change in the caret $\wedge_{m+(l+q)(p-1)+2q}$ in the negative tree from type $\mathcal{R}_j$ to type $\mathcal{L}_{\mathrm{L}}$ changes the pairing from ($\mathcal{R}_j$,$\mathcal{R}_{\mathrm{R}}$), which has weight 2, to ($\mathcal{L}_{\mathrm{L}}$,$\mathcal{R}_{\mathrm{R}}$), which has weight 1. So, by induction, we can see that $|wx_0^{-q}| = |w| - q$ for all $q$ such that $0 \leq q < r$.

Now we consider iterated multiplications of $w$ by $x_0$. The argument is similar to that for $x_0$. We consider the tree-pair diagram $(R_-^q, R_+^q)$ for $wx_0^q$ where $0 \leq q < l$ (see Figure 3.3). When we multiply $w$ by a single $x_0$, $\wedge_{m+l(p-1)}$ (the root caret) in $R_-^0$ (i.e. $S_-$) changes from type $\mathcal{L}_{\mathrm{L}}$ to type $\mathcal{R}_j$ through a kind of clockwise rotation of the negative tree through its root. Through several iterations of multiplication by $x_0$, we can see that multiplying $wx_0^{q-1}$ by $x_0$ changes $\wedge_{m+(l-q+1)(p-1)-(q-1)}$ (the root caret) in $R_-^q$ from type $\mathcal{L}_{\mathrm{L}}$ to type $\mathcal{R}_j$. By looking at the tree-pair diagram we see that the caret $\wedge_{m+(l-i+1)(p-1)-(i-1)}$ on $R_+^q$ will always be of type $\mathcal{L}_{\mathrm{L}}$ for any given $i = 0, 1, \ldots, q$ because the only predecessors of the root in $R_+^q$ which are not of type $\mathcal{L}_{\mathrm{L}}$ are $\wedge_1$ and $\wedge_0$. Because $q < l \leq m$ and $p > 0$, $m + (l - i + 1)(p - 1) - (i - 1) > 1$ for all $i = 0, 1, ..., q$. Therefore, this change in the negative caret $\wedge_{m+(l-q+1)(p-1)-(q-1)}$ from type $\mathcal{L}_{\mathrm{L}}$ to type $\mathcal{R}_j$ changes the pairing from ($\mathcal{L}_{\mathrm{L}}$,$\mathcal{L}_{\mathrm{L}}$), which has weight 2, to ($\mathcal{R}_j$,$\mathcal{L}_{\mathrm{L}}$), which has weight 1. So, by

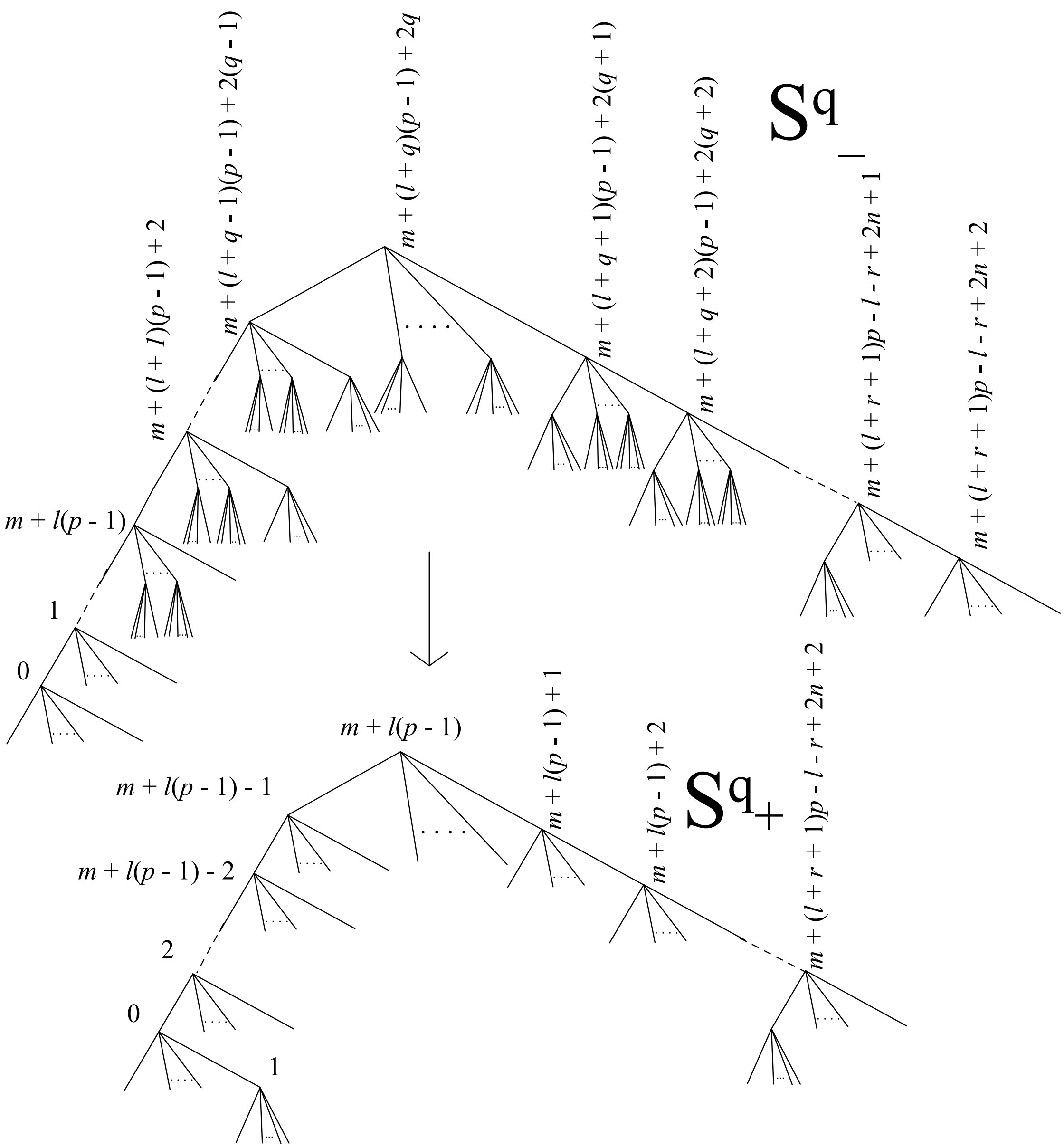


Figure 3.2: $wx_0^{-q}$ (when $0 \leq q < r$) in $F(p+1)$

induction, we can see that $|wx_0^q| = |w| - q$ for all $q$ such that $0 \leq q < l$.

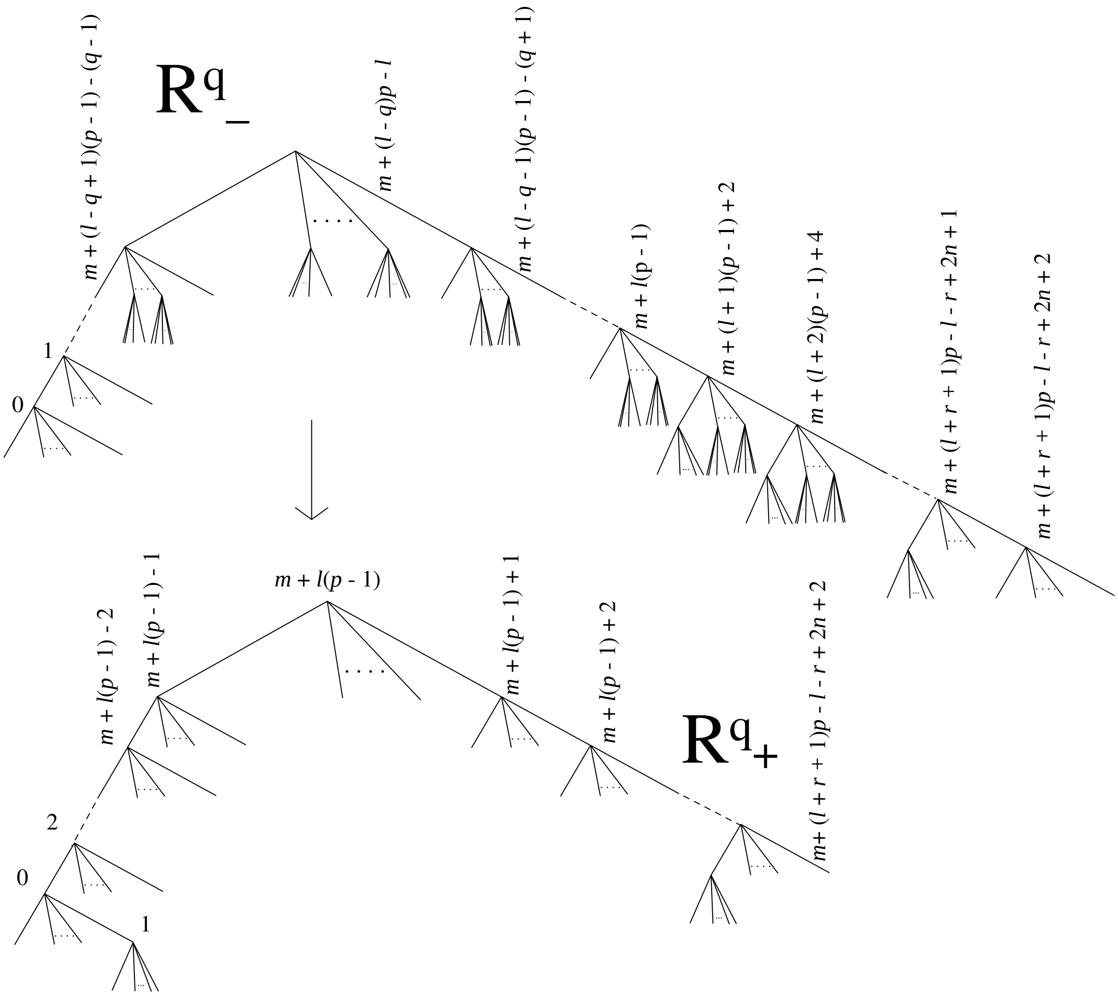


Figure 3.3: $wx_0^q$ (when $0 \leq q < l$) in $F(p+1)$

Now we consider the action of the generators $x_i^{\pm 1}$ for $i = 0, 1, \ldots, p$ on $wx_0^{\pm q}$.

Clearly multiplying $wx_0^{-q}$ by $x_0$ when $0 < q < r$ increases the length because $wx_0^{-q}x_0 = wx_0^{-(q-1)}$ and $|wx_0^{-(q-1)}| = |wx_0^{-q} - 1|$.

Likewise, multiplying $wx_0^q$ by $x_0^{-1}$ when $0 < q < l$ increases the length because $wx_0^q x_0^{-1} = wx_0^{q-1}$ and $|wx_0^{q-1}| = |wx_0^q - 1|$.

If we let $i = 1, \ldots, p$, multiplying $wx_0^{-q}$ by $x_i^{-1}$ when $0 < q < r$ changes $\wedge_{m+(l+q+1)(p-1)+2(q+1)}$ (the right child of the root) in $S_-^q$ from type $\mathcal{R}$ to type $\mathcal{M}_1^i$, changing the pairing from ($\mathcal{R}$,$\mathcal{R}_j$), which has weight 2, to ($\mathcal{M}_1^i$,$\mathcal{R}_j$), which has weight 3. So $|wx_0^{-q}x_i| = |w| - q + 1$ for $0 < q < r$ and $i = 1, \ldots, p$.

If we let $i = 1, \ldots, p$, multiplying $wx_0^q$ by $x_i^{-1}$ when $0 < q < l$ changes $\wedge_{m+(l-q-1)(p-1)-(q+1)}$ (the right child of the root) in $R_-^q$ from type $\mathcal{R}$ to type $\mathcal{M}_1^i$, changing the pairing from ($\mathcal{R}$,$\mathcal{R}_j$), which has weight 2, to ($\mathcal{M}_1^i$,$\mathcal{R}_j$), which has weight 3. So $|wx_0^q x_i| = |w| - q + 1$ for $0 < q < l$ and $i = 1, \ldots, p$.

If we let $i = 1, \ldots, p$, multiplying $wx_0^{-q}$ by $x_i$ when $0 < q < r$ changes $\wedge_{m+(l+q)(p-1)+2q+i}$ (the *ith* child of the root) in $S_-^q$ from type $\mathcal{M}_\emptyset^i$ to type $\mathcal{R}_{i+1}$, changing the pairing from ($\mathcal{M}_\emptyset^i$,$\mathcal{R}_{\mathrm{R}}$), which has weight 1, to ($\mathcal{R}_{i+1}$,$\mathcal{R}_{\mathrm{R}}$), which has weight 2. So $|wx_0^{-q}x_i| = |w| - q + 1$ for $0 < q < r$ and $i = 1, \ldots, p$.

If we let $i = 1, \ldots, p-1$, multiplying $wx_0^{-q}$ by $x_i^{-1}$ when $0 < q < l$ changes $\wedge_{m+(l-q)(p-1)-q+i}$ (the *ith* child of the root) in $R_-^q$ from type $\mathcal{M}_\emptyset^i$ to type $\mathcal{R}$, changing the pairing from ($\mathcal{M}_\emptyset^i$,$\mathcal{R}_{\mathrm{R}}$), which has weight 1, to ($\mathcal{R}$,$\mathcal{R}_{\mathrm{R}}$), which has weight 2. So $|wx_0^{-q}x_i^{-1}| = |w| - q + 1$ for $0 < q < l$ and $i = 1, \ldots, p-1$.

If we let $i = p$, multiplying $wx_0^q$ by $x_i$ when $0 < q < l$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_p$ because we must add a left caret child to the right child of the root in $R_-^q$ before we can complete the multiplication, so we know from Theorem 2.1.3 that $|wx_0^q x_p| > |wx_0^q|$.

Therefore, all $w \in \mathcal{S}$ are seesaw words, and we can create such words with any given swing $k$.

□

The seesaw words created in the proof of Theorem 3.2.1 can be asymmetric, so for simplicity we can choose a family of seesaw words such that $m = n = k+2$ and $r = l = k$. This gives us a family of symmetric seesaw words of swing $k$. The words in this family have the following normal form:

$$x_0^{kp+1} \cdot x_p \cdots x_{2p} \cdot x_{2kp^2+(k+8)p} \cdots x_{2kp^2+(k+9)p} \cdot$$
$$\cdot x^{-1}_{2kp^2+(k+j+3)p} \cdots x^{-1}_{2kp^2+(k+j+2)p} \cdots x^{-1}_{2kp^2+(k+2)p} \cdots x^{-1}_{kp^2+2p+1} \cdot$$
$$\cdot x^{-1}_{tp^2+2p-1} \cdots x^{-1}_{(t-1)p^2+2p+1} \cdots x_0^{-(k+2)}$$

where $j = 5, 4, 3, 2, 1$ and $t = k, \ldots, 1$

The tree-pair diagram of an element of this form can be seen in Figure 3.4.

**Remark 3.2.2.** *In the case when* $p = 1$*, these two families of seesaw words for* $F(p+1)$ *(one asymmetrical, one symmetrical) are exactly the families of seesaw words for* $F(2)$ *defined by Cleary and Taback in [11].*

The existence of these seesaw words shows us that it is not possible to obtain a family of geodesics in the Cayley graph of $F(p+1)$ with the k-fellow traveller property.

**Proposition 3.2.3.** *Given any constant k, there exists a word* $w \in \mathcal{S}$ *such that no geodesic paths from the identity to* $wx_0$*,* $w$*, and* $wx_0^{-1}$ *satisfy the k-fellow traveler property.*

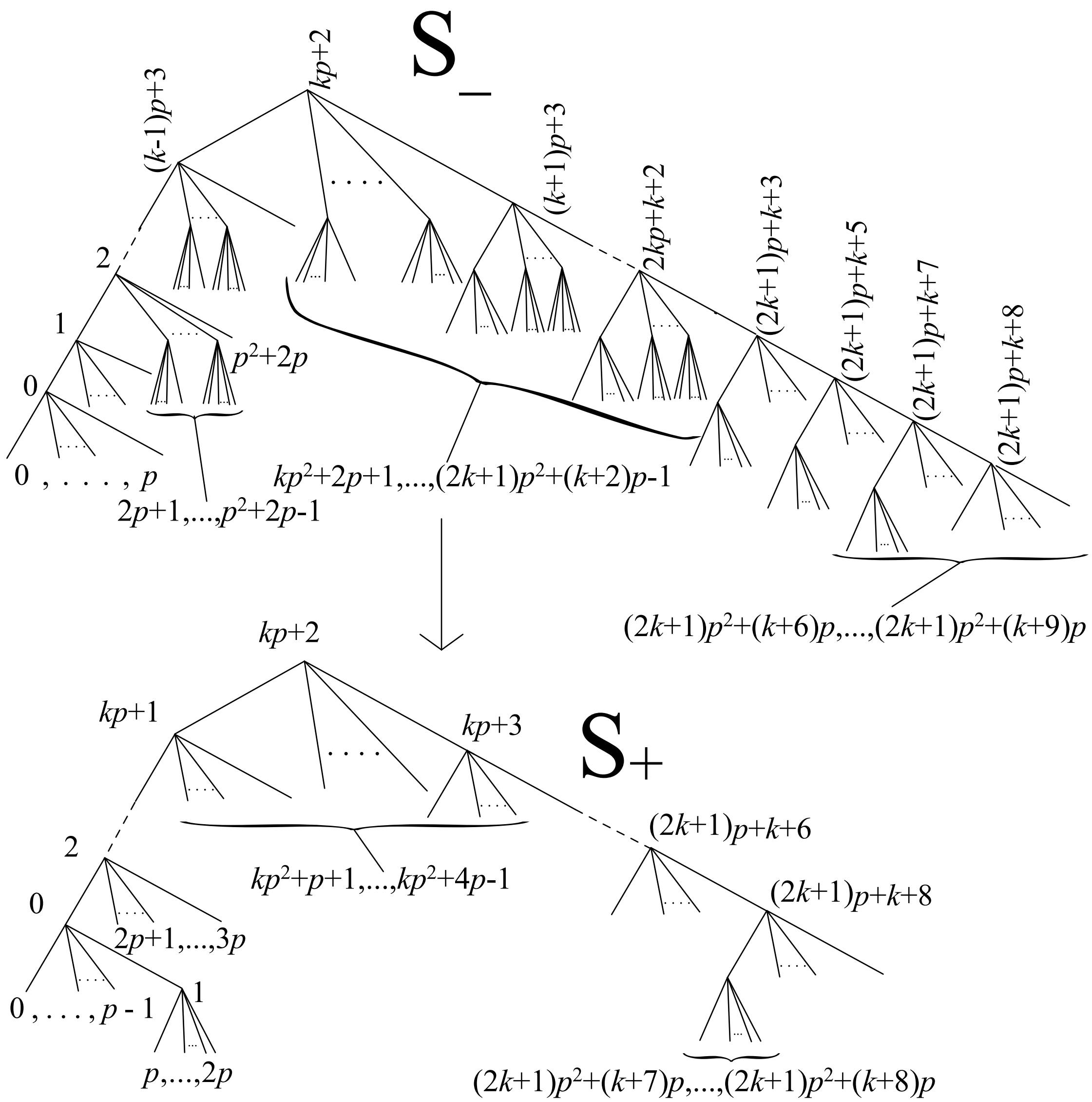


Figure 3.4: Symmetric Seesaw Words in $F(p+1)$

*Proof.* This proof generalizes the structure used by Cleary and Taback in [11] to prove this property in the case when $p = 1$.

If we take an any seesaw word $w \in \mathcal{S}$ with swing n, any geodesic path in the Cayley graph of $F(p+1)$ must pass through either the vertex $wx_0$ or the vertex $wx_0^{-1}$; this geodesic path must then end in either $x_0^{-n}$ or $x_0^n$ respectively.

Let $\gamma$ be a geodesic path from the identity to $w$. Suppose that $\gamma$ passes through the vertex $wx_0$. (A similar argument follows for the case where $\gamma$ passes through the vertex $wx_0^{-1}$.) Let $\eta$ be a geodesic path from the identity to $wx_0^{-1}$. Clearly $d_\Gamma(w, wx_0^{-1}) = 1$. We can rewrite $\gamma = \gamma' x_0^s$ and $\eta = \eta' x_0^{-(s-1)}$ as long as we choose $s \leq n$. So $wx_0^{s-1}$ is on the path $\gamma$ and $wx_0^{-(s-1)}$ is on the path $\eta$. Because of the properties of seesaw words, $d_\Gamma(wx_0^{s-1}, wx_0^{-(s-1)}) = 2(s-1)$.

Because seesaw words exist with arbitrarily large swing, given any constant $k$, we can choose seesaw word $w$ with swing $s > \frac{k}{2} + 1$. Then, using the above construction, there exist vertices $h = wx_0^{s-1}$ and $g = wx_0^{-(s-1)}$, on paths $\lambda$ and $\eta$ respectively, such that $d_\Gamma(h, g) = 2(s-1) > k$. □

**Theorem 3.2.4.** *Thompson's group $F(p+1)$ is not combable by geodesics.*

*Proof.* This proof generalizes the structure used by Cleary and Taback in [11] to prove this property in the case when $p = 1$. Suppose there does exist a combing of $F(p+1)$ by geodesics. If we let $\gamma$ represent the geodesic combing path from the identity to $w$, then we know that the vertex $wx_0^\epsilon$ is on the path $\gamma$ where $\epsilon \in \{-1, 1\}$, but the vertex $wx_0^{-\epsilon}$ is not. Let $\eta$ denote the geodesic combing path from the identity to $wx_0^{-\epsilon}$. We have already shown in Proposition 3.2.3 that $\gamma$

and $\eta$ as described here do not have the k-fellow traveller property, and therefore $F(p+1)$ is not combable by geodesics. □

### 3.3 Regular languages of geodesics in $F(p+1)$

Proposition 3.2.3 also leads to the consequence that no regular language of geodesics exists for $F(p+1)$ with respect to the standard finite generating set. To prove this we will first need a few definitions and theorems:

**Definition (cone type) 3.3.1.** *(Cannon [6]). The cone type of $w \in G$ is the set of all geodesic extensions of $w$ in the Cayley graph. Or, more formally: The cone type of $w$ is: $C(w) = \{\lambda | w\lambda$ is a geodesic$\}$.*

It becomes important to note that there may be paths in a cone type which terminate in the middle of an edge on the Cayley graph unless all the relators for the given presentation of the group are of even length.

**Theorem 3.3.2.** *Thompson's group $F(p+1)$ has infinitely many cone types with respect to the generating set $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$.*

*Proof.* This proof generalizes the structure used by Cleary and Taback in [8] to prove this property in the case when $p = 1$. If we let $w$ represent a seesaw element of swing $s$, then the possible elements of the cone of $wx_0^n$ where $n \leq s$ include $x_0^n$ but not $x_0^{n+1}$. Varying $n$ produces a finite set of elements of different cone types, and because there exist seesaw elements in $F(p+1)$ of arbitrary swing, the value of $s$ is unbounded, and therefore there exist infinitely many distinct cone types in $F(p+1)$ with respect to the standard finite generating set. □

**Lemma 3.3.3.** *(Neumann and Shapiro [16]). If a group is finitely presented and all relators are of even length, and the full language of geodesics is regular with* *respect to the given generating set, then the group has finitely many cone types with respect to the given generating set.*

**Theorem 3.3.4.** *There does not exist a regular language of geodesics for Thompson's group $F(p+1)$ with respect to the generating set $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$.*

*Proof.* This proof generalizes the structure used by Cleary and Taback in [8] to prove this property in the case when $p = 1$. All relators for $F(p+1)$ in the standard finite presentation are in commutator form, so they are all clearly of even length. If we suppose that $F(p+1)$ has a regular language of geodesics with respect to the standard finite generating set, by Lemma 3.3.3 we must conclude that $F(p+1)$ has finitely many cone types. But Theorem 3.3.2 states that $F(p+1)$ has infinitely many cone types, so we must conclude that there does not exist a regular language of geodesics for $F(p+1)$ in the standard finite generating set. □

# Chapter 4

# Dead Ends in Thompson's group $F(p+1)$

## 4.1 Dead Ends and K-pockets

If we consider the Cayley graph $\Gamma$ of $F(p+1)$ in the standard finite generating set, we encounter a certain family of elements with the property that every element in the family is reduced in length when multiplied on the right by any generator or its inverse.

**Definition (dead ends) 4.1.1.** *An element $w$ of a group $G$ is a dead end with respect to the given generating set $X$ if $|wg^{\pm 1}| < |w|$ for all $g \in X$.*

The significance of dead end elements is that no geodesic path in $\Gamma$ which passes through a dead end $w$ can continue past $w$, i.e. the cone type of a dead end element is $\emptyset$. In this section we give a general form for all dead end elements in $F(p+1)$.

**Definition (depth of a dead end element) 4.1.2.** *For a dead end element $w$ in an infinite group $G$, let $|w| = n$. The depth of a dead end element $w$ in the generating set $X$ is the number $m$ such that $|wg_1g_2 \cdots g_{m+1}| > n$ for some $g_1, g_2, ..., g_{m+1} \in X$ but $|wh_1h_2 \cdots h_m| \leq n$ for all possible selections $h_1, h_2, ..., h_m \in X \cup Id$. In other words, the depth of a dead end is the integer $m$ such that all paths of length $m$ emanating from $w$ remain in the ball $B_n$ centered at the identity, but for which there exists a path of length $m+1$ which leaves $B_n$.*

Clearly all dead ends have depth greater than or equal to 1. The depth of a dead end essentially describes how severe the dead end behavior is; if a group has a dead end $w$ with depth $k \geq 1$, we say that $w$ is a *k-pocket* in the Cayley graph of the group. We will show that while $F(p+1)$ has dead ends, it does not have deep k-pockets, because all dead ends in $F(p+1)$ have depth less than or equal to 2.

## 4.2 Dead Ends in $F(p+1)$

Cleary and Taback [10] showed that there are dead ends of depth 2 in $F(2)$. We now generalize their result to $F(p+1)$ for all $p \in \mathbb{N}$.

**Theorem 4.2.1.** *All dead ends in $F(p+1)$ under the standard finite generating set are of the form given in Figure 4.1 where the subtrees labeled $e_1, ..., e_p$ and $f_0$ in both $T_-$ and $T_+$ are all leaves, the subtree labeled $b_0$ in $T_+$, at least one of the subtrees labeled $f_1, ..., f_p$ in $T_-$, and at least one of the subtrees labeled $f'_1, ..., f'_p$ in $T_+$ are non-empty; all other labeled subtrees can be empty but may not be.*

*Proof.* We begin by choosing an arbitrary dead end $w = (T_-, T_+)$ and then evaluating what requirements must be made on $w$ so that $wg^{\pm 1}$ with $g \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$ will have length less than $|w|$ for all possible choices of $g$. We can assume that $w$ satisfies the required conditions of Theorem 2.1.2, because if it does not, we know by Theorem 2.1.3 that $|wg^{\pm 1}| > |w|$, which means $w$ is not a dead end. Because of this, we can assume that the negative tree in the tree-pair diagram has the form given in Figure 4.2. Also, because $w$ will satisfy the required conditions of Theorem 2.1.2, we know that no carets will need to be added to either $T_-$ or $T_+$ to perform multiplication by a generator on the right; it follow from Remark 1.2.1 that all carets in $T_+$ will remain unchanged by multiplication by a generator, and that multiplying by a generator will only change the type of one caret of $T_-$.

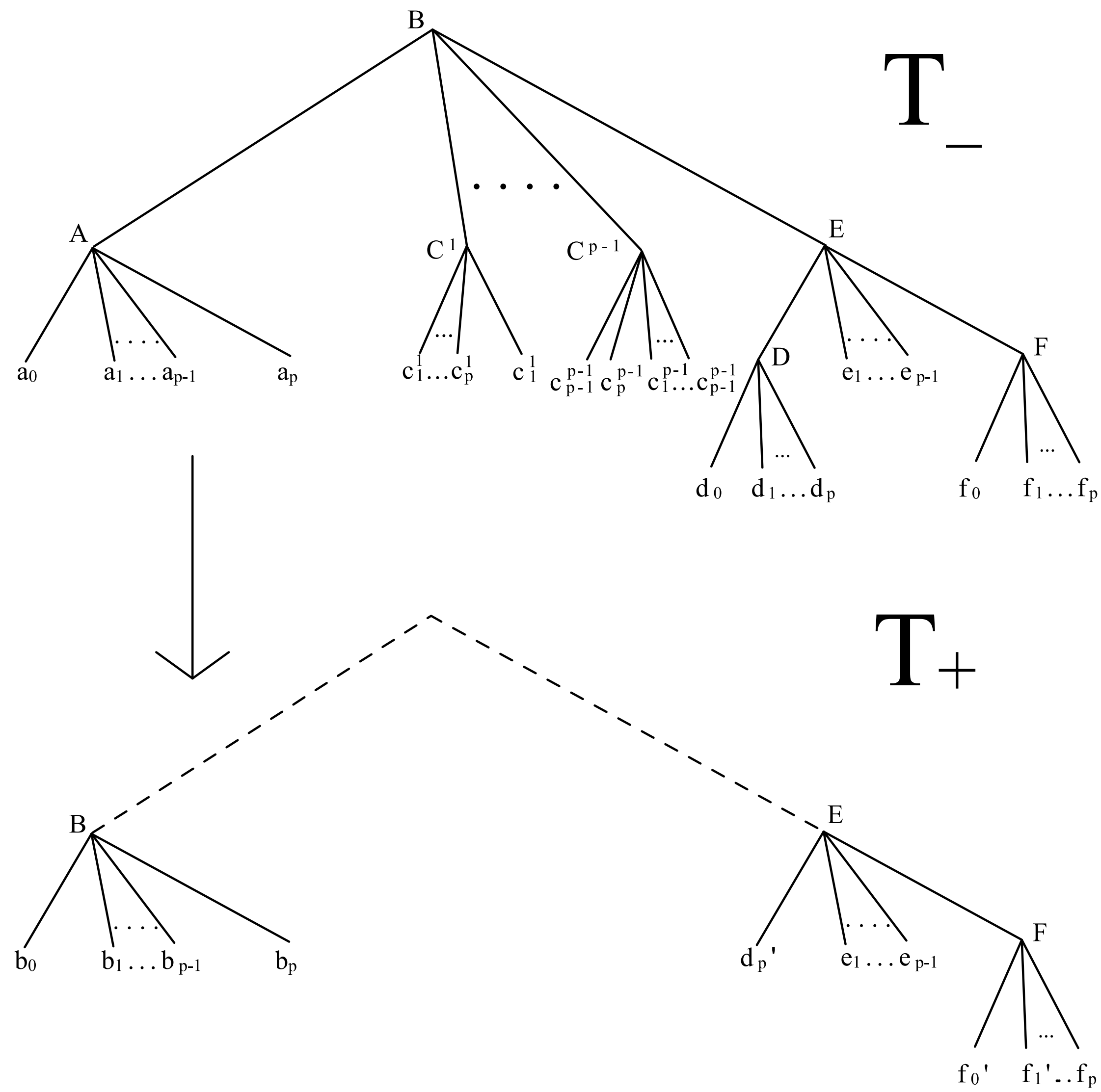


Figure 4.1: **Form of All Dead Ends** in $F(p+1)$ where $a_0, ..., a_p, b_1, ..., b_p, c_1^1, ..., c_1^1, ..., ..., c_{p-1}^{p-1}, ..., c_{p-1}^{p-1}, d_0, ..., d_p, d'_p, f_1, ..., f_p, f'_1, ..., f'_p$ are all (possibly) empty subtrees, $b_0$ is a non-empty subtree, $e_1, ..., e_{p-1}, f_0$ are all leaves, and where at least one of the $f_1, ..., f_p$ and at least one of the $f'_1, ..., f'_p$ are not leaves, in a way that prevents $\wedge_F$ from canceling. Also, if $\wedge_{C^i}$ is of type $\mathcal{M}_\emptyset^i$ in $T_-$, then $\wedge_{C^i}$ must be of type $\mathcal{L}_\mathrm{L}$ in $T_+$; similarly, if $\wedge_D$ is of type $\mathcal{M}_\emptyset^p$ in $T_-$, then $\wedge_D$ cannot be of type $\mathcal{R}_\mathrm{R}$ or $\mathcal{R}_\emptyset$ in $T_+$.

Throughout this proof, we will refer to caret types $\mathcal{R}_*$ and $\mathcal{M}^i_*$ where $\mathcal{R}_*$ denotes any element of the set $\{\mathcal{R}_\emptyset, \mathcal{R}_i, \mathcal{R}_\mathrm{R}\}$ and where $\mathcal{M}^i_*$ denotes any element of the set $\{\mathcal{M}^i_\emptyset, \mathcal{M}^i_j\}$, although we may change the indexes as needed to distinguish them from other indexes in the given case. We also note that if we say two carets in a given case both have type $\mathcal{R}_*$, for example, it is not assumed that they are both of the same caret type. For example, if in a given case we say that $\wedge_i$ in $T_-$ is of type $\mathcal{R}_*$ and $\wedge_i$ in $T_+$ is of type $\mathcal{R}_*$, it may be the case that $\wedge_i = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$ or $\wedge_i = (\mathcal{R}_\mathrm{R}, \mathcal{R}_\mathrm{R})$, but it may also be the case that $\wedge_i = (\mathcal{R}_\emptyset, \mathcal{R}_j)$ or $\wedge_i = (\mathcal{R}_\mathrm{R}, \mathcal{R}_\emptyset)$ or $\wedge_i = (\mathcal{R}_j, \mathcal{R}_k)$ where $j \neq k$, etc.

So now we consider multiplying our dead end element $w$ by each generator; by evaluating how the multiplication by each generator modifies a caret in $T_-$ and ruling out any caret types in $T_-$ or $T_+$ that would result in a caret pairing whose weight increases after multiplication by that generator, we can refine our picture of our dead end $w$:

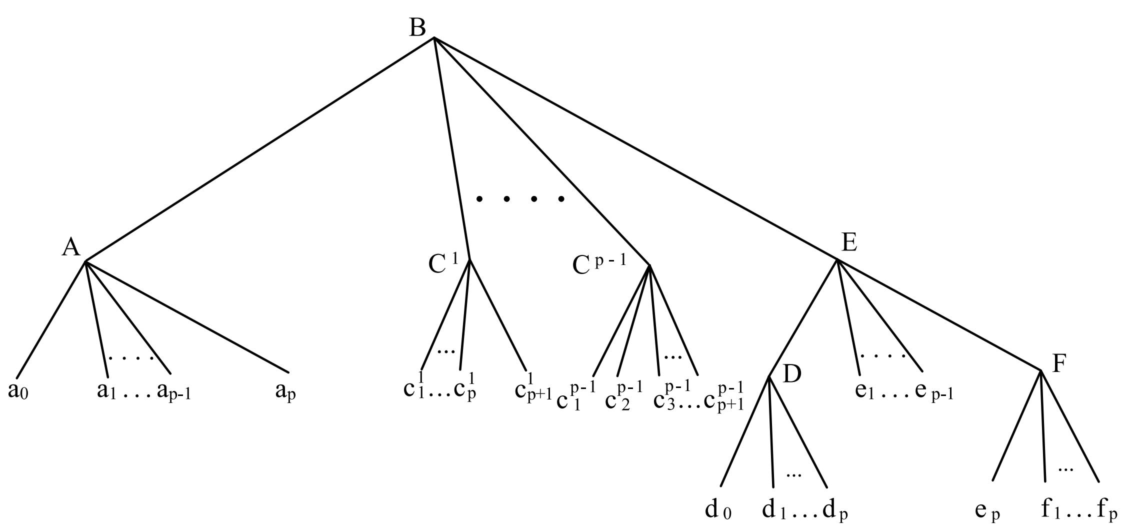


Figure 4.2: Form of Negative Tree of Dead End in $F(p+1)$ where $a_0, ..., a_p, c^1_1, ..., c^1_1, ..., ..., c^{p-1}_{p-1}, ..., c^{p-1}_{p-1}, d_0, ..., d_p, e_1, ...e_p, f_1, ..., f_p$ are all (possibly) empty subtrees

1. $wx_0$: Multiplying $w$ by $x_0$ changes the caret labeled $B$ in $T_-$ from type $\mathcal{L}_\mathrm{L}$ to type $\mathcal{R}_1$. According to Fordham's table of weights for caret pairings in tree-pair diagrams of elements of $F(p+1)$ (see table 2.1), $\wedge_B$ in $T_+$ must be of type $\mathcal{L}_\mathrm{L}$ because this is the only caret pairing in $(T_-, T_+)$ for $\wedge_B$ such that $w(\wedge_B \in ((Tx)_-, (Tx)_+) < w(\wedge_B \in (T_-, T_+))$ and therefore the only caret pairing for $\wedge_B$ which will result in $|wx_0| < |w|$.

2. $wx_0^{-1}$: Multiplying $w$ by $x_0^{-1}$ changes the caret labeled $E$ from type $\mathcal{R}_*$ to type $\mathcal{L}_\mathrm{L}$. First we enumerate the conditions that will determine the type of $\wedge_E$ in $T_-$, which is $\mathcal{R}_* \in \{\mathcal{R}_\emptyset, \mathcal{R}_i, \mathcal{R}_\mathrm{R}\}$.

   (a) If $e_i$ is a non-empty subtree in $T_-$ for some $i \in \{1, ..., p\}$, then the type of $\wedge_E$ in $T_-$ is $\mathcal{R}_*{=}\mathcal{R}_i$.

   (b) If $e_i$ is a leaf in $T_-$ for all $i \in \{1, ..., p\}$ and $\wedge_F$ in $T_-$ is not of type $\mathcal{R}_\emptyset$, then the type of $\wedge_E$ in $T_-$ is $\mathcal{R}_*{=}\mathcal{R}_\mathrm{R}$.

   (c) If $e_i$ is a leaf in $T_-$ for all $i \in \{1, ..., p\}$ and $\wedge_F$ in $T_-$ is of type $\mathcal{R}_\emptyset$, then the type of $\wedge_E$ in $T_-$ is $\mathcal{R}_*{=}\mathcal{R}_\emptyset$.

   According to Fordham's table, $\wedge_E$ in $T_+$ must be of one of the following types: $\mathcal{M}_q^p$ or $\mathcal{R}_*$(as long as it does not produce the caret pairing $\wedge_E = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$ for $\wedge_E$ in $(T_-, T_+)$); these are the only possible caret pairings for $\wedge_E$ in $w$ which will result in $|wx_0| < |w|$ for all three cases above. Additionally, if the type of $\wedge_E$ in $T_-$ is $\mathcal{R}_*{=}\mathcal{R}_i$, $\wedge_E$ in $T_+$ may be of type $\mathcal{M}_\emptyset^j$ when $i \leq j$; in this case only, this will also result in $|wx_0^{-1}| < |w|$.

3. $wx_i$: Multiplying $w$ by $x_i$ changes the caret labeled $C^i$ from type $\mathcal{M}_*^i$ to type $\mathcal{R}_*$. First we enumerate the conditions imposed by the type of $\wedge_{C^i}$ in $T_-$, which is $\mathcal{M}_*^i \in \{\mathcal{M}_\emptyset^i, \mathcal{M}_j^i\}$.

(a) If $\wedge_{C^i} = \mathcal{M}_j^i$ in $T_-$, then $\wedge_{C^i} = \mathcal{R}_j$ in $(Tx_i)_-$.

(b) If $\wedge_{C^i} = \mathcal{M}_\emptyset^i$ in $T_-$, then $\wedge_{C^i} = \mathcal{R}_{i+1}$ in $(Tx_i)_-$.

According to Fordham's table, $\wedge_{C^i}$ in $T_+$ must be of type $\mathcal{L}_\mathrm{L}$; this is the only possible caret pairing for $\wedge_{C^i}$ in $w$ which will result in $|wx_0| < |w|$ for both cases above. Additionally, if the type of $\wedge_{C^i}$ in $T_-$ is $\mathcal{M}_*^i{=}\mathcal{M}_\emptyset^i$, $\wedge_{C^i}$ in $T_+$ may be of any type; in this case only, this will also result in $|wx_i| < |w|$.

4. $wx_i^{-1}$: Multiplying $w$ by $x_i^{-1}$ changes the caret labeled $E$ from type $\mathcal{R}_*$ to type $\mathcal{M}_*^i$. The conditions that will determine the type of $\wedge_E$ in $T_-$ are the same as those enumerated for the case when we are multiplying by $x_0^{-1}$. In addition:

   (a) If $\wedge_E$ is of type $\mathcal{R}_j$ in $T_-$ such that $j \leq p - i$, then $\wedge_E$ in $(Tx_i^{-1})_-$ is type $\mathcal{M}_j^i$.

   (b) Otherwise, $\wedge_E$ in $(Tx_i^{-1})_-$ is type $\mathcal{M}_\emptyset^i$.

   According to Fordham's table, there is no possible caret pairing for $\wedge_E$ in $(T_-, T_+)$ which will decrease the weight of $\wedge_E$ when multiplying by $x_i^{-1}$ if $\wedge_E$ is of type $\mathcal{R}_j$ in $T_-$ where $j \leq p - i$. So $\wedge_E$ must be of type $\mathcal{R}_j$ such that $j > p - i$ or of type $\mathcal{R}_\emptyset$ or $\mathcal{R}_\mathrm{R}$ in $T_-$. Fordham's table also shows us that $\wedge_E$ in $T_+$ must be of one of the following types: $\mathcal{M}_q^p$ when $i < q$, $\mathcal{M}_\emptyset^k$ when $\wedge_E$ in $T_-$ is of type $\mathcal{R}_j$ and $j > p - i$, or $\mathcal{R}_*$ (as long as it does not produce the caret pairing $\wedge_E = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$); these are the only possible caret pairings for $\wedge_E$ in $w$ which will result in $|wx_i^{-1}| < |w|$ for all cases enumerated above.

5. $wx_p$: Multiplying $w$ by $x_p$ changes the caret labeled $D$ from type $\mathcal{M}_*^p$ to type $\mathcal{R}_*$. First we enumerate the conditions that will determine the type of $\wedge_D$ in $T_-$, which is $\mathcal{M}_*^p \in \{\mathcal{M}_\emptyset^p, \mathcal{M}_j^p\}$:

(a) If there is an $i \in \{1, ..., p\}$ such that $d_i$ is not a leaf, then $\wedge_D$ in $T_-$ is of type $\mathcal{M}_i^p$.

(b) If the node $d_i$ is a leaf, for all $i \in \{1, ..., p\}$, then $\wedge_D$ in $T_-$ is of type $\mathcal{M}_\emptyset^p$.

We also note that if $\wedge_D$ in $T_-$ is of type $\mathcal{M}_j^p$, then $\wedge_D$ in $(Tx_p)_-$ is of type $\mathcal{R}_j$; or if $\wedge_D$ in $T_-$ is of type $\mathcal{M}_\emptyset^p$, then $\wedge_D$ in $(Tx_p)_-$ is of type $\mathcal{R}_\mathrm{R}$ or $\mathcal{R}_\emptyset$. According to Fordham's table, $\wedge_D$ in $T_+$ must be of type $\mathcal{L}_\mathrm{L}$, $\mathcal{R}_j$, $\mathcal{M}_\emptyset^k$ or $\mathcal{M}_l^k$; this is the only possible caret pairing for $\wedge_D$ in $w$ which will result in $|wx_p| < |w|$ for all the above cases. Additionally, if the type of $\wedge_D$ in $T_-$ is $\mathcal{M}_*^p$=$\mathcal{M}_j^p$, $\wedge_D$ in $T_+$ may be of any type and in this case only, this will also result in $|wx_i| < |w|$.

6. $wx_p^{-1}$: Multiplying $w$ by $x_p^{-1}$ changes the caret labeled $E$ from type $\mathcal{R}_*$ to type $\mathcal{M}_*^p$. The conditions that will determine the type of $\wedge_E$ in $T_-$ are the same as those enumerated for the case when we are multiplying by $x_0^{-1}$. In addition:

(a) If $\wedge_E$ is of type $\mathcal{R}_j$ in $T_-$, then $\wedge_E$ in $(Tx_p^{-1})_-$ is type $\mathcal{M}_j^p$.

(b) If $\wedge_E$ is of type $\mathcal{R}_\emptyset$ or $\mathcal{R}_\mathrm{R}$ in $T_-$, then $\wedge_E$ in $(Tx_p^{-1})_-$ is type $\mathcal{M}_\emptyset^p$.

According to Fordham's table, there is no possible caret pairing for $\wedge_E$ in $(T_-, T_+)$ which will decrease the weight of $\wedge_E$ when multiplying by $x_p^{-1}$ if $\wedge_E$ is of type $\mathcal{R}_j$ in $T_-$. So $\wedge_E$ must be of type $\mathcal{R}_\emptyset$ or $\mathcal{R}_\mathrm{R}$ in $T_-$. Fordham's table also shows us that $\wedge_E$ in $T_+$ must be of one of the following types: $\mathcal{R}_\emptyset$ or $\mathcal{R}_\mathrm{R}$ (as long as it does not produce the caret pairing $\wedge_E = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$); these are the only possible caret pairings for $\wedge_E$ in $w$ which will result in $|wx_p^{-1}| < |w|$ when $\wedge_E$ is of type $\mathcal{R}_\emptyset$ or $\mathcal{R}_\mathrm{R}$ in $T_-$.

We note that multiplying by each $x_j^{-1}$ for $j = 0, 1, 2, ..., p$ imposes a condition on the caret $\wedge_E$. We recall:

1. The possible caret pairings for $\wedge_E$ in $(T_-, T_+)$ determined by multiplication because they reduce caret weight after $x_0^{-1}$ are:

   (a) $(\mathcal{R}_*, \mathcal{R}_*)$ excluding $(\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$

   (b) $(\mathcal{R}_*, \mathcal{M}_q^p)$

   (c) $(\mathcal{R}_i, \mathcal{M}_\emptyset^j)$ such that $i \leq j$

2. The possible caret pairings for $\wedge_E$ in $(T_-, T_+)$ determined because they reduce caret weight after multiplication by $x_i^{-1}$ with $i = 1, 2, 3, ..., p-1$ are:

   (a) $(\mathcal{R}_*', \mathcal{R}_*)$ excluding $(\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$ where $\mathcal{R}_*' \in \{\mathcal{R}_\emptyset, \mathcal{R}_\mathrm{R}, \mathcal{R}_j | j > p - i\}$

   (b) $(\mathcal{R}_*', \mathcal{M}_q^p)$ with $q > i$ where $\mathcal{R}_*'$ is as defined above

   (c) $(\mathcal{R}_j, \mathcal{M}_\emptyset^k)$ such that $j > p - i$

   These conditions must hold for all $i \in \{1, 2, ..., p-1\}$.

3. The possible caret pairings for $\wedge_E$ in $(T_-, T_+)$ determined by multiplication because they reduce caret weight after $x_p^{-1}$ are:

   (a) $(\mathcal{R}_\emptyset, \mathcal{R}_\mathrm{R})$

   (b) $(\mathcal{R}_\mathrm{R}, \mathcal{R}_\emptyset)$

   (c) $(\mathcal{R}_\mathrm{R}, \mathcal{R}_\mathrm{R})$

So,the only way for the weight of $\wedge_E$ to be reduced during multiplication by each of the inverses of the generators $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$ is for it to be of one of the following types:

1. $(\mathcal{R}_\emptyset, \mathcal{R}_\mathrm{R})$

2. $(\mathcal{R}_{\mathrm{R}}, \mathcal{R}_{\emptyset})$

3. $(\mathcal{R}_{\mathrm{R}}, \mathcal{R}_{\mathrm{R}})$

This tells us that $e_1, ..., e_p$ must all be leaves in both $T_-$ and $T_+$. Then, in order to guarantee that $\wedge_F$ in $T_-$ does not cancel $\wedge_F$ in $T_+$ and thus violate the required conditions of Theorem 2.1.2, the type of the caret pairing $\wedge_F$ cannot be $(\mathcal{R}_{\emptyset}, \mathcal{R}_{\emptyset})$.

To sum up the possible caret pairings outlined above for the other carets in $(T_-, T_+)$ (none of these carets should cancel, as this would violate the required conditions of Theorem 2.1.2):

1. $\wedge_A$:
   (a) $(\mathcal{L}_*,\mathcal{L}_*)$
   (b) $(\mathcal{L}_*,\mathcal{M}^i_*)$

2. $\wedge_B$:
   (a) $(\mathcal{L}_{\mathrm{L}},\mathcal{L}_{\mathrm{L}})$

3. $\wedge_{C^i}$:
   (a) $(\mathcal{M}^i_*,\mathcal{L}_{\mathrm{L}})$
   (b) $(\mathcal{M}^i_j,*)$ where * can be any caret type

4. $\wedge_D$:
   (a) $(\mathcal{M}^p_*,*)$ where * can be any caret type (as long as $\wedge_D$ is not of type $(\mathcal{M}^p_{\emptyset},\mathcal{R}_{\mathrm{R}})$ or type $(\mathcal{M}^p_{\emptyset}, \mathcal{R}_{\emptyset})$)

This is exactly the set of conditions depicted in Figure 4.1.

□

## 4.3 Depth of Dead Ends in $F(p+1)$

**Theorem 4.3.1.** *All dead ends in $F(p+1)$ with respect to the standard finite generating set have depth equal to 2. Equivalently, there are no k-pockets in $F(p+1)$ for $k > 2$.*

*Proof.* Let $w$ be a dead end in $F(p+1)$. Supppose $|w| = n$; we have seen already that $|wg^{\pm 1}| = n-1$ for $g \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$. So $|wg_1^{\pm 1}g_2^{\pm 1}| \leq n$ for $g_1, g_2 \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$, which shows that $w$ cannot have depth 1, and $|wg_1^{\pm 1}g_2^{\pm 1}g_3^{\pm 1}| \leq n+1$ for $g_1, g_2, g_3 \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$. So, to show that a dead end $w$ in $F(p+1)$ has depth 2, we need only find $g_1, g_2, g_3 \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$ such that $|wg_1^{\epsilon_1}g_2^{\epsilon_2}g_3^{\epsilon_3}| \geq n+1$ where $\epsilon_1, \epsilon_2, \epsilon_3 \in \{-1, 1\}$.

If we consider the tree-pair diagram for $w$ given in Figure 4.1, we can see that multiplying $w$ by $x_0^{-1}$ will result in the tree-pair diagram given in Figure 4.3.

We note that $|wx_0^{-1}| = n-1$. We also note that to multiply $wx_0^{-1}$ by $x_i$ for $i = 1, 2, ..., p$, we would need to add a caret to the tree-pair diagram for $wx_0^{-1}$ on the leaf with index number $e_i$ (note: for $i = p$, $e_p = f_0$); we call this new caret $E^i$. So the tree-pair diagram for $wx_0^{-1}x_i$ will be as shown in Figure 4.4.

We note that since we had to add a caret to the tree-pair diagram for $wx_0^{-1}$ to get $wx_0^{-1}x_i$, by Theorem 2.1.3, $|wx_0^{-1}x_i| \geq n$. We also note that to multiply $wx_0^{-1}x_i$ by $x_j$ where $j = 1, 2, ..., p$, we would need to add a caret to the tree-pair diagram for $wx_0^{-1}x_i$ on the leaf with index number $e_j$, and then by Theorem 2.1.3, $|wx_0^{-1}x_ix_j| > n$. Therefore all dead ends have depth 2 in $F(p+1)$ under the standard finite generating set. □

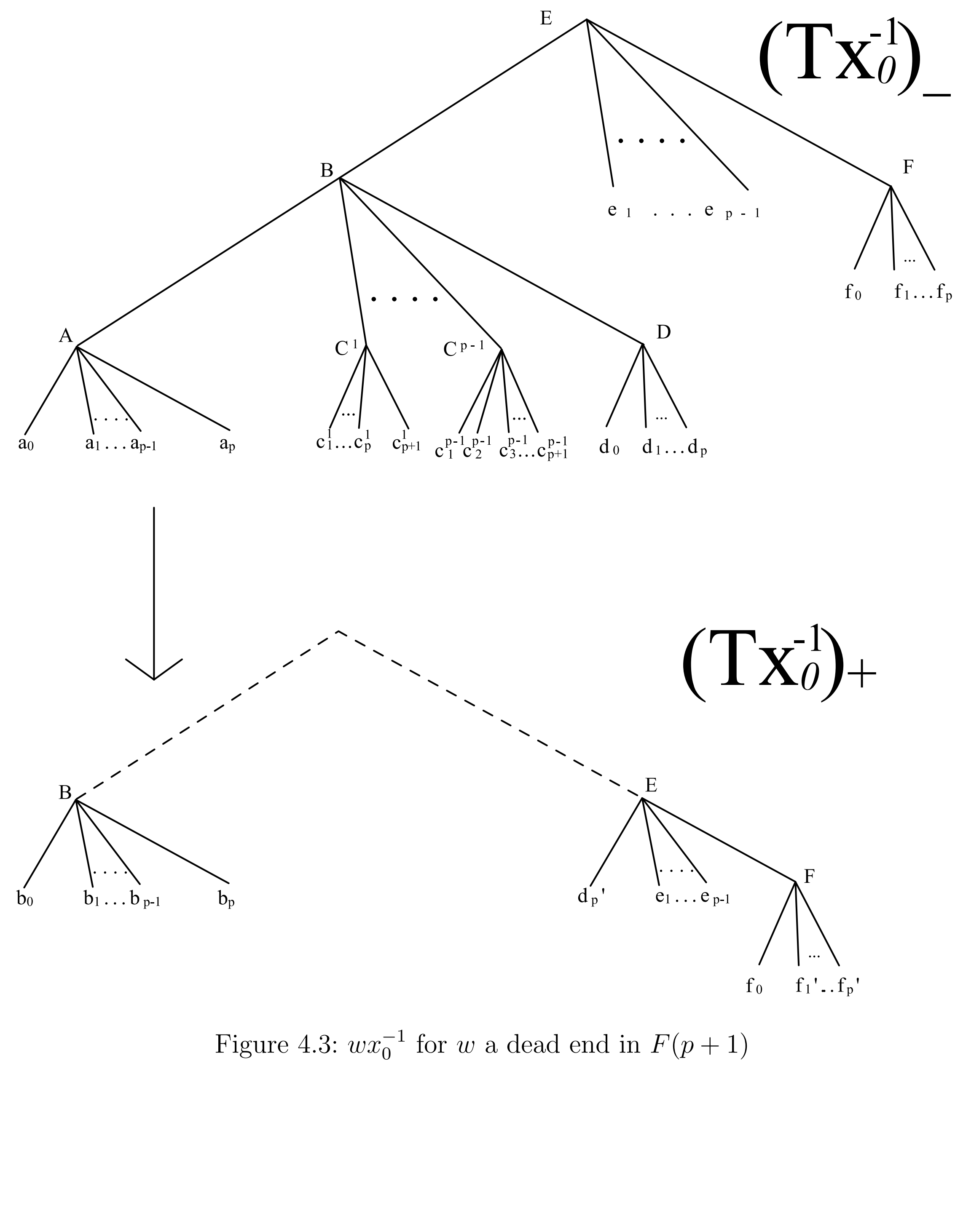


Figure 4.3: $wx_0^{-1}$ for $w$ a dead end in $F(p+1)$

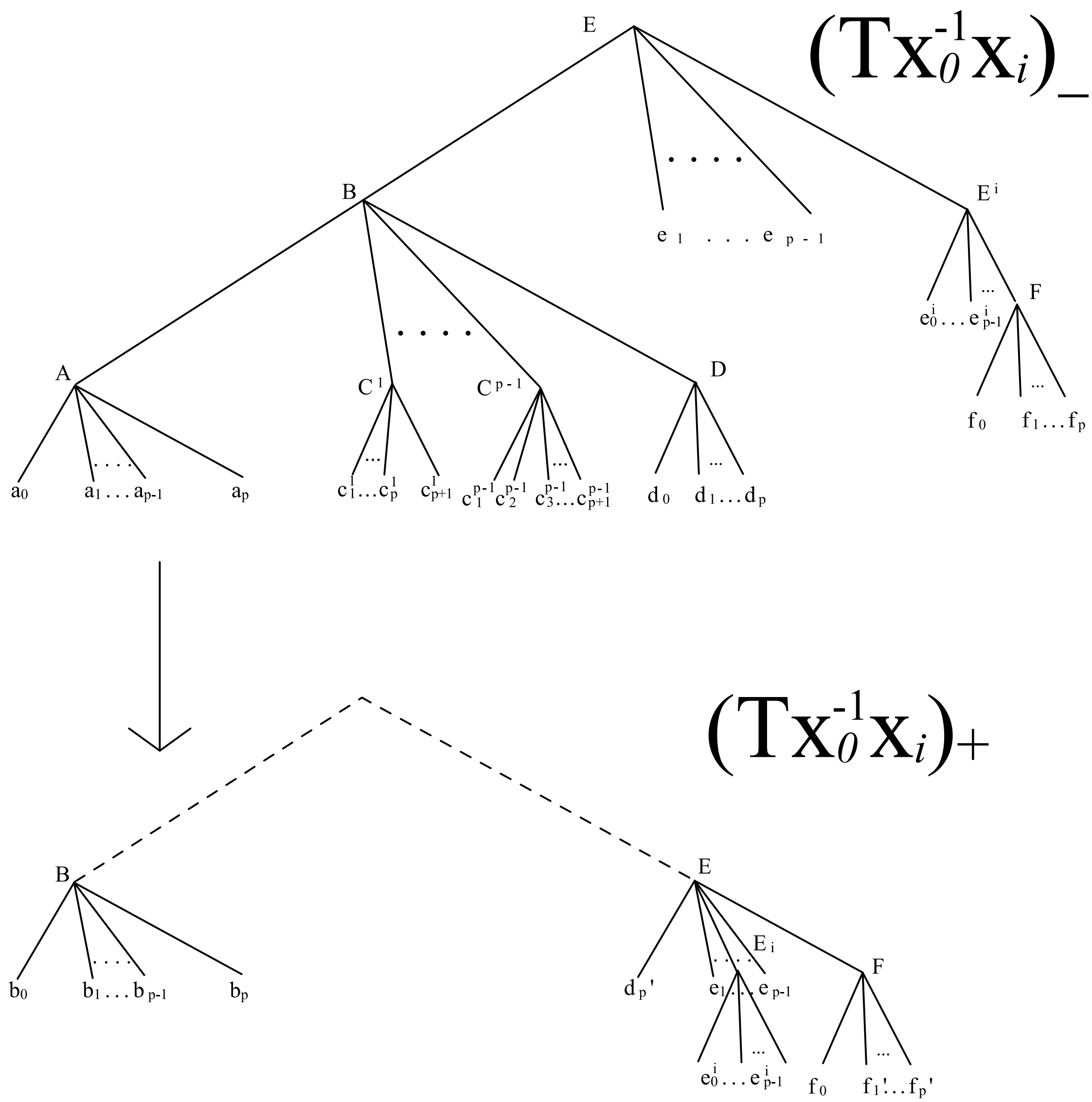


Figure 4.4: $wx_0^{-1}x_i$ for $i = 1, 2, ..., p$ and $w$ a dead end in $F(p+1)$

# Chapter 5

# Thompson's group F(p+1) is not Minimally Almost Convex

## 5.1 Almost Convexity

The concept of almost convexity was first developed by Cannon in [5] to develop algorithms for drawing the Cayley graphs of groups. If a group is almost convex with respect to a given generating set, then an algorithm exists which can be used to draw the portion of the Cayley graph of $G$ which can be depicted by the ball of radius $n$ centered at the identity [5]. Minimal almost convexity is a weaker condition than almost convexity.

We let $B_n$ denote the ball of radius $n$ in the Cayley graph $\Gamma$ of a group $G$ with the finite generating set $X$. The *convexity function* $c(n)$ of the group is defined to be $c(n) = max\{d_{B_n}(g,h) | g,h \in B_n \text{ and } d_\Gamma(g,h) = 2\}$.

**Definition (almost convex) 5.1.1.** *(due to Cannon in [5]). A group $G$ is almost convex with respect to the finite generating set $X$ if its convexity function $c(n)$ with respect to the finite generating set $X$ is bounded by a constant $C$, i.e. $c(n) \leq C$ for all $n$.*

The condition of minimal almost convexity is, in a sense, the weakest possible (without being trivial) generalization of almost convexity:

**Definition (minimally almost convex) 5.1.2.** *A group $G$ is minimally almost convex with respect to the finite generating set $X$ if its convexity function $c(n)$ with respect to the finite generating set $X$ is bounded by the constant $2n-1$ for* *sufficiently large $n$, i.e. if there exists a constant $N$ such that for all $n > N$, $c(n) \leq 2n - 1$.*

Since there is always a path from $g$ to $h$ in $B_n$ through the identity which has length in $B_n$ bounded by $2n$ ($g^{-1}h$, for example), we will always have $c(n) \leq 2n$. So showing that a group is not minimally almost convex with respect to a given finite generating set is equivalent to stating that any minimal length path in $B_n$ between two elements $g, h \in B_n$ such that $d_\Gamma(g, h) = 2$ will be $2n$, the same distance as the path $h^{-1}g$ or $g^{-1}h$ through the origin.

In this paper we will show that Thompson's group $F(p+1)$ is not minimally almost convex. The result that Thompson's group $F = F(2)$ is not minimally almost convex has already been proven by Belk and Bux in [1]; this paper uses the same overall structure for the proof that $F(p+1)$ is not minimally almost convex as Belk and Bux used for the proof that $F$ is not minimally almost convex, but all the details have had to be worked out for the extra generators and relators present in the groups $F(p+1)$ when $p > 1$. In addition, all definitions and techniques are translated from Belk and Bux's representation of elements of $F$ using forest diagrams to representations of elements of $F(p+1)$ as tree-pair diagrams, as a simple and useful method for interpreting elements of $F(p+1)$ as forest diagrams in which the action of specific generators can be easily and uniquely identified from changes in the diagram is not yet known to the author.

## 5.2 Thompson's group $F(p+1)$ is not minimally almost convex

Throughout this section $L(g)$ is used to denote the *length* of an element $g$ in the group with respect to the given generating set. Similarly, $d(x, y)$ is used to denote the *distances* between two vertices $x$ and $y$ in the Cayley graph of the group with respect to the given generating set. In addition, $w(\wedge_i)$ is used to denote the *weight* (as given by Fordham's method explained in a previous section) of the pair of carets which have index number $i$ in the tree-pair diagram of the element. For all tree-pair diagrams given in this section, circles are used to denote (possibly empty) subtrees, and leaves without circles are used to denote leaves. Also, throughout this section, unless otherwise stated, $x_i^{\pm 1}$ stands for any $i \in \{1, 2, ..., p-1\}$.

**Theorem 5.2.1.** *For all $n \leq 1$ there exist $l, r \in F(p+1)$ such that with respect to the standard finite generating set $\{x_0, x_1, \ldots, x_{p-1}, x_p\}$:*

1. $d(l, r) = 2$
2. $L(l) = L(r) = 2n + 2$
3. *Any path $p$ from $l$ to $r$ which remains in $B_{2n+2}$ is such that $L(p) \geq 4n + 4$.*

We choose $r = x_p^n x_0^{-(n+1)} x_p^{-1}$ and $l = r x_0^2$.

*Proof of part 1 of Theorem 5.2.1.* . Since $l = r x_0^2$,

$$r^{-1} l = r^{-1} r x_0^2$$

$$\text{So } d(l, r) = L(r^{-1} l) = 2$$

□

*Proof of part 2 of Theorem 5.2.1.* . We compute the lengths of $r$ and $l$:

For $r$:

$\wedge_0 = (\mathcal{L}_\emptyset, \mathcal{L}_\emptyset)$ so $w(\wedge_0) = 0$

$\wedge_i = (\mathcal{L}_\text{L}, \mathcal{M}_\emptyset^p)$ so $w(\wedge_i) = 2$ for $i = 1, ..., n$

$\wedge_{n+1} = (\mathcal{L}_\text{L}, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+1}) = 1$

$\wedge_{n+2} = (\mathcal{M}_\emptyset^p, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+2}) = 1$

$\wedge_{n+3} = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+3}) = 0$

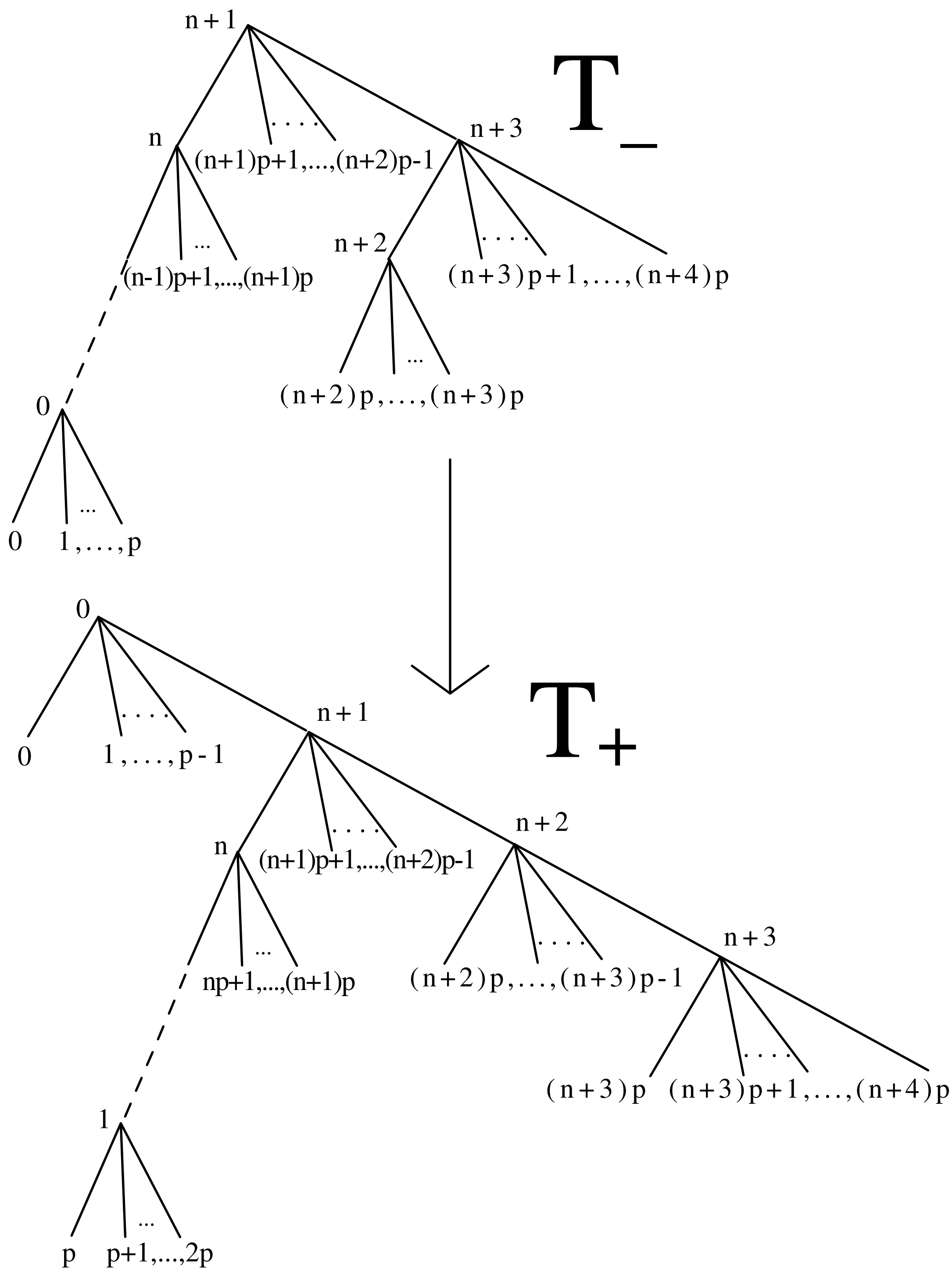


Figure 5.1: $r$ in $F(p+1)$

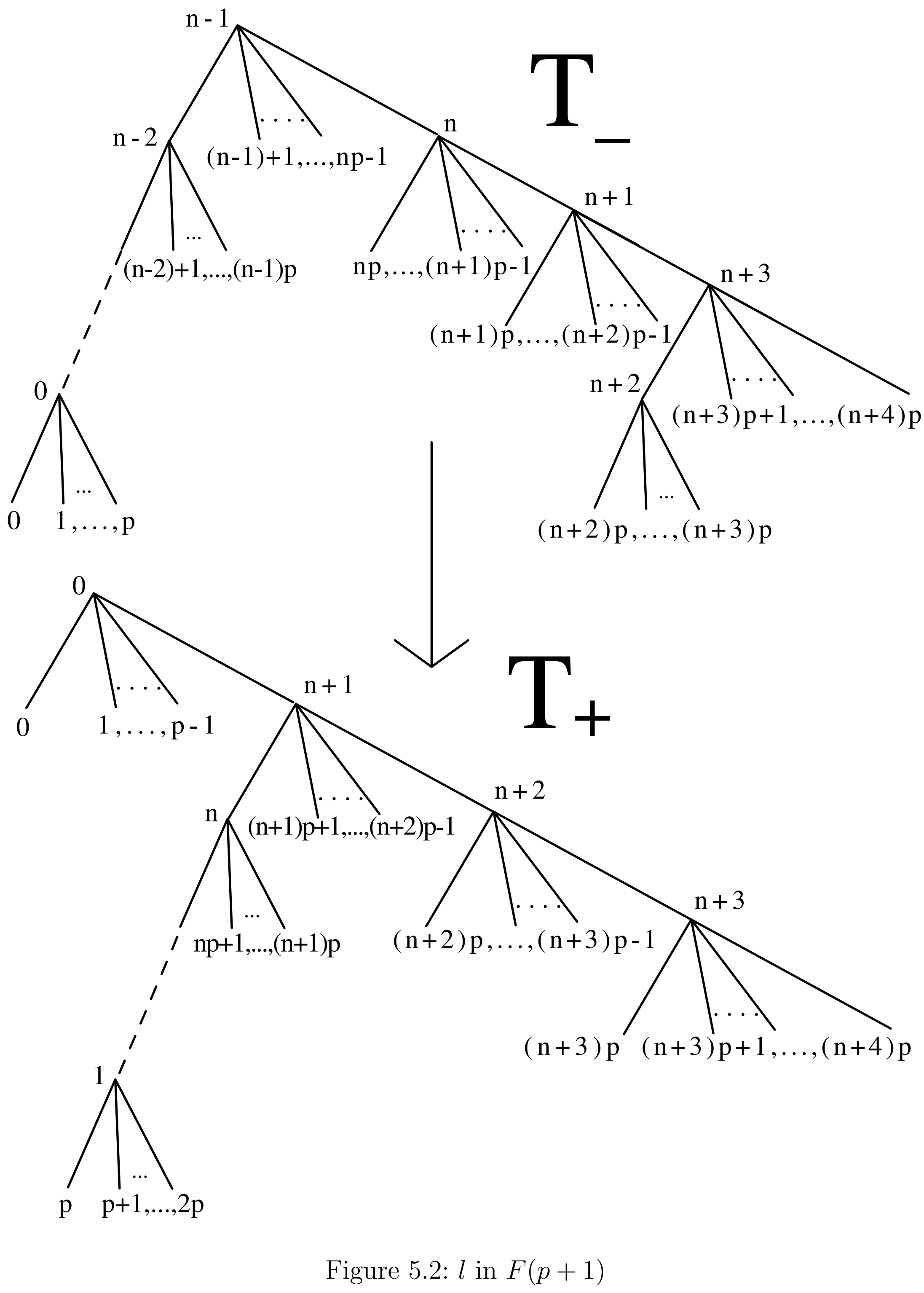


Figure 5.2: $l$ in $F(p+1)$

Summing the weights for each of these yields the length of $r$:

$L(r) = 2n + 2$

For $l$:

$\wedge_0 = (\mathcal{L}_\emptyset, \mathcal{L}_\emptyset)$ so $w(\wedge_0) = 0$

$\wedge_i = (\mathcal{L}_\mathrm{L}, \mathcal{M}^p_\emptyset)$ so $w(\wedge_i) = 2$ for $i = 1, ..., n-1$

$\wedge_n = (\mathcal{R}_\mathrm{R}, \mathcal{M}^p_\emptyset)$ so $w(\wedge_n) = 1$

$\wedge_{n+1} = (\mathcal{R}_p, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+1}) = 2$

$\wedge_{n+2} = (\mathcal{M}^p_\emptyset, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+2}) = 1$

$\wedge_{n+3} = (\mathcal{R}_\emptyset, \mathcal{R}_\emptyset)$ so $w(\wedge_{n+3}) = 0$

Summing the weights for each of these yields the length of $l$:

$L(l) = 2n + 2$

□

The rest of this section will be devoted to a proof of part 3 of Theorem 5.2.1.

### 5.2.1 Any Path from $l$ to $r$ which passes through vertices with distance at least $2n+3$ apart must leave $B_{2n+2}$

**Lemma 5.2.2.** *(Belk and Bux [1], explanation for Lemma 4.1) With respect to the standard finite generating set for $F(p+1)$, if there are two vertices $h_r$ and $h_l$ on the path $p$ from $l$ to $r$ such that $d(h_l, h_r) \geq 2n+3$ , then $L(p) \geq 4n+4$.*

*Proof.* (Belk and Bux [1], explanation for Lemma 4.1)

By definition,

$$L(p) \geq d(l, h_l) + d(h_l, h_r) + d(h_r, r)$$

And by the triangle inequality,

$$d(h_l, h_r) \leq d(h_l, l) + d(l, r) + d(r, h_r)$$

We already know that $d(l,r) = 2$. Substituting this into the second inequality and solving each inequality for $d(l,h_l) + d(r,h_r)$ yields the following two equations:

$$L(p) - d(h_l,h_r) \geq d(l,h_l) + d(r,h_r) \text{ and } d(h_l,h_r) - 2 \leq d(l,h_l) + d(r,h_r)$$

Putting these together with $d(h_l,h_r) \geq 2n+3$ yields:

$$L(p) \geq 2d(h_l,h_r) - 2 \geq 4n+4$$

□

So now we seek to find vertices on the path $p$ that satisfy the condition of Lemma 5.2.2.

### 5.2.2 Finding the vertex $h_r$

First we seek to find candidates for $h_r$:

**Lemma 5.2.3.** *Any path $p$ from $l$ to $r$ which remains in $B_{2n+2}$ must pass through the vertex $rx_px_0$.*

*Proof.* We have already calculated that $L(r) = 2n+2$.
We begin by computing the length of $rg$ where $g^{\pm 1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$:

1. $rx_0$: Multiplying $r$ by $x_0$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+1}$ in $T_-$ from type $\mathcal{L}_\mathrm{L}$ to type $\mathcal{R}_p$. This changes the caret pairing from ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_\emptyset$), which has weight 1, to ($\mathcal{R}_p$,$\mathcal{R}_\emptyset$), which has weight 2, thereby increasing the length by one. So $L(rx_0) = 2n+3$.

2. $rx_0^{-1}$: Multiplying $r$ by $x_0^{-1}$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+3}$ in $T_-$ from type $\mathcal{R}_\emptyset$ to type $\mathcal{L}_\mathrm{L}$. This changes the caret pairing from ($\mathcal{R}_\emptyset$,$\mathcal{R}_\emptyset$), which has weight 0, to ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_\emptyset$), which has weight 1, thereby increasing the length by one. So $L(rx_0^{-1}) = 2n+3$.

3. $rx_i$: Multiplying $r$ by $x_i$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_i$ because we must add a middle caret child to the $ith$ leaf of the root in $T_-$, so we know from Theorem 2.1.2 that $L(rx_i) > 2n+2$, thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one. So $L(rx_i) = 2n + 3$.

4. $rx_i^{-1}$: Multiplying $r$ by $x_i^{-1}$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+3}$ in $T_-$ from type $\mathcal{R}_\emptyset$ to type $\mathcal{M}_\emptyset^i$. This changes the caret pairing from ($\mathcal{R}_\emptyset$,$\mathcal{R}_\emptyset$), which has weight 0, to ($\mathcal{M}_\emptyset^i$,$\mathcal{R}_\emptyset$), which has weight 1, thereby increasing the length by one. So $L(rx_i^{-1}) = 2n + 3$.

5. $rx_p$: Multiplying $r$ by $x_p$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+2}$ in $T_-$ from type $\mathcal{M}_\emptyset^i$ to type $\mathcal{R}_\emptyset$. This changes the caret pairing from ($\mathcal{M}_\emptyset^i$,$\mathcal{R}_\emptyset$), which has weight 1, to ($\mathcal{R}_\emptyset$,$\mathcal{R}_\emptyset$), which has weight 0, thereby decreasing the length by one. So $L(rx_p) = 2n + 1$.

6. $rx_p^{-1}$: Multiplying $r$ by $x_p^{-1}$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_p^{-1}$ because we must add a right caret child to the last leaf of the tree in $T_-$, so we know from Theorem 2.1.2 that $L(rx_p^{-1}) > 2n + 2$, thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one. So $L(rx_p^{-1}) = 2n + 3$.

Because $rx_0$, $rx_0^{-1}$, $rx_i$, $rx_i^{-1}$, and $rx_p^{-1}$ all have length equal to $2n + 3$, we know that any path from $l$ to $r$ which passes through one of these vertices must leave $B_{2n+2}$. Therefore, any path from $l$ to $r$ which does not leave $B_{2n+2}$ must pass through the vertex $rx_p$.

Now we proceed by considering the length of $rx_pg$ where $g^{\pm 1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$:

1. $rx_px_0$: Multiplying $rx_p$ by $x_0$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+1}$ in $T_-$ from type $\mathcal{L}_\mathrm{L}$ to type $\mathcal{R}_\emptyset$. This changes the caret pairing from ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_\emptyset$), which has weight 1, to ($\mathcal{R}_\emptyset$,$\mathcal{R}_\emptyset$), which has weight 0, thereby decreasing the length by one. So $L(rx_px_0) = 2n$.

2. $rx_px_0^{-1}$: Multiplying $rx_p$ by $x_0^{-1}$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_0^{-1}$ because we must add a right caret child to the last leaf of the positive and negative trees. This will not affect the types of any other caret pair in the tree-pair diagram, but the added caret pair will be of type ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_\emptyset$), which has weight 1, thereby increasing the total length by one. So $L(rx_px_0^{-1}) = 2n + 2$.

3. $rx_px_i$: Multiplying $rx_p$ by $x_i$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_i$ because we must add a middle caret child to the *ith* leaf of the last caret in the positive and negative trees. This will not affect the types of any other caret pair in the tree-pair diagram, but the added caret pair will be of type ($\mathcal{R}_\emptyset$,$\mathcal{M}_\emptyset^i$), which has weight 1, thereby increasing the total length by one. So $L(rx_px_i) = 2n + 2$.

4. $rx_px_i^{-1}$: Multiplying $rx_p$ by $x_i^{-1}$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_i^{-1}$ because we must add a right caret child to the last leaf of the positive and negative trees. This will not affect the types of any other caret pair in the tree-pair diagram, but the added caret pair will be of type ($\mathcal{M}_\emptyset^i$,$\mathcal{R}_\emptyset$), which has weight 1, thereby increasing the total length by one. So $L(rx_px_i^{-1}) = 2n + 2$.

5. $rx_p^2$: Multiplying $rx_p$ by $x_p$ does not satisfy the required conditions of Theorem 2.1.2 with respect to the generator $x_p$ because we must add two carets to the tree-pair diagram on both the positive and negative trees: a right caret

child to the last leaf and then a middle child to the first leaf of this new caret. This will not affect the types of any other caret pair in the tree-pair diagram, but the added caret pairs will be of type $(\mathcal{R}_\emptyset,\mathcal{M}^i_\emptyset)$, which has weight 1, and $(\mathcal{R}_\emptyset,\mathcal{R}_\emptyset)$, which has weight 0, thereby increasing the total length by one. So $L(rx_p^2) = 2n + 2$.

Since all elements of the form $rx_pg$ where $g^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$ are of length less than or equal to $2n + 2$, they all remain inside $B_{2n+2}$, so we proceed to consider the length of all elements of the form $rx_pg_1g_2$ where $g_1^{\pm1}, g_2^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$:

We begin by looking at the tree-pair diagram for $rx_px_0^{-1}$, which we can see in Figure 5.3.

So we consider the length of $rx_px_0^{-1}g$ where $g^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$: $rx_px_0^{-1}g$ where $g^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$ does not satisfy the required conditions of Theorem 2.1.2 unless $g = x_0$, in which case $rx_px_0^{-1}g$ will reduce to $rx_p$. So we know from Theorem 2.1.2 that $L(rx_px_0^{-1}g) > 2n + 2$ whenever $rx_px_0^{-1}g \neq rx_p$ thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one, so that $L(rx_px_0^{-1}g) = 2n + 3$.

So $L(rx_px_0^{-1}g) = 2n + 3$ whenever $rx_px_0^{-1}g \neq rx_p$. Therefore any path from $l$ to $r$ which remains in $B_{2n+2}$ cannot pass through $rx_px_0^{-1}$.

Now we look at the tree-pair diagram for $rx_px_i$, which we can see in Figure 5.4.

So we consider the length of $rx_px_ig$ where $g^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$: $rx_px_ig$ does not satisfy the required conditions of Theorem 2.1.2 in the cases when $g = x_i, x_p, x_p^{-1}$, so we know from Theorem 2.1.2 that $L(rx_px_ig) > 2n+2$ whenever $g = x_i, x_p, x_p^{-1}$, thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one. So

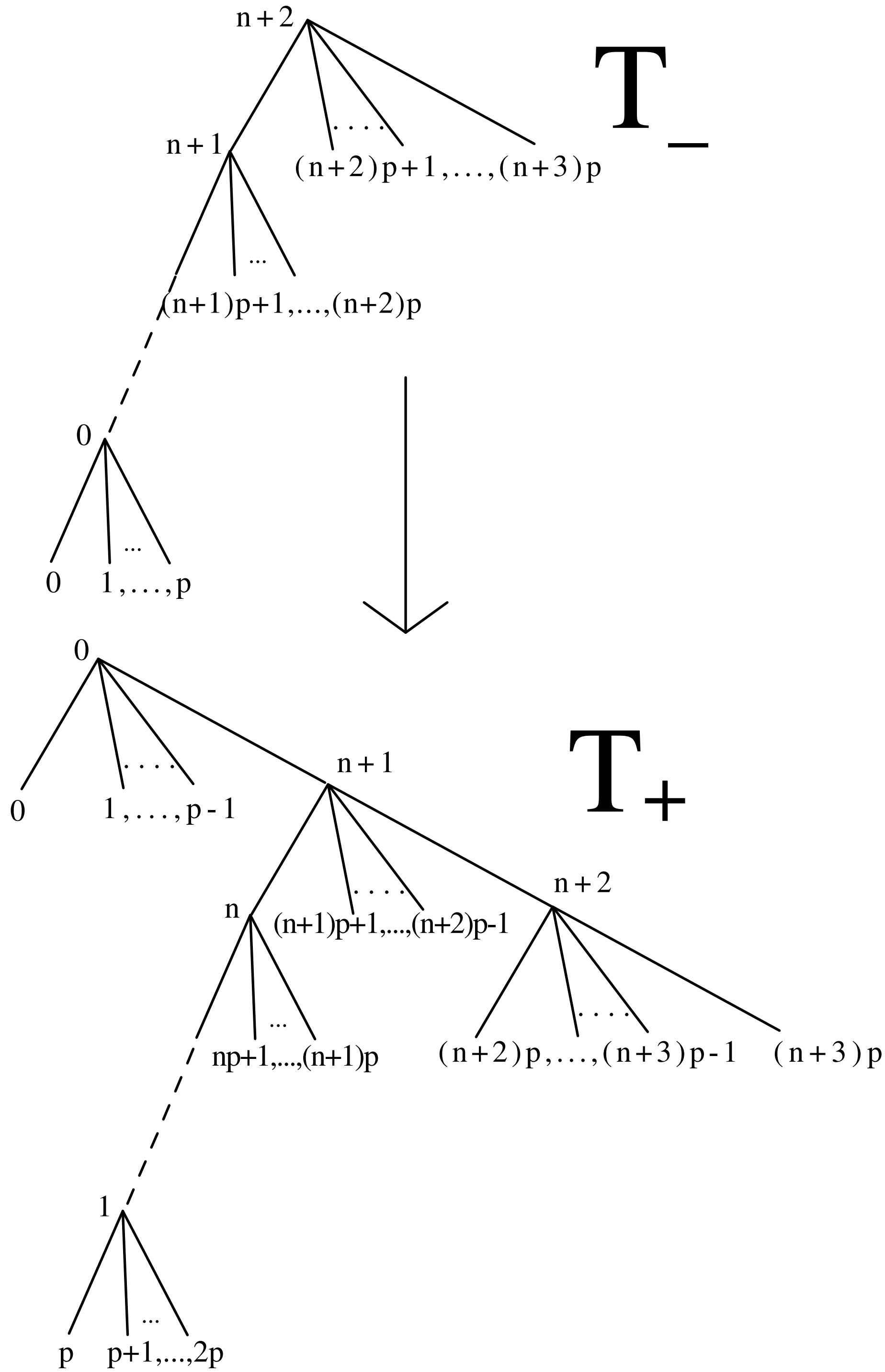


Figure 5.3: $rx_px_0^{-1}$ in $F(p+1)$

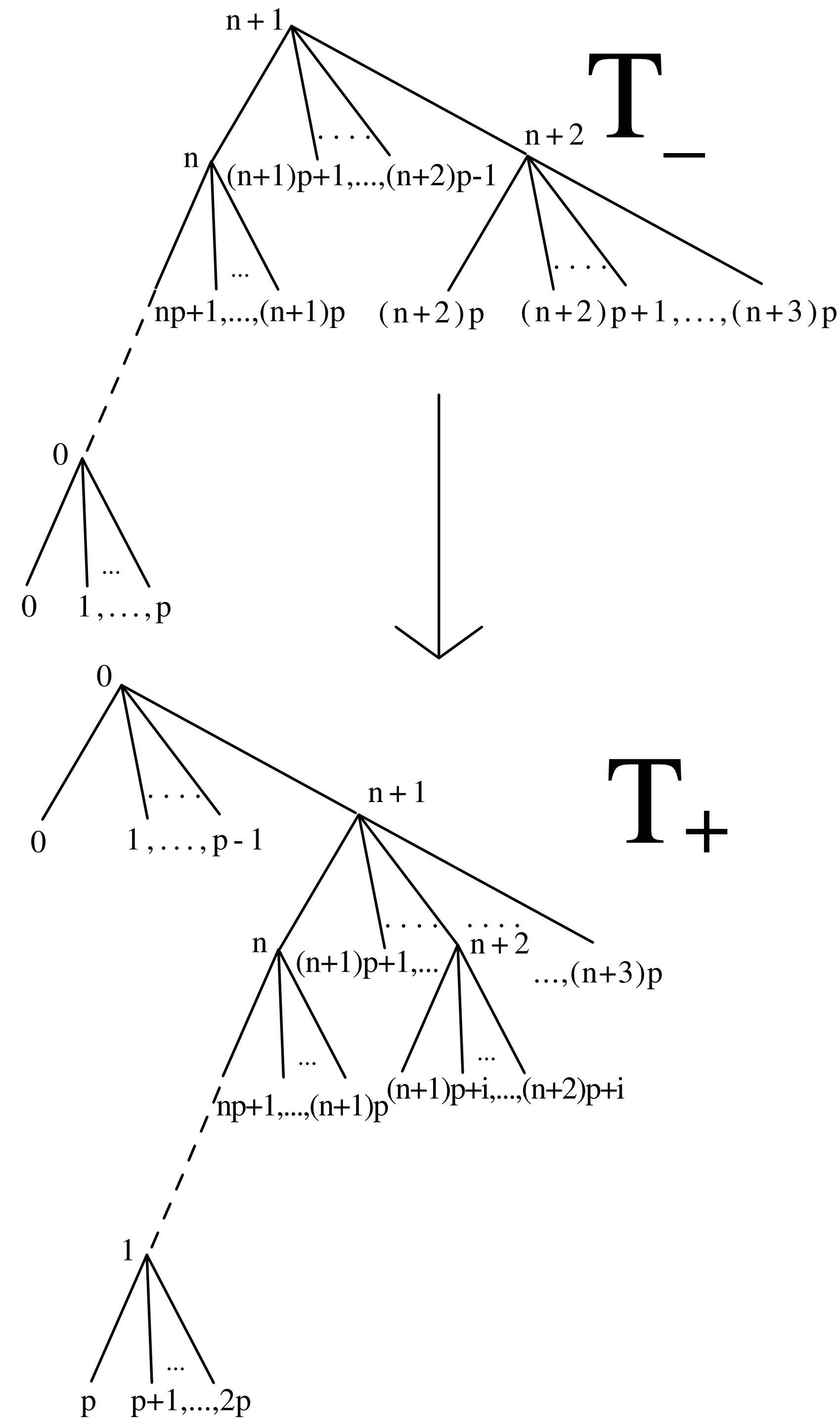


Figure 5.4: $rx_px_i$ in $F(p+1)$

$L(rx_px_ig) = 2n + 3$ whenever $g = x_i, x_p, x_p^{-1}$. Now all that remains is to check the length of $rx_px_ig$ when $g = x_0, x_0^{-1}$.

1. $rx_px_ix_0$: Multiplying $rx_px_i$ by $x_0$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+1}$ in $T_-$ from type $\mathcal{L}_\mathrm{L}$ to type $\mathcal{R}_\emptyset$. This changes the caret pairing from ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_i$), which has weight 1, to ($\mathcal{R}_\emptyset$,$\mathcal{R}_i$), which has weight 2, thereby increasing the length by one. So $L(rx_px_ix_0) = 2n + 3$.

2. $rx_px_ix_0^{-1}$: Multiplying $rx_px_i$ by $x_0^{-1}$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+2}$ in $T_-$ from type $\mathcal{R}_\emptyset$ to type $\mathcal{L}_\mathrm{L}$. This changes the caret pairing from ($\mathcal{R}_\emptyset$,$\mathcal{M}^i_\emptyset$), which has weight 1, to ($\mathcal{L}_\mathrm{L}$,$\mathcal{M}^i_\emptyset$), which has weight 2, thereby increasing the length by one. So $L(rx_px_ix_0^{-1}) = 2n + 3$.

Therefore any path from $l$ to $r$ which remains in $B_{2n+2}$ cannot pass through $rx_px_i$.

Now we look at the tree-pair diagram for $rx_px_i^{-1}$, which we can see in Figure 5.5.

So we consider the length of $rx_px_i^{-1}g$ where $g^{\pm 1} \in \{x_0, x_1, \dots, x_{p-1}, x_p\}$: $rx_px_i^{-1}g$ does not satisfy the required conditions of Theorem 2.1.2 in the cases when $g = x_0^{-1}, x_j(\text{when } j \neq i), x_j^{-1}, x_p, x_p^{-1}$, so we know from Theorem 2.1.2 that $L(rx_px_i^{-1}g) > 2n + 2$ whenever $g = x_0^{-1}, x_j(\text{when } j \neq i), x_j^{-1}, x_p, x_p^{-1}$, thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one. So $L(rx_px_i^{-1}g) = 2n+3$ whenever $g = x_0^{-1}, x_j(\text{when } j \neq i), x_j^{-1}, x_p, x_p^{-1}$. Now all that remains is to check the length of $rx_px_i^{-1}g$ when $g = x_0$.

$rx_px_i^{-1}x_0$: Multiplying $rx_px_i^{-1}$ by $x_0$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+1}$ in $T_-$ from type $\mathcal{L}_\mathrm{L}$ to type $\mathcal{R}_i$. This changes the caret pairing from ($\mathcal{L}_\mathrm{L}$,$\mathcal{R}_\emptyset$), which has weight 1, to ($\mathcal{R}_i$,$\mathcal{R}_\emptyset$), which has weight 2, thereby increasing the length by one. So $L(rx_px_i^{-1}x_0) = 2n + 3$. Therefore any path from $l$ to $r$ which remains in $B_{2n+2}$ cannot pass through $rx_px_i^{-1}$.

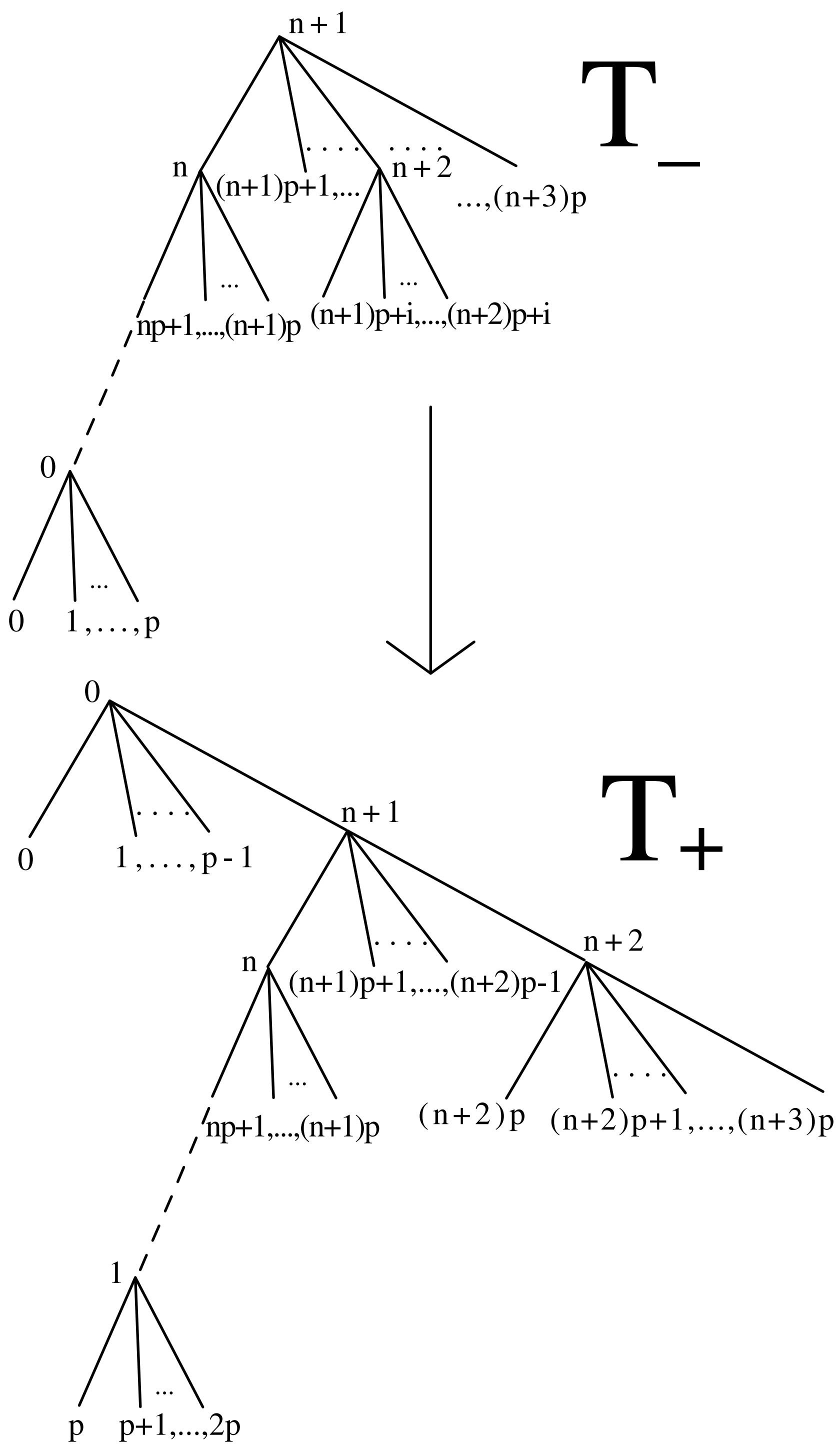


Figure 5.5: $rx_px_i^{-1}$ in $F(p+1)$

Now we look at the tree-pair diagram for $rx_p^2$, which we can see in Figure 5.6.

So we consider the length of $rx_p^2g$ where $g^{\pm1} \in \{x_0, x_1, \ldots, x_{p-1}, x_p\}$: $rx_p^2g$ does not satisfy the required conditions of Theorem 2.1.2 in the cases when $g = x_i, x_p$, so we know from Theorem 2.1.2 that $L(rx_p^2g) > 2n+2$ whenever $g = x_i, x_p$, thereby increasing the total length by at least one, and since we are multiplying by a generator, the length can increase by at most one. So $L(rx_p^2g) = 2n+3$ whenever $g = x_i, x_p$. Now all that remains is to check the length of $rx_p^2g$ when $g = x_0, x_0^{-1}, x_i^{-1}$.

1. $rx_p^2x_0$: Multiplying $rx_p^2$ by $x_0$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+1}$ in $T_-$ from type $\mathcal{L}_{\mathrm{L}}$ to type $\mathcal{R}_\emptyset$. This changes the caret pairing from ($\mathcal{L}_{\mathrm{L}}$,$\mathcal{R}_p$), which has weight 1, to ($\mathcal{R}_\emptyset$,$\mathcal{R}_p$), which has weight 2, thereby increasing the length by one. So $L(rx_p^2x_0) = 2n+3$.

2. $rx_p^2x_0^{-1}$: Multiplying $rx_p^2$ by $x_0^{-1}$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+2}$ in $T_-$ from type $\mathcal{R}_\emptyset$ to type $\mathcal{L}_{\mathrm{L}}$. This changes the caret pairing from ($\mathcal{R}_\emptyset$,$\mathcal{M}_\emptyset^p$), which has weight 1, to ($\mathcal{L}_{\mathrm{L}}$,$\mathcal{M}_\emptyset^p$), which has weight 2, thereby increasing the length by one. So $L(rx_p^2x_0^{-1}) = 2n+3$.

3. $rx_p^2x_i^{-1}$: Multiplying $rx_p^2$ by $x_i^{-1}$ satisfies the required conditions of Theorem 2.1.2 and changes $\wedge_{n+2}$ in $T_-$ from type $\mathcal{R}_\emptyset$ to type $\mathcal{M}_\emptyset^i$. This changes the caret pairing from ($\mathcal{R}_\emptyset$,$\mathcal{M}_\emptyset^p$), which has weight 1, to ($\mathcal{M}_\emptyset^i$,$\mathcal{M}_\emptyset^p$), which has weight 2, thereby increasing the length by one. So $L(rx_p^2x_i^{-1}) = 2n+3$.

Therefore any path from $l$ to $r$ which remains in $B_{2n+2}$ cannot pass through $rx_p^2$.

Because any path from $l$ to $r$ which remains in $B_{2n+2}$ cannot pass through $rx_pg$ whenever $g \in \{x_0^{-1}, x_1^{\pm1}, x_2^{\pm1}, ..., x_p^{\pm1}\}$, any path from $l$ to $r$ which remains in $B_{2n+2}$ must pass through $rx_px_0$. $\square$

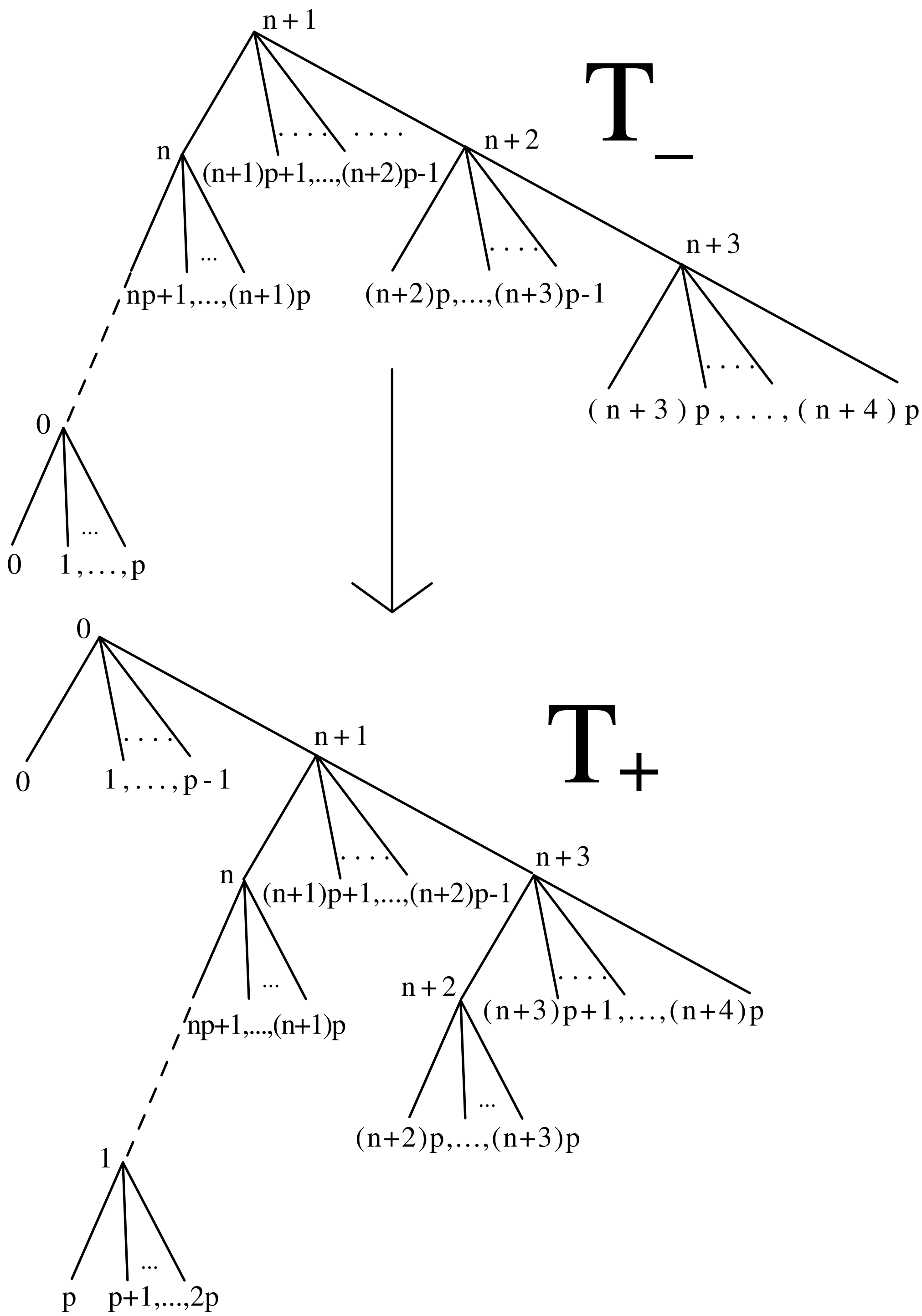


Figure 5.6: $rx_p^2$ in $F(p+1)$

### 5.2.3 Finding the vertex $h_l$

Before we seek candidates for the vertex $h_l$ on $p$, we begin with a few definitions that apply to tree-pair diagrams of elements of $F(p+1)$ that have been adapted from definitions given with respect to forest diagrams of F(2) in [1]:

**Definition (right foot) 5.2.4.** *Let $\wedge_s$ be the first right caret in the negative tree. If $\wedge_s$ has a nonempty left subtree, then we consider this subtree: within this new subtree, the right foot is the rightmost child of the last right caret in the subtree, which will be a leaf (if it were not a leaf, there would be another right caret in the tree). If $\wedge_s$ has an empty left subtree, then the right foot is the leftmost leaf of $\wedge_s$ in the negative tree. If the negative tree has no carets of type $\mathcal{R}$, then the right foot is the rightmost leaf of the root caret in the negative tree.*

**Definition (critical leaf) 5.2.5.** *Likewise, let $\wedge_t$ be the first right caret in the positive tree. If $\wedge_t$ has a nonempty left subtree, then we consider this subtree: within this new subtree, the critical leaf is the rightmost child of the last right caret in the subtree, which will be a leaf (if it were not a leaf, there would be another right caret in the tree). If $\wedge_t$ has an empty left subtree, then the critical leaf is the leftmost leaf of $\wedge_t$ in the positive tree. If the positive tree has no carets of type $\mathcal{R}$, then the critical leaf is the rightmost leaf of the root caret in the positive tree.*

It is important to note here that for the element $l$, the right foot is to the left of the critical leaf (index of right foot is $np$ and index of critical leaf is $(n+1)p$), and for the element $r$, the right foot is to the right of the critical leaf (index of right foot is $(n+3)p$ and index of critical leaf is $(n+1)p$). So it is clear that as we progress from vertex to vertex on the path from $l$ to $r$, the right foot must cross the critical leaf, i.e. the index of the right foot must change at some point from being less than that of the critical leaf to greater than that of the critical leaf.

Let $p$ denote a path from $l$ to $r$ which does not leave $B_{2n+2}$ and let $h_l$ be the first vertex along $p$ in which the index number of the right foot equals the index number of the critical leaf. We note that $h_r$ is the last vertex along $p$ in which the index number of the right foot equals the index number of the critical leaf. We want to show that $d(h_r, h_l) \geq 2n+3$, because by Lemma 5.2.2, this will show that $L(p) \geq 4n+4$.

For the following two lemmas, we will use $h_{l_1}$ to denote the first vertex along $p$ (in between $l$ and $h_l$) where the right foot has an index equal to or greater than the critical leaf (if the indexes are equal, then clearly $h_{l_1} = h_l$). We will use $h_{ll}$ to denote an arbitrary vertex on the path $p$ between $l$ and $h_{l_1}$ (including $l$ and $h_{l_1}$); this is a vertex on the path where the right foot has not yet crossed the critical leaf, although if $h_{ll} = h_{l_1}$, the right foot may have index equal to the critical leaf.

**Lemma 5.2.6.** *The last 3 carets in the positive and negative trees of $h_{ll}$ will remain the same as they were in $l$: the last 3 carets in the negative tree of $h_{ll}$ will be the same as the last 3 carets in the negative tree of $l$ (see Figure 5.2) and the last 3 carets of the positive tree will be a string of 3 carets of type $\mathcal{R}_\emptyset$. These are the carets which have an index number higher than the caret containing the critical leaf.*

*Proof.* We need to show that multiplying an arbitrary element $h_{ll}$ on the part of the path $p$ before the right foot has crossed the critical leaf will not modify the carets in the positive or negative trees of $h_{ll}$ with an index higher that that of the caret containing the critical leaf. Multiplication by any one of the generators will modify a tree in the tree-pair diagram as follows:

If the tree-pair diagram and generator satisfy the required conditions of Theorem 2.1.2, these changes will be made to the negative tree: multiplication by $x_0$ will change the caret type of the root, multiplication by $x_i$ will change the caret type of the $ith$ child of the root, multiplication by $x_p$ will change the caret type of the left child of the right child of the root, and multiplication by $x_0^{-1}$, $x_i^{-1}$, or $x_p^{-1}$ will change the caret type of the right child of the root. So the positive tree in the tree-pair diagram will remain completely unchanged, and as long as the index number of the right foot remains lower than or equal to the index number of the critical leaf, this will leave any carets with index number greater than the index number of the caret which contains the critical leaf unchanged in the negative tree.

If the tree-pair diagram and generator do not satisfy the required conditions of Theorem 2.1.2, it will be necessary to add carets before considering the changes described in the preceding paragraph that would follow from multiplication by a generator. For an arbitrary caret, which we'll denote $\wedge_v$, in either tree to have its type changed by the addition of carets in the tree, $\wedge_v$ must be type $\mathcal{R}$ or $\mathcal{M}^i$ and the added caret must have index number greater than $v$. But the only places that carets might need to be added in order for multiplication by a generator to take place are as follows (each of these added carets are described by their placement in the negative tree in the tree-pair diagram; a caret is then also added at the leaf with the same index in the positive tree of the tree-pair diagram, which will usually not be in the same location with respect to the root of the positive tree): multiplication by $x_0$ may require the addition of a caret as the left child of the root, multiplication by $x_i$ may require the addition of a caret as the $ith$ child of the root, multiplication by $x_p$ may require the addition of a caret as the left child of the right child of the root. Each of these carets will have index number less than or equal to the index number of the caret which contains the critical leaf. While the index number of the right foot remains less than or equal to the index number of the critical leaf, we will never need to add a caret when multiplying by $x_0^{-1}$, $x_i^{-1}$, or $x_p^{-1}$ because we will already have a caret as the right child of the root; the caret in the negative tree which is paired with the caret in the positive tree which immediately follows the caret containing the critical leaf is of type $\mathcal{R}_p$, so either it will be the right child of the root, or it will have some predecessor caret which will be the right child of the root. □

Because multiplication by a generator can change the index of the right foot by more than one, it is not obvious that $h_l$ is in fact a distinct vertex from $h_r$ ($h_r$ may be the only vertex on the path $p$ where the right foot and the critical leaf have the same index number); even if $h_l$ is a distinct vertex from $h_r$, we cannot necessarily assume that the index of the right foot has been less than the index of the critical leaf at all vertices along the path $p$ that come before $h_l$ (it is possible that the right foot has already crossed the critical leaf one or more times before we have reached a vertex in $p$ where the index of the right foot and the critical leaf are the same). However, if we look carefully, we can actually assert that $h_l$ is a distinct vertex from $h_r$ and that $h_l$ is the first vertex along the path $p$ at which the index of the right foot is not less than the index of the critical leaf.

**Lemma 5.2.7.** $h_l = h_{l_1}$ *i.e.* $h_l$ *is the first vertex on the path* $p$ *at which the right foot hits the critical leaf (by definition), and it is the first vertex on the path* $p$ *at which the index of the right foot is greater than or equal to the index of the critical leaf.*

*Proof.* For the duration of this proof, we will let $qp$ denote the index of the right foot, and $rp$ denote the index of the critical leaf for $h_{ll}$, which we recall is an arbitrary vertex on the path $p$ before the right foot has crossed the critical leaf; however, for the remainder of this proof, we will only consider $h_{ll}$ such that $h_{ll} \neq h_{l_1}$ so that $r > q$ (this assumption is necessary because we want to show that there is no vertex $h_{ll}$ on the path with $r > q$ such that the next vertex on the path is $h_{l_1} \neq h_l$, i.e. $h_{l_1}$ such that the index of the right foot is larger than the index of the critical leaf). We note that it is clear from the definitions of right foot and critical leaf that the index of the right foot and the index of the critical leaf must always be a multiple of $p$ apart. We need to explore when multiplying by a generator will change the relationship between the index number of the right foot and the index number of the critical leaf by enough for the right foot to cross the critical leaf as we progress along $p$ without ever hitting a vertex on $p$ where the right foot and the critical leaf have the same index.

First we consider the effect of adding a caret to the tree-pair diagram for $h_{ll}$, which will be required before we can multiply $h_{ll}$ by a generator in the case where $h_{ll}$ and the generator do not satisfy the required conditions of Theorem 2.1.2.

1. If we add a caret on a leaf with index less than or equal to $qp$, then the new index of the right foot will be $(q+1)p$ and the new index of the critical leaf will be $(r+1)p$. Because $(r+1)-(q+1)=r-q$ we can see that the right foot has gotten neither closer nor farther away from the critical leaf in the tree pair diagram .

2. If we add a caret on a leaf with index $i$ such that $qp < i \leq rp$, then the index of the right foot remains $qp$ and the index of the critical leaf becomes $(r+1)p$. Since $(r+1)-q=r-q+1$, the distance between the right foot and the critical leaf increases by $1p$, so the right foot cannot cross the critical leaf at this point along the path $p$.

3. If we add a caret on a leaf with index greater than $rp$, both the index of the right foot and the index of the critical leaf will remain unchanged.

Now we consider the movement of carets in the tree-pair diagram of $h_{ll}$ that can be induced by multiplication by a generator after any carets have been added as needed. For simplicity in this section, we will let $qp$ and $rp$ denote the index of the right foot and the critical leaf respectively after any carets have been added as needed so that multiplication by the generator can be done.

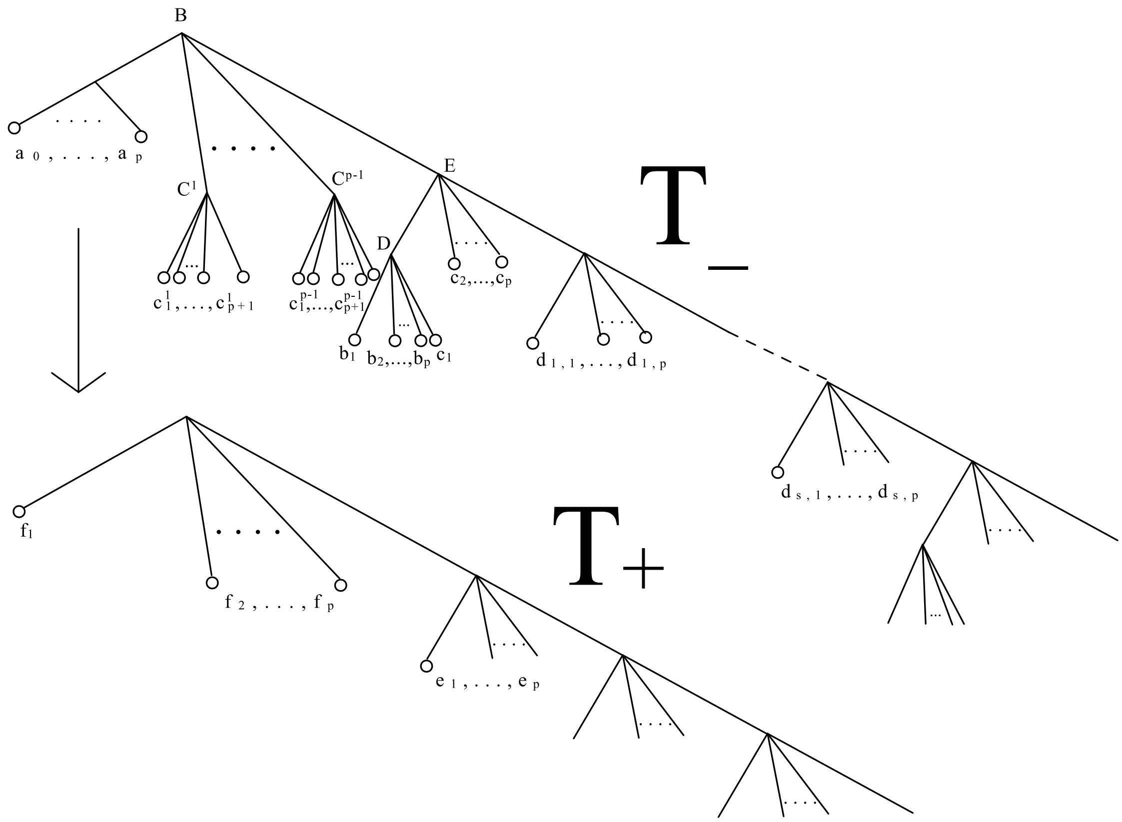


Figure 5.7: $h_{ll}$ in $F(p+1)$ after carets have been added as needed so that $T^-$ satisfies the required conditions of Theorem 2.1.2

To proceed, we will consider Figure 5.7, which is the tree-pair diagram of $h_{ll}$ once any carets have been added as needed to satisfy the required conditions of Theorem 2.1.2. We note that by Lemma 5.2.6, the index of the rightmost leaf of subtree $d_{s,1}$ and the index of the rightmost leaf of subtree $e_1$ must be the same. We will use the convention that the subtree $d_{0,1}$ is the subtree consisting of $\wedge_D$ and all its subtrees and that similarly, the subtree $d_{0,i}$ is the subtree $c_i$ for $i = 2, ..., p$. Since multiplication by a generator will only change the structure of the negative tree, when multiplying by each generator, we know that the index of the critical leaf will remain $rp$, and only the index of the right foot will change with the multiplication. We also note that in Figure 5.7, the right foot will be the rightmost leaf in the subtree labeled $c_1$.

Now we consider the effect of multiplying by the generators:

1. Multiplication by $x_0$:

   The right foot will become the rightmost leaf in subtree $a_p$, which clearly has index less than $qp$, so the right foot will not cross the critical leaf during multiplication by $x_0$.

2. Multiplication by $x_0^{-1}$:

   The right foot will become the rightmost leaf in subtree $d_{1,1}$; if $s = 1$ then the index of $d_{1,1}$ will equal the index of $e_1$, and thus the index of the right foot will equal the index of the critical leaf. If, however, $s > 1$, then the index of the right foot will remain lower than the index of the critical leaf, and the right foot will not cross the critical leaf during multiplication by $x_0^{-1}$.

3. Multiplication by $x_i$:

   The right foot will become the rightmost leaf in subtree $c^i_{p-i+1}$, which clearly has index less than $qp$, so the right foot will not cross the critical leaf during multiplication by $x_i$.

4. Multiplication by $x_i^{-1}$:

   The right foot will become the rightmost leaf in subtree $d_1$; in this case the same argument follows as in multiplication by $x_0^{-1}$, and the right foot will not cross the critical leaf during multiplication by $x_i^{-1}$ (although the index of the right foot might equal the index of the critical leaf after multiplication by $x_i^{-1}$).

5. Multiplication by $x_p$:

   The right foot will become the rightmost leaf in subtree $b_1$, which clearly has index less than $qp$, so the right foot will not cross the critical leaf during multiplication by $x_p$.

6. Multiplication by $x_p^{-1}$:

   The right foot will become the rightmost leaf of subtree $d_{1,1}$, and the right foot will not cross the critical leaf during multiplication by $x_p^{-1}$ (although the index of the right foot might equal the index of the critical leaf after multiplication by $x_p^{-1}$).

□

Now we need to establish a few more definitions and facts that will be needed to complete the proof:

**Definition (left-sided) 5.2.8.** *An element $f$ of $F(p+1)$ is left-sided if:*

1. *The index number of the right foot is the same as the index number of the critical leaf (i.e. $t = s$), and*

2. *All right carets in the tree-pair diagram are of type $\mathcal{R}_\emptyset$. (We note that part one of the definition then implies that in a reduced tree-pair diagram for a left-sided element, there is either one caret of type $\mathcal{R}_\emptyset$ in each tree, or no right carets in either tree.)*

**Remark 5.2.9.** *$h_r$ is left-sided.*

Throughout the rest of this section, we let $N(f)$ denote the number of carets in either the positive or negative tree of the minimal tree-pair diagram of $f$ in

$F(p+1)$ (since the positive and the negative tree have the same number of carets, this number will be the same for either tree in the tree pair).

**Lemma 5.2.10.** *For any left-sided element $f$ of $F(p+1)$, $L(f) \geq 2(N(f)-2)$.*

*Proof.* We consider all the possible types of caret pairings in the tree-pair diagram in a left-sided element: (see Figure 5.9)

1. The first caret pair must have type pair ($\mathcal{L}_\emptyset$,$\mathcal{L}_\emptyset$), which has weight 0.

2. The only caret pair which contains right carets may have type pair ($\mathcal{R}_\emptyset$,$\mathcal{R}_\emptyset$) if it exists, which has weight 0, or this caret pair may not exist if there are no right carets in the tree-pair diagram, so the weight of this caret pair is greater than or equal to 0.

3. All other caret pairs in the tree-pair diagram must be one of the following types: (note that the order in which the types appear in the caret pair does not change its weight, i.e. $w(\tau_1,\tau_2) = w(\tau_2,\tau_1)$ where $\tau_1$ and $\tau_2$ are caret types)
   ($\mathcal{L}_{\mathrm{L}}$,$\mathcal{L}_{\mathrm{L}}$) which has weight 2
   ($\mathcal{M}_\emptyset^i$, $\mathcal{L}_{\mathrm{L}}$) which has weight 2
   ($\mathcal{M}_\emptyset^i$, $\mathcal{M}_\emptyset^j$) which has weight 2
   ($\mathcal{M}_\emptyset^i$, $\mathcal{M}_l^k$) which has weight 2 if $j \leq k$ and weight 4 if $j > k$
   ($\mathcal{M}_k^i$, $\mathcal{L}_{\mathrm{L}}$) which has weight 2
   ($\mathcal{M}_k^i$, $\mathcal{M}_\emptyset^j$) which has weight 2 if $l > i$ and weight 4 if $l \leq i$
   ($\mathcal{M}_k^i$, $\mathcal{M}_l^k$) which has weight 2

So the weight of each caret pair in the tree-pair diagram is greater than or equal to 2, with the exception of the first and last carets.

Supposing that $N(f) = n$, adding all of these together gives us the following length of the element $f$:

$L(f) \geq 2(n-2)$ □

**Definition 5.2.11.** *We define $y = x_0^{-1} x_p x_0$. We can see the tree-pair diagram for $y$ in Figure 5.8.*

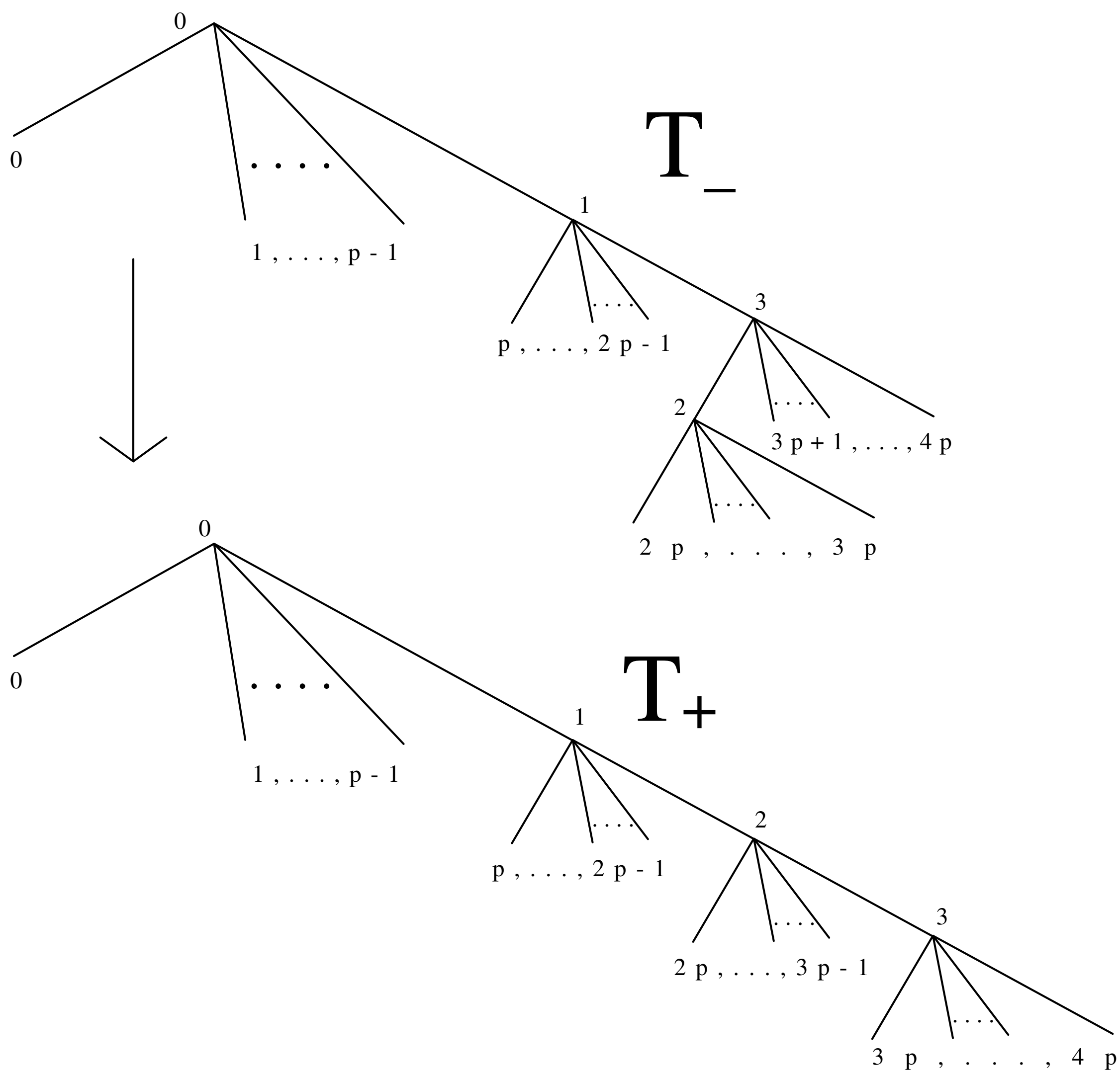


Figure 5.8: $y$ in $F(p+1)$

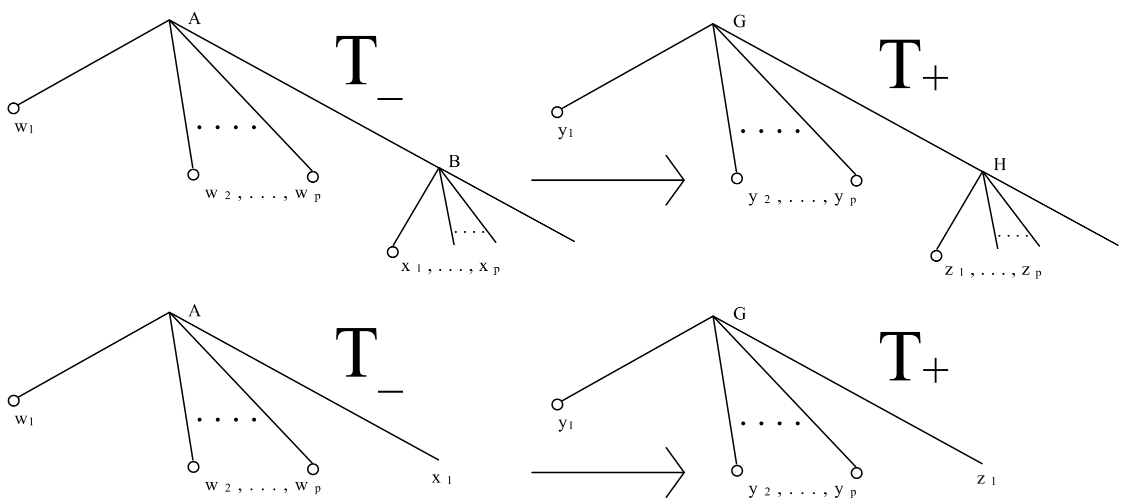


Figure 5.9: General form of left-sided elements in $F(p+1)$ (a left-sided element $f$ either has one right caret in each tree (top) or no right carets (bottom))

All left-sided elements can be depicted by the tree-pair diagram in Figure 5.9 where we note the that right foot is the rightmost leaf in the subtree $x_1$ and that the critical leaf is the rightmost leaf in the subtree $z_1$ (if $x_1$ ($z_1$) is empty, then the right foot (critical leaf) is $x_1$ ($z_1$)). We also note here that a left-sided element will only have right carets if the subtree $x_1$ or $z_1$ is not empty; if both of these subtrees are empty, then the right carets will cancel each other and the reduced tree-pair diagram will not contain right carets.

**Remark 5.2.12.** *Left-sided elements commute with y.*

**Lemma 5.2.13.** *For any left-sided element $f$ in $F(p+1)$, $L(fy) = L(f) + 3$.*

*Proof.* We can see by considering the tree-pair diagram of $fy$ (for $f$ a left-sided element) in Figure 5.10 that $fy$ will have exactly the same caret pairings as $f$ did for any carets in any of the subtrees labeled $w_i, x_i, y_i, z_i$ for $i = 1, ...p$. Likewise,

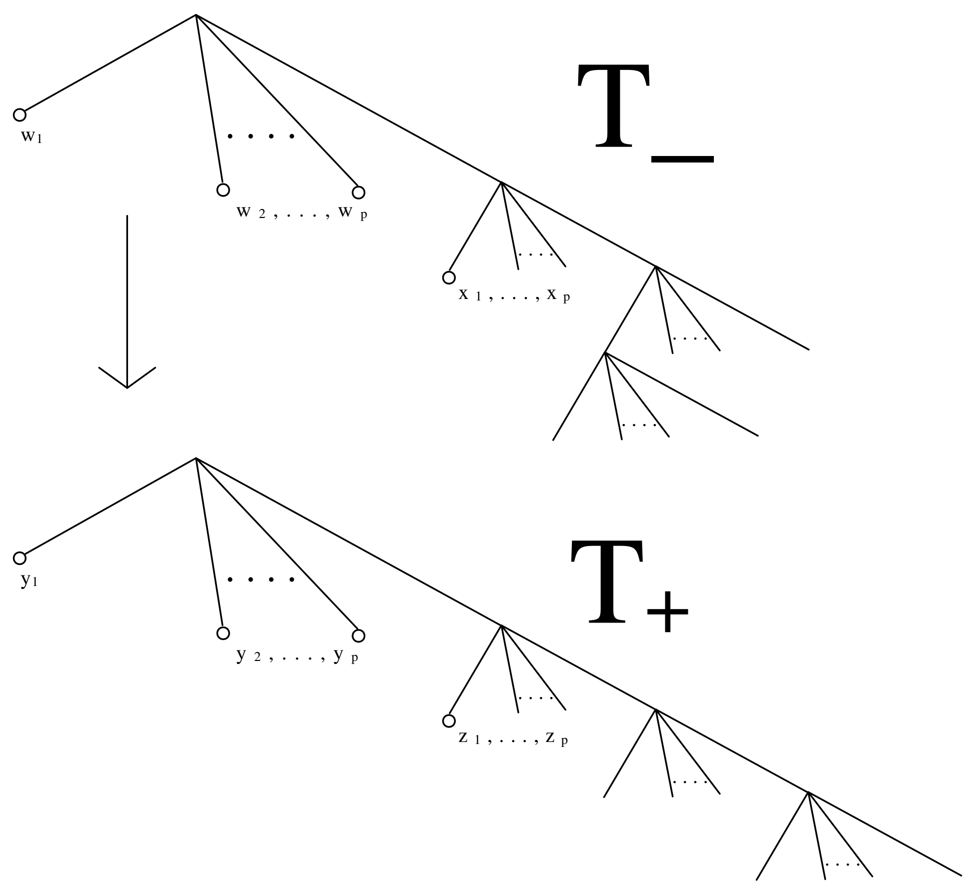


Figure 5.10: $yf = fy$ for left-sided element $f$ in $F(p+1)$

the carets $\wedge_A$ and $\wedge_G$ will have the same type and pairing in $fy$ as they did in $f$. The only changes in caret type from $f$ to $fy$ are to the carets with the indexes $B, C, D, H, I, J$. We note that $B = H$, $C = I$, and $D = J$, so we will have to consider the weight of the caret pairs $(\wedge_B, \wedge_H)$, $(\wedge_C, \wedge_I)$, and $(\wedge_D, \wedge_J)$ and look at how the weight of each of these pairs changes from $f$ to $fy$. If present in $f$, $(\wedge_B, \wedge_H)$ produces the type pair $(\mathcal{R}_\emptyset,\mathcal{R}_\emptyset)$ which has weight 0, but in $fy$ the type of $\wedge_B$ is changed from $\mathcal{R}_\emptyset$ to $\mathcal{R}_p$, producing the type pair $(\mathcal{R}_p,\mathcal{R}_\emptyset)$ which has weight 2. (If $(\wedge_B, \wedge_H)$ was not present in $f$, it will still be present in $fy$ with the type pairing $(\mathcal{R}_p,\mathcal{R}_\emptyset)$.) The caret pair $(\wedge_C, \wedge_I)$ is not present in $f$, but it is present in $fy$ where it has type pair $(\mathcal{M}_\emptyset^p,\mathcal{R}_\emptyset)$, which has weight 1. The caret pair $(\wedge_D, \wedge_J)$ is also not present in $f$ but present in $fy$ where it has the type pair $(\mathcal{R}_\emptyset,\mathcal{R}_\emptyset)$, which has weight 0. These are the only difference between the length of $f$ and $fy$, so $L(fy) = L(f) + 3$.

□

**Lemma 5.2.14.** *There exists a left-sided element $h_l'$ on the path $p$ from $l$ to $r$ such that $h_l = yh_l'$.*

*Proof.* To simplify things, we begin by rewriting the tree-pair diagram for $h_{ll}$ which originally appeared in Figure 5.7 by denoting the caret with subtrees labeled $b_1, b_2, ..., b_p, c_1$ by a single subtree which we will label $d_{0,1}$. We also denote each subtree with root caret labeled $C^i$ for some $i = 1, ..., p-1$ by a single subtree labeled $a_i$. We also denote the subtree consisting of the caret with subtrees $a_0, ..., a_p$ by a single subtree which we call $a_0$. This yields the depiction of the tree-pair diagram of an arbitrary $h_{ll}$ as seen in Figure 5.11. We can suppose that this new simpler depiction of arbitrary $h_{ll}$ is reduced because none of the visible carets which make up the basic structure will cancel.

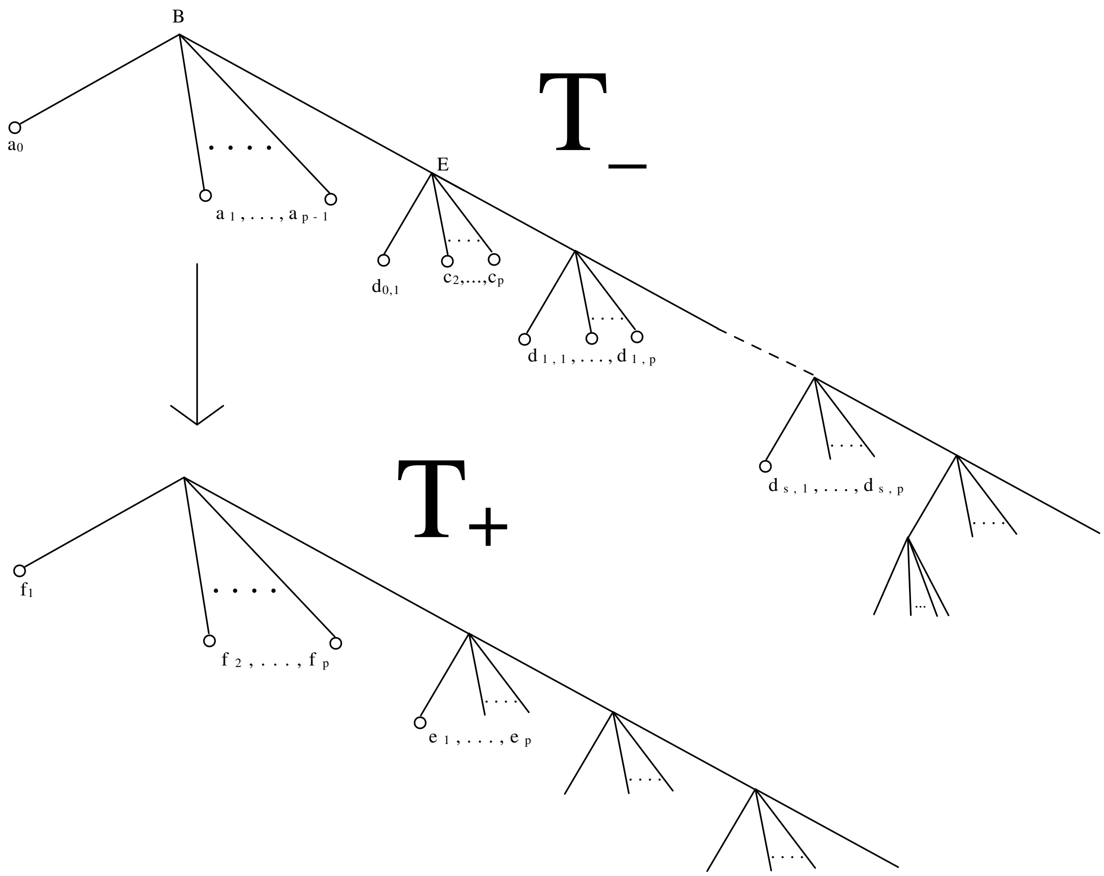


Figure 5.11: Rewritten version (see proof of Lemma 5.2.14) of $h_{ll}$ in $F(p+1)$ after carets have been added as needed so that $T^{-}$ satisfies the required conditions of Theorem 2.1.2

We see that if $s = 0$, Figure 5.11 becomes a tree-pair diagram for $h_l$ (see proof of Lemma 5.2.14). By definition, any left-sided element $f$ in $F(p+1)$ will have the form given by the tree-pair diagram in Figure 5.9. If we compute $yf$, we can then see that the tree-pair diagram of $yf$ is the same as the tree-pair diagram of $h_l$ if $f$ is the left-sided element depicted in Figure 5.9 such that $w_i = a_i$, $x_1 = d_{0,1}$, $y_i = f_i$, and $z_1 = e_1$ for $i = 0, 1, 2, ..., p$. We denote this left-sided element by $h_l'$. □

We also note that whenever $h_l$ (as depicted in Figure 5.11 when $s = 0$) is reduced, the appropriate tree-pair diagram for $h_l'$ given in Figure 5.9 when $w_i = a_i$, $x_1 = d_{0,1}$, $y_i = f_i$ and $z_1 = e_1$ for $i = 0, 1, 2, ...p$ will also be reduced. (When $d_{0,1}$ and $e_1$ are both leaves in $h_l$, this implies that $h_l'$ will be represented by the reduced tree-pair diagram in Figure 5.9 with no right carets; conversely, if $d_{0,1}$ or $e_1$ is not a leaf in $h_l$, then $h_l'$ will be represented by the reduced tree-pair diagram in Figure 5.9 with one right caret in each tree.)

**Lemma 5.2.15.** *The tree-pair diagram of $h_r^{-1}h_l'$ is already reduced.*

*Proof.* Computing the product $h_r^{-1}h_l'$ yields the tree-pair diagram shown in Figure 5.12.

The only way in which a tree-pair diagram can be reduced is by finding a pair of exposed carets with the same index number. If we can consider all possible exposed carets in the negative tree of $h_r^{-1}h_l'$ and show that none of them can possibly pair up with an exposed caret in the positive tree, then this will be sufficient to show that the diagram is reduced. So we begin by considering all possible exposed carets in the negative tree:

Case 1: $\wedge_E$ or $\wedge_{D_i}$ for $i = 1, ..., p$ in the negative tree or a caret present in one of the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$ is exposed.

Here $E = G_n$, so we consider whether or not both $\wedge_E$ and $\wedge_{G_n}$ could be exposed. Because right carets are present in the reduced tree-pair diagram for $h'_l$, we know that the subtree $d_{0,1}$ or $e_1$ is nonempty. Any carets present in the subtree $d_{0,1}$ in $h'_l$ will also be present in the subtree $d'_{0,1}$ in $h_r^{-1}h'_l$ and any carets present in the subtree $e_1$ in $h'_l$ will also be present in the subtree $e_{1(n,p)}$ in $h_r^{-1}h'_l$. So we know that the subtree $d'_{0,1}$ or $e_{1(n,p)}$ is nonempty and therefore that the caret pair $(\wedge_E, \wedge_{G_n})$ will not cancel.

(a) Suppose the reduced tree-pair diagram for $h'_l$ contained a right caret in each tree.

i. $\wedge_{D_i}$ for $i = 1, ..., p$ in the negative tree:

These carets are present in the negative tree of $h_r^{-1}h'_l$ because they are necessary in order for the positive tree of $h'_l$ to be identical to the negative tree of $h_r^{-1}$ so that the product $h_r^{-1}h'_l$ can be calculated. Each of these may have already been present in the negative tree of $h'_l$ or may have had to be added to the negative tree of $h'_l$ so that the multiplication with $h_r^{-1}$ can be performed. In order for the positive tree of $h'_l$ to be identical to the negative tree of $h_r^{-1}$ so that the multiplication can be performed, there must be a string of $n$ carets of type $\mathcal{M}_\emptyset^p$ hanging off the leftmost leaf of the right child of the root in the positive tree of $h'_l$ so that the left side of each caret in the string lines up with the left side of all other carets in the string. Let us call this string of carets in the positive tree of $h'_l$ and the negative tree of $h_r^{-1}$ $\wedge_{E_1}, ..., \wedge_{E_n}$. Then $\wedge_{D_1}$ is the caret in the negative tree of $h'_l$ that has the same index number as the leftmost caret in this string in the positive tree of $h'_l$ (we denote this by $\wedge_{E_1}$), $\wedge_{D_2}$ is the caret in the negative tree of $h'_l$ that has the same index as the second leftmost caret in this string in the positive tree of $h_l'$ (we denote this by $\wedge_{E_2}$), etc., so that $\wedge_{D_n}$ is the caret in the negative tree of $h_l'$ that has the same index number as the rightmost caret in this string in the positive tree of $h_l'$ (we denote this by $\wedge_{E_n}$). Likewise, the carets $\wedge_{E_1}, ..., \wedge_{E_n}$ appear in the negative tree of $h_r^{-1}$ (which must be identical to the positive tree of $h_l'$ in order for multiplication to be performed) and they will have the same index numbers in the final product as the carets in the positive tree of $h_r^{-1}$ which originally (before any carets were added so that multiplication could be carried out) had index numbers $1, ..., n$ in the positive tree of $h_r^{-1}$. These carets in the positive tree of $h_r^{-1}$ are exactly those carets in $h_r^{-1}h_l'$ labeled $E_0, ..., E_n$.

If $\wedge_{D_i}$ for some $i \in \{1, ..., n\}$ is exposed in $h_r^{-1}h_l'$, it cannot be paired with an exposed caret in the positive tree of $h_r^{-1}h_l'$ because it will be paired with $\wedge_{G_{i-1}}$, which is clearly not an exposed caret for any $i$. So the caret pair $(\wedge_{D_i}, \wedge_{G_{i-1}})$ will never cancel.

ii. a caret present in one of the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$:

If a given $\wedge_{E_i}$ $i \in \{1, ..., n\}$ in the positive tree of $h_l'$ was already present in that tree before any carets had to be added so that multiplication could be carried out, then it might itself have contained subtrees. Any subtree hanging off the $jth$ child (using the convention that the child numbering begins with 0) of $\wedge_{E_i}$ we will denote by $e_{i,j}'$; then we will have $e_{i,j}' \sim d_{0,1(i-1,j)}$ where $a \sim b$ is defined such that the subtree $a$ and the subtree $b$ have the same number of carets and these carets have the same index numbers (although the subtrees themselves will generally not be identical). When $a \sim b$, we will say that the subtree $a$ *corresponds* to the subtree $b$.

The subtree $e'_{i,j}$ must be added to the negative tree of $h_r^{-1}$ in exactly the same place it appeared in the positive tree of $h'_l$ so that multiplication can be performed, and so the subtree $e'_{i,j}$ must also be added to the positive tree of $h_r^{-1}$ at the leaf which was originally (before any carets were added so that multiplication could be carried out) numbered $(i-1)p+j$. This is exactly the subtree labeled $e_{1(i,j)}$ in the positive tree of $h_r^{-1}h'_l$. Therefore we will have $d_{0,1(0,0)} \sim e_{1(1,0)}, ..., d_{0,1(n-1,p)} \sim e_{1(n,p)}$ where $e_{1(k,l)}$ ($k \in \{1, ..., n\}$ and $l \in \{0, ..., p\}$) is identical to the subtree $d_{0,1(k-1,l)}$ in the negative tree of $h'_l$.

So if there exists an exposed caret in the subtree $d_{0,1(k-1,l)}$ in the negative tree of $h_r^{-1}h'_l$ then that same caret in the subtree $d_{0,1(k-1,l)}$ in the negative tree of $h'_l$ was exposed. The subtree $e_{1(k,l)}$ in the positive tree of $h_r^{-1}h'_l$ is identical to the subtree $e'_{k,l}$ in the positive tree of $h'_l$, and for any caret in $e_{1(k,l)}$ in $h_r^{-1}h'_l$ paired with a caret in $d_{0,1(k-1,l)}$ in $h_r^{-1}h'_l$, that same caret in $e'_{k,l}$ in $h'_l$ must be paired with the same caret in $d_{0,1(k-1,l)}$ in $h'_l$. If we suppose then that the caret in $e_{1(k,l)}$ paired with the exposed caret in $d_{0,1(k-1,l)}$ in the positive tree of $h_r^{-1}h'_l$ were also exposed, then the caret in $e'_{k,l}$ paired with the exposed caret in $d_{0,1(k-1,l)}$ in $h'_l$ must also be exposed, which would imply that $h'_l$ is not reduced. Since we can assume that $h'_l$ is in reduced form, we can conclude that none of the carets in the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$ in the negative tree of $h_r^{-1}h'_l$ will cancel.

(b) Suppose the reduced tree-pair diagram for $h'_l$ contained no right carets. then the subtree $d'_{0,1}$ will be exactly a string of the $\wedge_{D_1}, ..., \wedge_{D_n}$ carets such that each $\wedge_{D_i}$ is of type $\mathcal{M}_\emptyset^p$ and such that the left edge of each caret in the string lines up with the left edge of all other carets in the string. Also, in this case the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}, e_{1(0,1)}, ..., e_{1(n,p)}$ are all empty. So the only possible exposed caret in the negative tree of $h_r^{-1}h_l'$ is $\wedge_{D_1}$, but $\wedge_{D_1}$ will be paired with $\wedge_{G_0}$ in the positive tree, which is clearly not exposed. So the caret pair $(\wedge_{D_i}, \wedge_{G_{i-1}})$ will never cancel, and there will be no carets present in the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$ to cancel.

Case 2: A caret in the subtrees $a_0', ..., a_{p-1}', c_1$ in $h_r^{-1}h_l'$ (other than $\wedge_{D_1}, ..., \wedge_{D_n}$ or a caret present in $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$) is exposed.

Note that the subtree $a_i$ for $i = 0, ..., p-1$ in $h_r^{-1}h_l'$ is equal to the subtree $a_{i-1}$ in $h_l'$ and $c_1$ in $h_r^{-1}h_l'$ is equal to the subtree $d_{0,1}$ in $h_l'$ once any of the $\wedge_{D_1}, ..., \wedge_{D_n}$ have been added to leaves in $h_l'$ as needed for the multiplication to take place. So, for a given caret in the subtree $a_i$ or the subtree $c_1$ in $h_r^{-1}h_l'$ that is not one of the $\wedge_{D_1}, ..., \wedge_{D_n}$ or present in one of the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$, there is a corresponding caret in one of the subtrees $f_1, ..., f_p$ in the positive tree of $h_l'$. The negative tree of $h_r^{-1}$ does not contain any of these carets initially (before carets have been added so that multiplication can take place). So any of these carets which are present in the positive tree of $h_l'$ must be added in exactly the same place in the negative tree of $h_r^{-1}$.

So the subtrees $f_1, ..., f_p$ in the positive tree of $h_l'$ will be present in exactly the same places in the negative tree of $h_r^{-1}$. Then the subtree $f_i$ ($i \in \{1, ..., p\}$) must be added to the leaf originally (before any carets were added so that multiplication could be carried out) numbered $i-1$ in the positive tree of $h_r^{-1}$

so that the subtrees $f_1, ..., f_p$ in the positive tree of $h_r^{-1}h_l'$ will be identical to the subtrees $f_1, ..., f_p$ found in the positive tree of $h_l'$.

Suppose that an exposed caret is present in the subtree $a_i$ for some $i \in \{0, ..., p-1\}$ or in the subtree $c_1$ in $h_r^{-1}h_l'$ that is not one of the $\wedge_{D_1}, ..., \wedge_{D_n}$ or present in the subtrees $d_{0,1(0,0)}, ..., d_{0,1(n-1,p)}$. Then that same caret will also have been exposed as a member of one of the subtrees $a_1, ..., a_p, d_{0,1}$ in $h_l'$. This exposed caret in the negative tree of $h_r^{-1}h_l'$ will be paired with a caret in one of the subtrees $f_1, ..., f_p$ in $h_r^{-1}h_l'$; if this caret it is paired with in the positive tree is also exposed, then the same caret in its corresponding subtree $f_j$ in the positive treee of $h_l'$ will also have been exposed. Since we can assume that $h_l'$ was already in reduced form, we can conclude that none of the carets in the subtrees $a_0, ..., a_{p-1}, c_1$ in the negative tree of $h_r^{-1}h_l'$ will cancel.

□

**Corollary 5.2.16.** $L(h_r^{-1}h_l') \geq 2n$

*Proof.* By looking at Figure 5.12, we can see clearly that the number of carets in the positive tree is greater than or equal to $n+2$, because simply counting the labeled carets $\wedge_F, \wedge_{G_0}, ..., \wedge_{G_n}$ yields $n+2$ carets. We recall that by Lemma 5.2.15, $h_r^{-1}h_l'$ in Figure 5.12 is already reduced, so we know that $w(h_r^{-1}h_l') \geq n$.

By looking at Figure 5.12, we can also see that $h_r^{-1}h_l'$ is left sided, so by Lemma 5.2.10, $L(h_r^{-1}h_l') \geq 2(N(h_r^{-1}h_l') - 2)$. Putting this together with $N(h_r^{-1}h_l') - 2 \geq n$ yields $L(h_r^{-1}h_l') \geq 2n$. □

**Remark 5.2.17.** *$h_r^{-1}h_l'$ is left-sided.*

**Theorem 5.2.18.** *$F(p+1)$ is not minimally almost convex.*

*Proof of part 3 of Theorem 5.2.1.*

$$d(h_l, h_r) = L(h_r^{-1}h_l) = L(h_r^{-1}h_l'y) = L(h_r^{-1}h_l') + 3 \geq 2n + 3$$

So $d(h_r, h_l) \geq 2n + 3$, and by Lemma 5.2.2 $L(p) \geq 4n + 4$. □

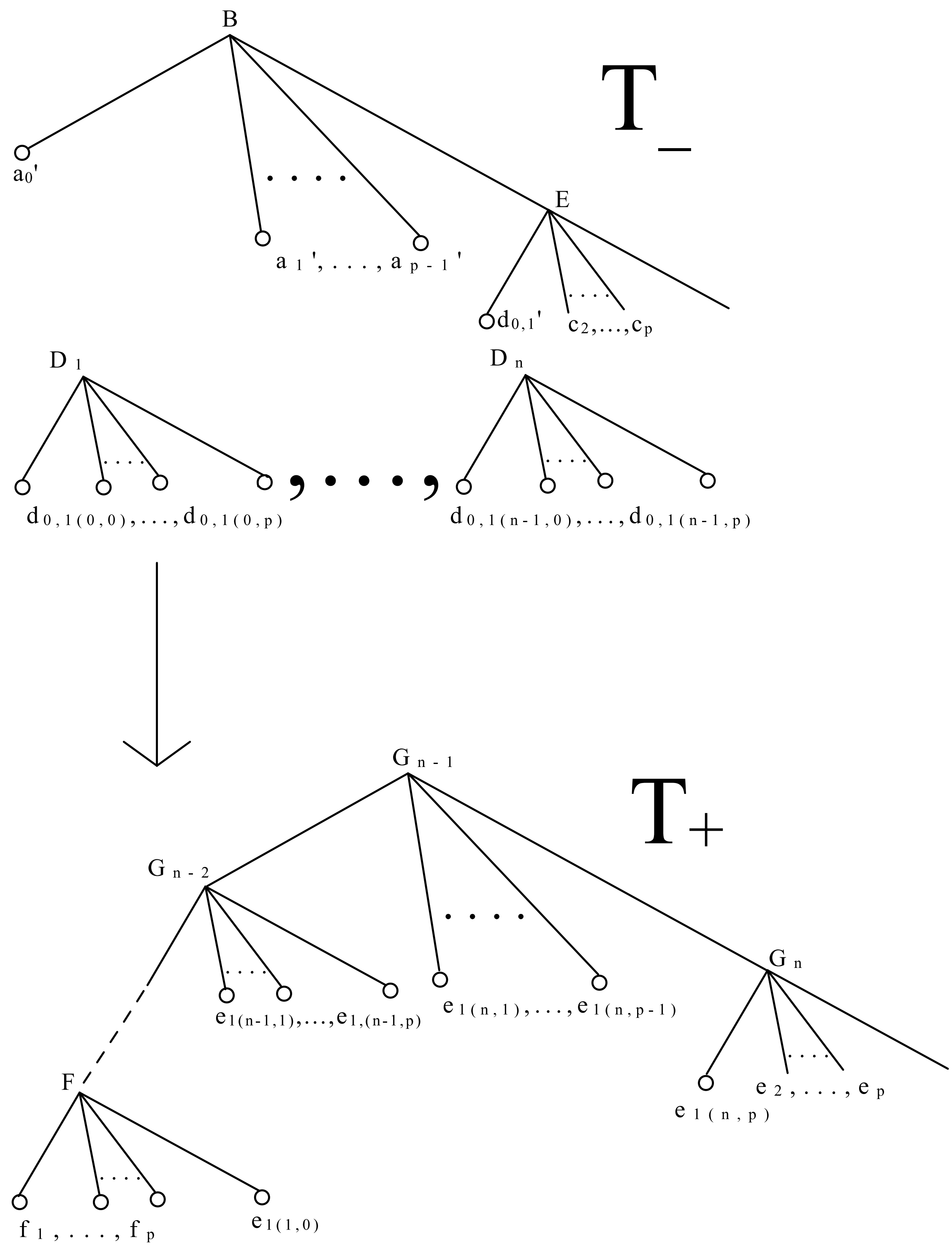


Figure 5.12: $h_r^{-1}h_l'$ in $F(p+1)$ ($n$ carets labeled $D_1, ..., D_n$ are present somewhere in the negative tree such that their leaves have index less than $c_2$)

# Part II

# Metric Properties of Thompson's Group $F(n, m)$

# Chapter 6

# Introduction to Thompson's Group $F(n, m)$

Just as in Part I, for notational simplicity we use the convention that $n = p + 1$ where $p \in \mathbb{N}$, and we add the analogous convention that $m = q + 1$ where $q \in \mathbb{N}$.

After Richard Thompson discovered Thompson's group $F(2)$ and G. Higman generalized this group to a family of groups $F(p+1)$, Melanie Stein in [17] studied an even more general family of groups, which we will consider in this section. While for the sake of simplicity we refer to these groups throughout this paper as Thompson groups, it would perhaps be more accurate to describe them as Thompson-Higman-Stein groups. We can think of these groups as a generalization of the groups $F(p + 1)$.

**Definition (Thompson's group $F(p+1, q+1)$) 6.0.1.** *The group $F(p+1, q+1)$, where $p, q \in \mathbb{N}$ is the group of piecewise-linear orientation-preserving homeomorphisms of the unit interval with finitely-many breakpoints in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$ and slopes in $\langle p + 1, q + 1\rangle$.*

We could similarly define $F(p_1 + 1, p_2 + 2, ..., p_l + l)$, however, we will restrict our discussion here to the group $F(p + 1, q + 1)$. In [17] Stein developed infinite and finite presentations for these groups which we will introduce in a later section.

Throughout this paper, when we refer to the group $F(p+1, q+1)$, our general assumption is that $p+1$ and $q+1$ are relatively prime. We make this assumption because when we have $F(p + 1, q + 1)$ such that $p + 1$ and $q + 1$ are not relatively prime, the resulting group is isomorphic to a group of the form $F(p_1 + 1, ..., p_l + 1)$

where $l = 1, 2$, or 3, and where the $p_i + 1$ are pairwise relatively prime for $i = 1, ..., l$. To see this, we suppose that we have a group $F(p+1, q+1)$ such that $r|p+1$ and $r|q+1$. Then we can rewrite $F(p+1, q+1) = F(rm_1, rm_2)$ for some $m_1, m_2 \in \mathbb{N}$ where $r$ is the greatest common divisor of $p+1$ and $q+1$ and therefore $m_1$ and $m_2$ are relatively prime. If $m_1 = 1$ and $m_2 = r^t$ for some $t \in \mathbb{N}$, then

$$\mathbb{Z}\left[\frac{1}{(p+1)(q+1)}\right] = \mathbb{Z}\left[\frac{1}{(r)(r^{t+1})}\right] = \mathbb{Z}\left[\frac{1}{r}\right]$$

and

$$\langle p+1, q+1\rangle = \langle r, r^{t+1}\rangle = \langle r\rangle$$

so

$$F(p+1, q+1) = F(rm_1, rm_2) \simeq F(r)$$

If $m_1 = 1$ and $m_2 \neq r^t$ for some $t \in \mathbb{N}$, then

$$\mathbb{Z}\left[\frac{1}{(p+1)(q+1)}\right] = \mathbb{Z}\left[\frac{1}{(r)(rm_2)}\right] = \mathbb{Z}\left[\frac{1}{rm_2}\right]$$

and

$$\langle p+1, q+1\rangle = \langle r, rm_2\rangle = \langle r, m_2\rangle$$

so

$$F(p+1, q+1) = F(rm_1, rm_2) \simeq F(r, m_2)$$

where $r$ and $m_2$ are relatively prime. If $m_1 \neq 1$ and $m_2 \neq 1$, then

$$\mathbb{Z}\left[\frac{1}{(p+1)(q+1)}\right] = \mathbb{Z}\left[\frac{1}{(rm_1)(rm_2)}\right] = \mathbb{Z}\left[\frac{1}{rm_1m_2}\right]$$

and

$$\langle p+1, q+1\rangle = \langle rm_1, rm_2\rangle = \langle r, m_1, m_2\rangle$$

so

$$F(p+1, q+1) = F(rm_1, rm_2) \simeq F(r, m_1, m_2)$$

where $r, m_1$ and $m_2$ are all pairwise relatively prime.

We now want to represent the elements of the group $F(p+1, q+1)$ as tree-pair diagrams, just as we did for the group $F(p+1)$. We begin by defining a $(p+1, q+1)$–ary tree. A tree which consists of both $(p+1)$–ary and $(q+1)$–ary carets will be referred to as a *$(p+1, q+1)$–ary tree*. Any pair of $(p+1, q+1)$–ary trees with the same number of leaves will then be a *$(p+1, q+1)$–ary tree-pair diagram*. Every $(p+1, q+1)$–ary tree-pair diagram clearly represents an element of $F(p+1, q+1)$. Every element of $F(p+1, q+1)$ can be represented by an infinite number of equivalent tree-pair diagrams. Clearly if an element of $F(p+1, q+1)$ can be represented by at least one tree-pair diagram, then it can be represented by an infinite number of equivalent ones, because to get a new tree-pair diagram from the old, all we need do is add a $(p+1)$–ary or $(q+1)$–ary caret to the leaf with the same index number in both trees in the diagram. To see that any element of $F(p+1, q+1)$ can be represented by at least one tree-pair diagram, we note that performing the algorithm given in Theorem 7.2.2 followed by addition on the chart (see Definition 7.1.24) will always yield a tree-pair diagram for any given element of $F(2, 3)$.

# Chapter 7

## Representing Elements of $F(p+1, q+1)$ Using Tree-Pair Diagrams

One of the differences between Thompson's groups $F(p+1)$ and $F(p+1, q+1)$ is apparent when constructing the tree-pair diagrams for each. In $F(p+1)$, any two equivalent trees will be identical, and therefore any two equivalent tree-pair diagrams will either be identical, or one of them will have pairs of exposed carets which can be reduced, and once all reducible pairs of exposed carets have been removed, the resulting tree-pair diagram will be identical to the minimal tree-pair diagram for that element. In $F(p+1, q+1)$, equivalent trees will not necessarily be identical, and so we may not be able to identify pairs of equivalent tree-pair diagrams simply by adding or removing caret pairs, and we may need to do more than simply remove exposed caret pairs in order to obtain a minimal tree-pair diagram. For example, the elements of $F(2, 3)$ depicted in Figure 7.1 and Figure 7.2 are both the identity element, yet many of the pairs of carets in either tree-pair diagram cannot be canceled unless we first perform other substitutions within the trees.

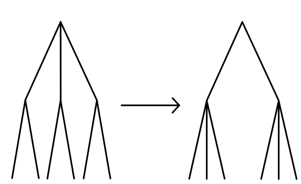

Figure 7.1: One tree-pair diagram for the identity element in $F(2, 3)$

We will observe that this is generally true; two equivalent tree-pair diagrams in $F(p+1, q+1)$ may require more than just the cancelation of all exposed caret

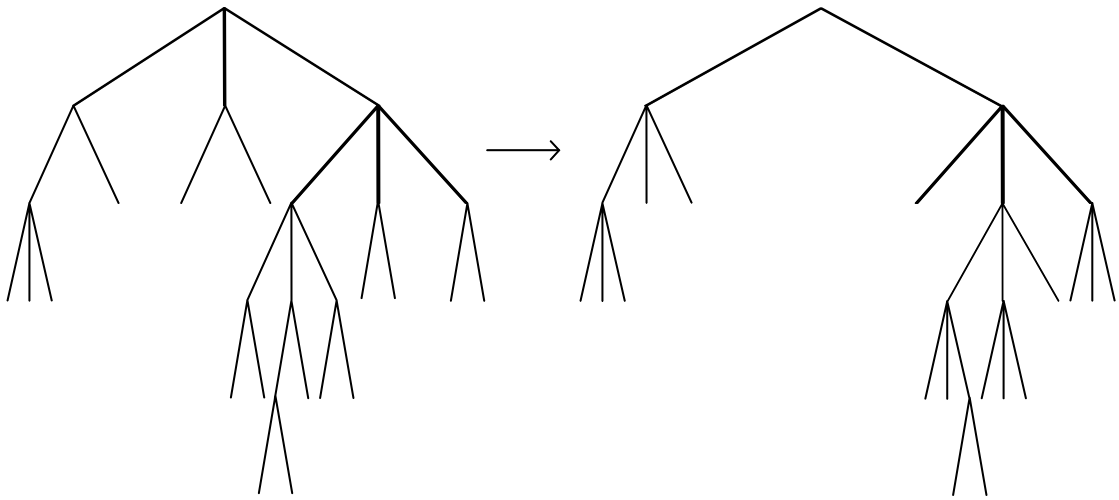

Figure 7.2: Another tree-pair diagram for the identity element in $F(2,3)$

pairs in order to render them identical. There will be other possible substitutions required within the trees in many cases before the cancelation of exposed caret pairs can be done and the minimal tree-pair diagram can be obtained.

Before we proceed to general theorems about how to tell when a pair of trees or tree-pair diagrams is equivalent or about how to reduce a $(p+1, q+1)$–ary tree-pair diagram, we begin with a few necessary definitions:

We will use the notation $l_i^{\pm}$ to represent the leaf numbered $i$ in the tree $T_{\pm}$, and the notation $I_i^{\pm}$ to represent the interval which is represented by the leaf $l_i^{\pm}$. We may leave out the $\pm$ if the tree in which the leaf is present is obvious from the context. In addition, when we refer to *the number of leaves in a tree-pair diagram* $(T_-, T_+)$, we mean the number of carets in $T_-$ or the number of carets in $T_+$, which will be equal, since $T_-$ and $T_+$ will always have the same number of carets.

We will also use the notation $L(I)$ to represent the length of the interval $I$.

**Definition (equivalent trees and tree-pair diagrams) 7.0.1.** *Two* $(p+1, q+1)$*–ary trees are equivalent if they represent the same subdivision of the unit* *interval. Two* $(p+1, q+1)$*–ary tree-pair diagrams are equivalent if they represent the same element of* $F(p+1, q+1)$*.*

**Definition (minimal/reduced tree-pair diagram) 7.0.2.** *A* $(p+1, q+1)$*–ary tree-pair diagram is minimal (or reduced) if it has the smallest number of leaves of any tree-pair diagram in the equivalence class of tree-pair diagrams representing a given element of* $F(p+1, q+1)$*.*

We note that, unlike the case for $F(p+1)$ there may be more than one distinct minimal tree-pair diagram and carets may need to be added in some cases to go from a non-minimal tree-pair diagram of an element to a minimal one. For example, in each of the Figures 7.3 and 7.4, we can see two tree-pair diagrams which represent the same element of $F(2,3)$. In each figure, both of the given tree-pair diagrams have the same number of leaves in each tree, and there is no other equivalent tree-pair diagram with fewer leaves (to prove this we could use Theorem 7.1.11). So each of the elements of $F(2,3)$ represented by the tree-pair diagrams in Figures 7.3 and 7.4 has 2 distinct minimal tree-pair diagrams.

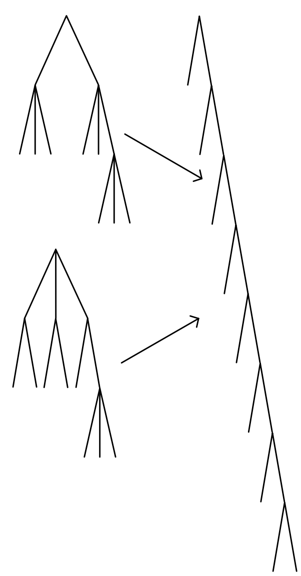

Figure 7.3: Two equivalent but distinct minimal tree-pair diagrams representing an element of $F(2,3)$

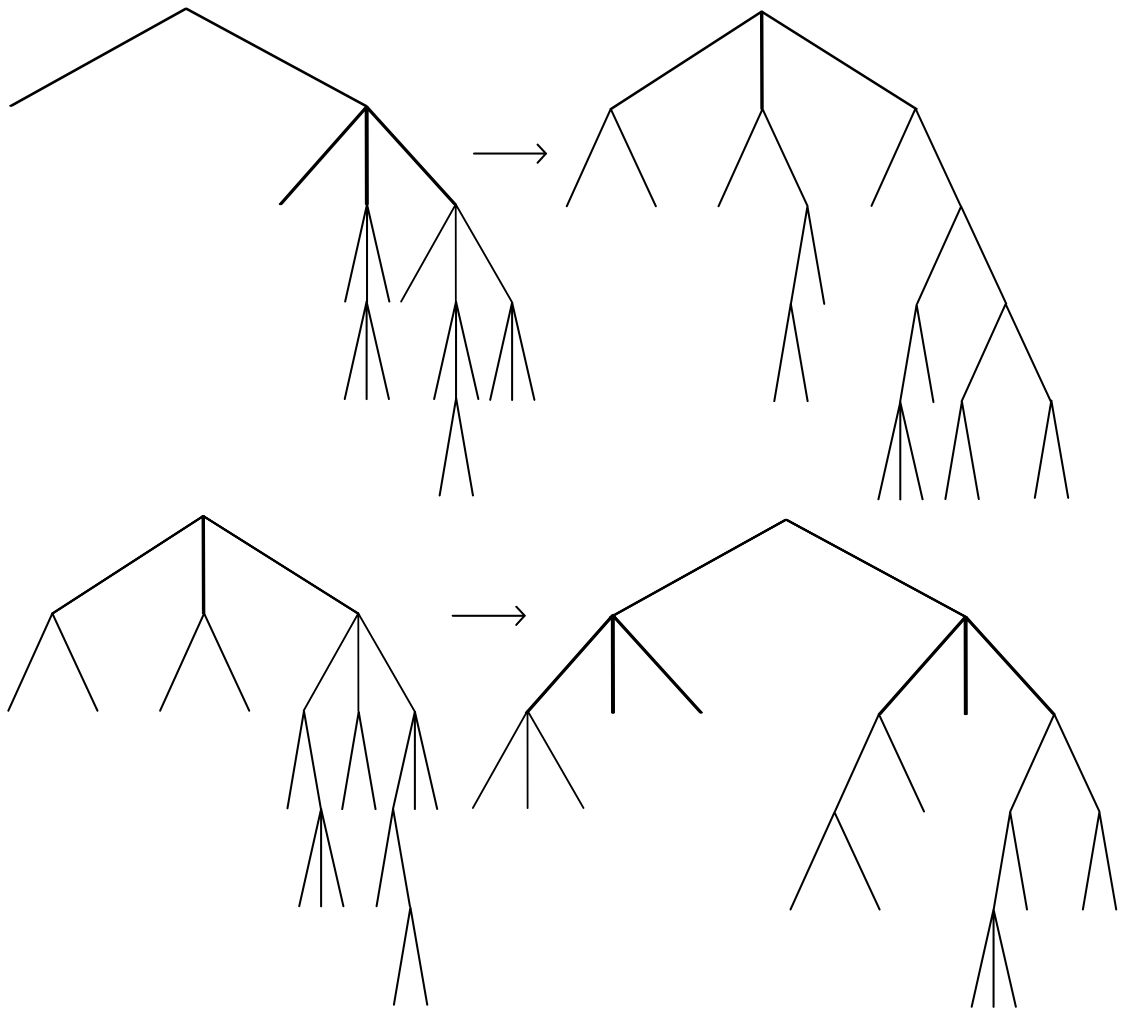

Figure 7.4: Two equivalent but distinct minimal tree-pair diagrams representing another element of $F(2,3)$

For another example of how behavior of minimal tree-pair diagram representatives can behave differently in $F(p+1, q+1)$ than in $F(p+1)$, we look at Figure 7.5, which depicts several tree-pair diagrams, each of which represents an element of $F(2,3)$. In this figure, the tree-pair diagram labeled $(T^1_-, T^1_+)$ has a 2–ary caret added to the last leaf in each tree, yielding the tree-pair diagram labeled $(T^2_-, T^2_+)$. Clearly $(T^2_-, T^2_+)$ is equivalent to the tree-pair diagram $(T^3_-, T^3_+)$ because $T^2_+ = T^3_+$ and $T^2_-$ and $T^3_-$ both represent the same subdivision of the unit interval, and the tree-pair diagram $(T^3_-, T^3_+)$ can be reduced by canceling the 3–ary caret with leaf index numbers 0, 1, and 2 to get the tree-pair diagram $(T^4_-, T^4_+)$. So clearly $(T^1_-, T^1_+)$ is equivalent to $(T^4_-, T^4_+)$, and $(T^4_-, T^4_+)$ is the minimal tree-pair diagram (as in the previous example, to prove this we could use Theorem 7.1.11). So in order to find the minimal tree-pair diagram equivalent to $(T^1_-, T^1_+)$, we had to add a caret to each tree. This would never happen in the $F(p+1)$ case.

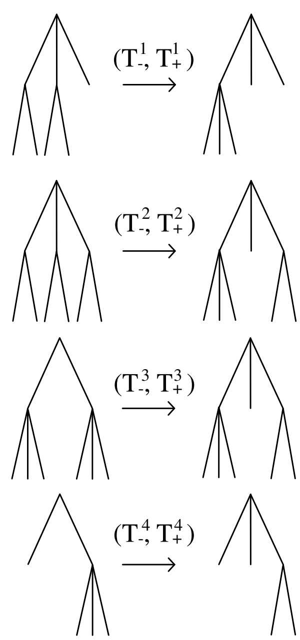


Figure 7.5: $(T^1_-, T^1_+)$ is a (2,3)–ary tree-pair diagram which must have carets added to it in order to be transformed into its equivalent minimal tree-pair diagram $(T^4_-, T^4_+)$

We can see a more complex example of this kind of behavior in Figure 7.6. In this figure, we add two 3–ary carets to the first tree-pair diagram $(T_{0-}, T_{0+})$ in the figure to get the tree-pair diagram $(T_{1-}, T_{1+})$. Then, through a series of substitutions of equivalent subtrees of the kind given in Figure 7.16 (to see these substitutions carried out step-by-step, see Figure 7.7), we can transform $T_{1-}$ into $T_{6-}$, yielding the tree-pair diagram $(T_{6-}, T_{6+})$. Then we can cancel 5 different 2–ary carets in the tree-pair diagram $(T_{6-}, T_{6+})$ to yield the tree-pair diagram $(T_{7-}, T_{7+})$. So $(T_{7-}, T_{7+})$ is clearly equivalent to $(T_{0-}, T_{0+})$ but has fewer leaves and can only be obtained from $(T_{0-}, T_{0+})$ by adding carets.

**Definition (leaf-path) 7.0.3.** *The leaf-path of a given leaf $l_i^{\pm}$ in $T_{\pm}$, which we will often denote by $\gamma_i^{\pm}$ (or simply by $\gamma_i$ if the tree in which it is present is clear from the context), is the string of carets from that leaf's parent caret to the root which have an edge (left, right or middle) on the directed path from the leaf to the root node.*

**Definition (leaf-path valence) 7.0.4.** *The $(p+1)$–valence of a leaf-path $\gamma_i \in T$ is the number of $(p+1)$–ary carets present on that leaf-path and is denoted by $v_{p+1}(l_i)$. Likewise, the $(q+1)$–valence of a leaf-path is the number of $(q+1)$–ary carets present on that leaf-path and is denoted by $v_{q+1}(l_i)$. If we refer to just the valence of a leaf-path or $\mathbf{v}(l_i)$ in a tree, this refers to the vector $\langle v_{p+1}(l_i), v_{q+1}(l_i)\rangle$. For simplicity, in some cases we will refer to $\mathbf{v}(l_i)$ as the valence of the leaf $l_i$. For $l_i^+ \in T_+$ and $l_-^- \in T_-$ with respect to a tree-pair diagram $(T_-, T_+)$, we will also call $\mathbf{v}(l_i^+)$ and $\mathbf{v}(l_i^-)$ the positive and negative valences respectively of the leaf with index $i$ (although this is technically an abuse of terminology, as $\mathbf{v}(l_i^+)$ and $\mathbf{v}(l_i^-)$ actually refer to the valence of leaves in two distinct trees).*

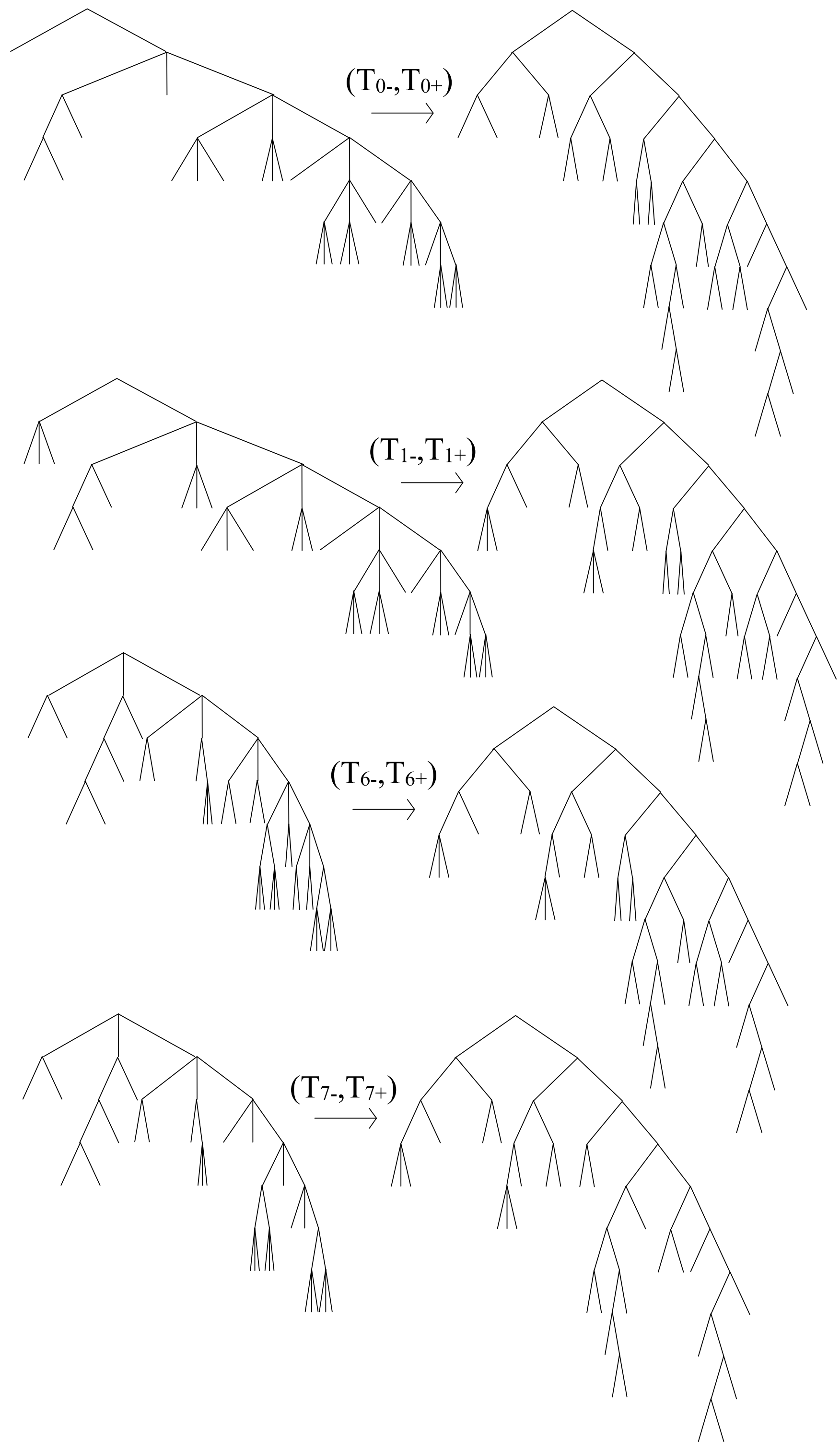


Figure 7.6: $(T_{0-}, T_{0+})$ is a (2,3)–ary tree-pair diagram which must have carets added to it in order to be transformed into its equivalent minimal tree-pair diagram $(T_{7-}, T_{7+})$

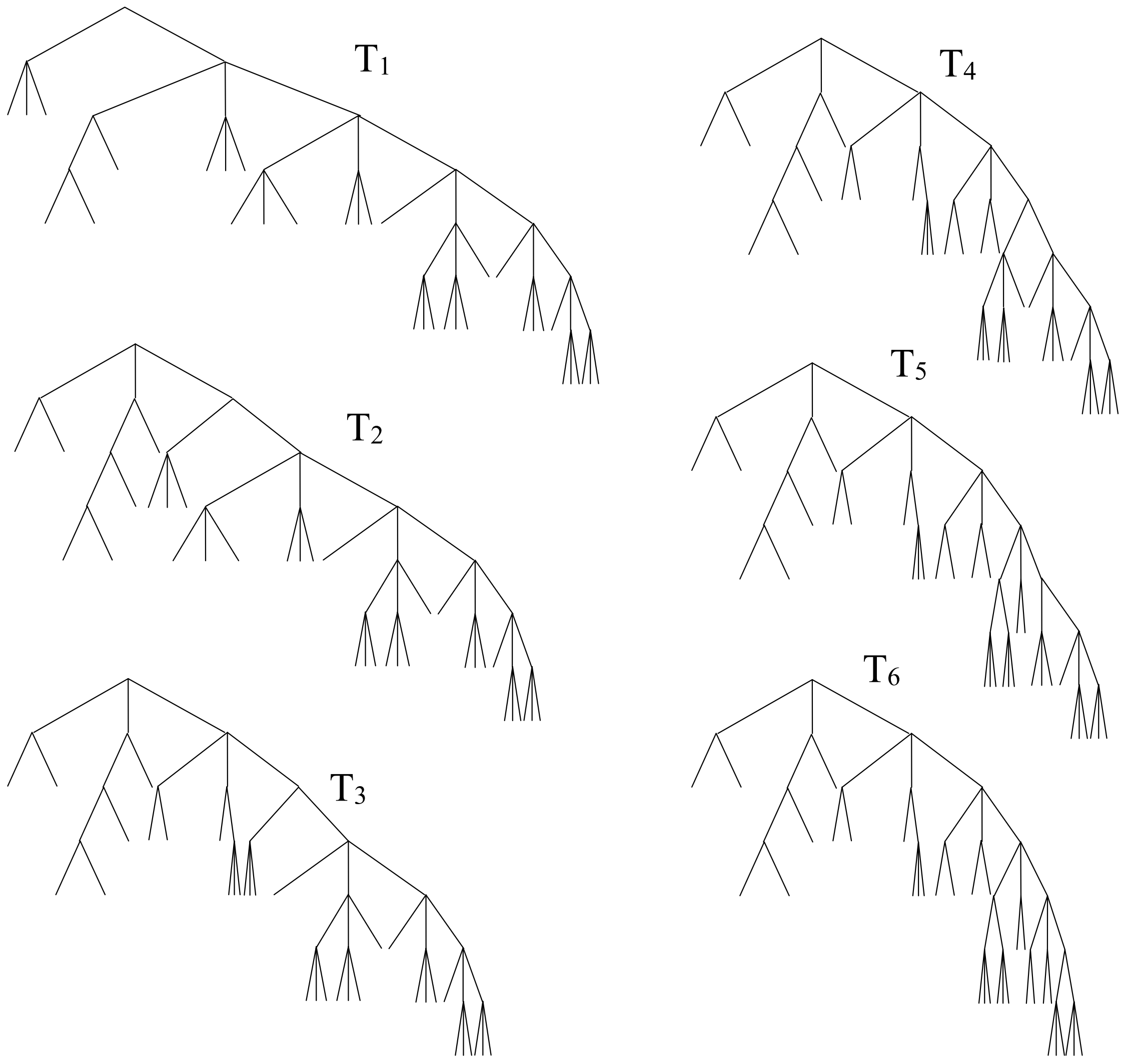


Figure 7.7: The step-by-step transformation using substitutions of the type given in Figure 7.16 taking the tree $T_1$ to the tree $T_6$

## 7.1 Equivalence of Trees and Tree-Pair Diagrams

We begin this section with a theorem:

**Theorem 7.1.1.** *The $(p+1, q+1)$–ary tree $T$ is equivalent to the $(p+1, q+1)$–ary tree $S$ if and only if the number of leaves in $T$ and $S$ is the same and $\mathbf{v}(l_i) = \mathbf{v}(k_i)$ for all leaves $l_i$ in $T$ and $k_i$ in $S$.*

*Proof.* The only if statement contained in this theorem follows immediately from the definition of tree equivalence. Now we prove the if statement contained in this theorem. We begin by considering two trees $T$ and $S$ as subdivisions of the closed unit interval with breakpoints in $\mathbb{Z}[\frac{1}{p+1}, \frac{1}{q+1}$ and we suppose that they have the same number of leaves and that $\mathbf{v}(l_i) = \mathbf{v}(k_i)$ for all leaves $l_i \in T$ and $k_i \in S$. If $\mathbf{v}(l_i) = \mathbf{v}(k_i)$ for all leaves $l_i$ in $T$ and $k_i$ in $S$, then for the intervals $I_i$ represented by the leaf $l_i$ in $T$ and $J_i$ represented by the leaf $k_i$ in $S$:

$$L(I_i) = (p+1)^{-v_{p+1}(l_i)}(q+1)^{-v_{q+1}(l_i)} = (p+1)^{-v_{p+1}(k_i)}(q+1)^{-v_{q+1}(k_i)} = L(J_i)$$

for all $i$. Since $I_0 = [0, a]$ for some $a \in [0, 1]$ and $J_0 = [0, b]$ for some $b \in [0, 1]$, and since $L(I_0) = L(J_0)$, we must have $a = b$. If we have $I_n = [a_n, b_n]$ for $a_n, b_n \in [0, 1]$ and $J_n = [a_n, c_n]$ for $a_n, c_n \in [0, 1]$, then since $L(I_n) = L(J_n)$, we must have $b_n = c_n$, which, if we let $a_{n+1} = b_n = c_n$, implies that $I_{n+1} = [a_{n+1}, b_{n+1}]$ for $a_{n+1}, b_{n+1} \in [0, 1]$ and $J_{n+1} = [a_{n+1}, c_{n+1}]$ for $a_{n+1}, c_{n+1} \in [0, 1]$. So by induction, we will have $I_i = J_i$ for all i. So clearly $T \equiv S$. □

**Corollary 7.1.2.** *The $(p+1, q+1)$–ary tree-pair diagram $(T_-, T_+)$ is equivalent to the $(p+1, q+1)$–ary tree-pair diagram $(S_-, S_+)$ if the number of leaves in $(T_-, T_+)$ and $(S_-, S_+)$ is the same and $\mathbf{v}(l_i^{\pm}) = \mathbf{v}(k_i^{\pm})$ (where the value of $\pm$ is the same for both occurrences in this equation) for all leaves $l_i^{\pm}$ in $(T_-, T_+)$ and $k_i^{\pm}$ in $(S_-, S_+)$.*

We should note that the converse of Corollary 7.1.2 does not hold. For example, Figure 7.8 shows two equivalent (2,3)–ary tree-pair diagrams which have different numbers of leaves in each tree.

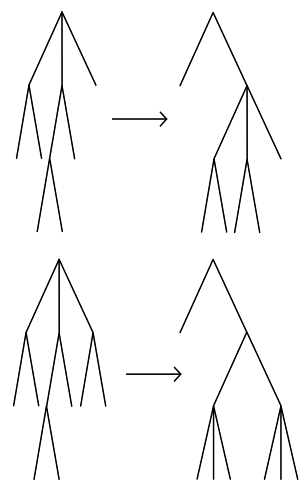

Figure 7.8: Two equivalent but distinct tree-pair diagrams, each with a different number of leaves in each tree

The natural next step is to explore in what circumstances the converse of Corollary 7.1.2 would hold. Is there a way to introduce a new way of classifying the tree-pair diagrams representing elements of $F(p+1, q+1)$ into a new set of equivalence classes so that we can obtain a theorem which gives a clear biconditional condition for determining tree-pair diagram equivalence? In order to ask this question, we first further refine our understanding of how one equivalent tree can be transformed into another.

**Proposition 7.1.3.** *The* $(p+1, q+1)$*–ary tree* $T$ *is equivalent to the* $(p+1, q+1)$*–ary tree* $S$ *if and only if* $T$ *can be transformed into* $S$ *(or vice versa) solely by rearranging the order of the carets on a given leaf-path, perhaps for multiple leaf-paths in the tree.*

*Proof.* This follows immediately from Theorem 7.1.1. □

**Theorem 7.1.4.** *The root caret of a $(p+1, q+1)$–ary tree can be written as an $i$–ary caret (for some $i \in \{p+1, q+1\}$) if and only if every leaf-path in the tree contains at least one $i$–ary caret.*

This obviously also holds for the root caret of subtrees within a larger tree; all we need do is replace the word "tree" everywhere it appears in Theorem 7.1.4 with the word "subtree."

*Proof.* The only if statement of this theorem is obvious. We proceed to prove the if statement. For the duration of this proof, we will let $m, n \in \{p+1, q+1\}$ such that $m \neq n$. We begin by supposing that a tree $T$ has an $n$–ary caret as the root and that no tree exists which is equivalent to $T$ and has an $m$–ary caret as the root. We also suppose that every leaf $l_i$ in the tree $T$ has $m$–valence greater than zero.

Then we can rewrite $T$ so that the bottommost caret visible in each leaf-path is an $m$–ary caret, and all carets above it are $n$–ary carets; the children of the $m$-ary carets on the tree may be non-empty subtrees, but we can draw the tree so that each of these subtrees is represented by a labeled subtree, and the carets contained in these subtrees which are descended from an $m$–ary caret are not themselves included in the drawing of the tree. We will refer to this form as the *$m$–ary leaf form* of $T$. For example, consider the (2,3)–ary tree shown in Figure 7.9. In this case, $n = 3$ and $m = 2$. We can see this tree written in $m$–ary leaf form in Figure 7.10. It is possible to write $T$ in this form because we are guaranteed that there will be an $m$–ary caret somewhere on every leaf-path, so all we need do is begin writing all the carets in the tree, starting with the root and working our way down each leaf-path, until we hit an $m$–ary caret.

Now we take the subtree of $T$ which can be created by removing all the $m$–ary carets in the tree, and all their children. We will call this the $n$–ary subtree of

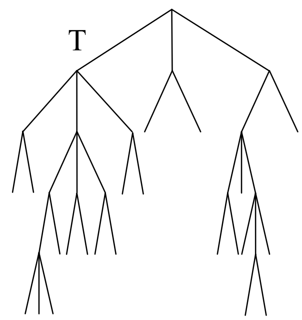


Figure 7.9: A (2,3)–ary tree

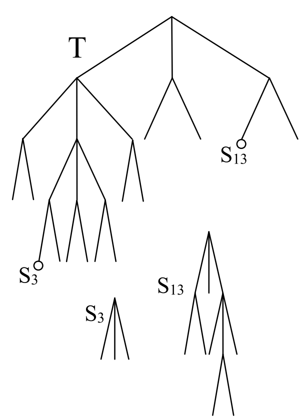


Figure 7.10: The 2–ary leaf form of the tree given in Figure 7.9, where the subtrees $S_3$ and $S_{13}$ are as indicated below the tree

$T$, and denote it by $S_n(T)$. By definition, $S_n(T)$ contains only $n$–ary carets. For example, in Figure 7.11, we can see the 3–ary subtree of the tree given in Figure 7.9.

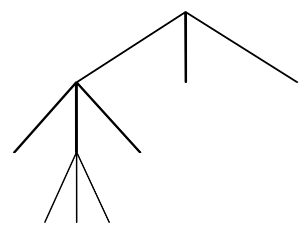

Figure 7.11: The 3–ary subtree of the tree given in Figure 7.9

Because there must be at least one exposed caret in any tree, we can conclude that there must be at least one exposed caret in $S_n(T)$. That same caret when viewed in $T$ will have $n$ $m$–ary children. Therefore, by Theorem 7.1.11, the subtree consisting of that caret and its $n$ children can be replaced in $T$ by the subtree consisting of a single $m$–ary caret with $m$ $n$–ary children; through this substitution of equivalent subtrees, we obtain a tree $T_1$ which is equivalent to $T$.

We let $N(S)$ denote the number of carets in a tree $S$ for the duration of this proof. If we let $r$ denote the number of carets in $S_n(T)$, then $N(S_n(T_1)) = r - 1$. Now we consider $S_n(T_1)$. Clearly $S_n(T_1)$ also contains an exposed $n$–ary caret which has $n$ $m$–ary child carets in $T_1$, so we can replace this subtree with its equivalent subtree as given in Figure 7.0.1. Clearly, if we continue this process so that $T_i$ is the tree obtained by performing the substitution given in Theorem 7.1.11, in the $ith$ step, by induction, $N(S_n(T_i)) = r - i$ and $T_i \equiv T$. So $N(S_n(T_r)) = r - r = 0$ and $T_r \equiv T$. Since $T$ has only a finite number of carets in it by definition, $r$ must be finite. But this means that the $m$–ary subtree of $T_r$ is empty, which implies that the root caret of $T_r$ is an $m$–ary caret, which contradicts our initial assumption that there is no tree equivalent to $T$ which has an $m$–ary root caret. Therefore, if all the leaves in a tree have $m$–valence greater than 0, then there exists a tree equivalent to the original tree which has an $m$–ary caret as the root.

□

For example, to see the step-by-step process by which we replace the tree given in Figure 7.9 with an equivalent tree with a 2–ary caret as the root by using the subtree substitutions given in Figure 7.16 in each step, see Figure 7.12.

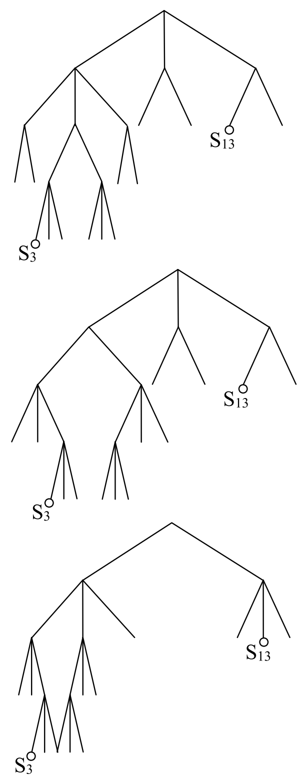


Figure 7.12: Rewriting the tree given in Figure 7.9 as a tree with a 2–ary root caret by using the subtree substitutions given in Figure 7.16 in each step

We will eventually come to the result that any tree can be transformed into any other equivalent tree by a sequence of equivalent subtree replacements of the type indicated by the two equivalent trees given in Figure 7.13 below. Before we can prove this, we will need to establish a definition and a few preliminary results.

**Definition (level) 7.1.5.** *A level of a tree refers to the union of all carets which are the same distance from the root. The root caret itself constitutes the first level of*

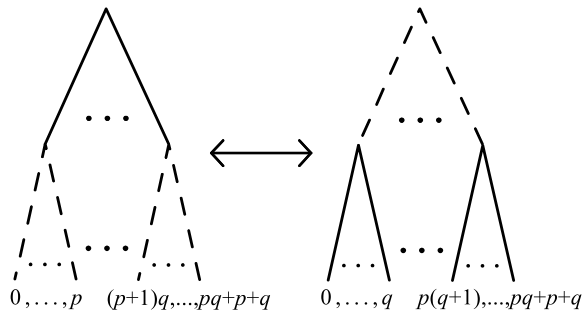


Figure 7.13: Two equivalent subtrees with 2 levels each. Here the carets composed of dotted lines are $(p+1)$–ary and the carets composed of solid lines are $(q+1)$–ary.

*the tree, all of its children constitute the second level of the tree, all of its children's children constitute the third level of the tree, etc. More formally, a caret is in level $i$ if its parent node is distance $i-1$ from the root node of the tree.*

Proposition 7.1.3 asserts that any tree can be transformed into any other equivalent subtree by switching the order of different caret types on the leaf-path of a given leaf (and perhaps on the leaf-paths of multiple leaves in the tree). This is because, since there are only two possible types for each caret, any rearrangement of carets on the path can be reduced to a sequence of type switching.

So we now explore under what conditions we can switch the order of leaves on a path. When we consider rearranging the order of leaves on a path, switching two carets on the path which are of the same type does not change the tree. Therefore, we will limit our discussion to those cases in which we are switching the location of two different types of carets on a leaf path.

**Lemma 7.1.6.** *We consider switching two carets $\wedge_i$ and $\wedge_j$ of opposite type which are on the same leaf path in a tree $T$. Without loss of generality, we suppose that $\wedge_i$ is closer to the root than $\wedge_j$. We take a subtree $S$ of $T$ which has $\wedge_i$ as the root and $\wedge_j$ as an exposed caret, and in which all carets which contain leaves as children* *in $S$ have the same type as $\wedge_j$. If no such subtree exists, then the types of $\wedge_i$ and $\wedge_j$ cannot be switched to produce a tree which is equivalent to $T$. If such a subtree does exist, then the types of $\wedge_i$ and $\wedge_j$ can be switched only if the type of every caret which has leaves as children in $S$ also has its type switched simultaneously.*

*Proof.* Without loss of generality, we suppose that caret $\wedge_i$ is a $(p+1)$–ary caret and that the caret $\wedge_j$ which contains the leaf with leaf-path under consideration is a $(q+1)$–ary caret. Then we suppose that in $T$ and in all possible subtrees of $T$ containing $\wedge_i$ and $\wedge_j$, there exists some other caret $\wedge_k$ in the tree of type $(p+1)$ whose child leaves all have valence $\langle x, y\rangle$ (note: all the child leaves in a caret will always have identical valences). After switching $\wedge_i$ with $\wedge_j$, the $(p+1)$–ary caret $\wedge_k$ will have leaves with valence $\langle x-1, y+1\rangle$, which is not equal to $\langle x, y\rangle$. So if no subtree of $T$ exists which satisfies the conditions of this lemma, then $\wedge_i$ and $\wedge_j$ cannot be switched without producing a tree which is not equivalent to $T$. However, if there exists a subtree $S$ which satisfies the conditions of this lemma, then all carets which have leaves as children in $S$ are $(q+1)$–ary, and we can switch all of these carets to $(p+1)$–ary carets and switch the root caret to a $(q+1)$–ary caret, which will produce leave valences which are identical to those in $S$. □

We note that the condition of Lemma 7.1.6 is not sufficient for the switch to result in a equivalent tree; it is merely a necessary condition. For example, in Figure 7.14, we can see an example of a tree $T$ where the middle child of the left child of the root cannot be switched with the root, because the number of leaves in the tree is not the same after the switch.

**Lemma 7.1.7.** *Suppose we plan to switch the root caret type in a tree with the type of a given exposed caret. For simplicity, suppose that $n, m \in \{p+1, q+1\}$ such that $n \neq m$, the root caret $\wedge_i$ is of type $n$ and the exposed caret under consideration $\wedge_j$ is of type $m$.*

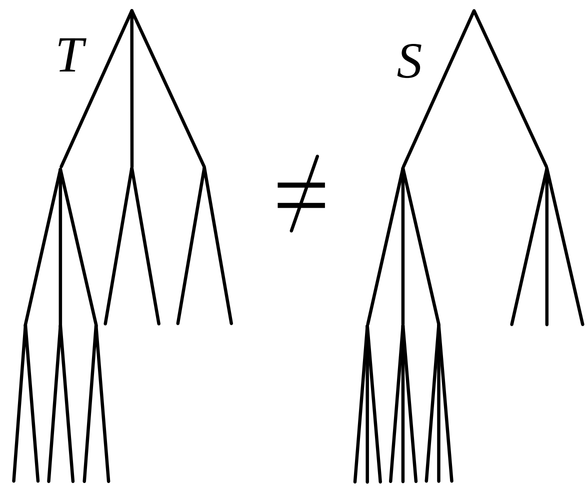


Figure 7.14: An example of a tree which satisfies the conditions of Lemma 7.1.6 but for which the two caret type still cannot be switched without producing a tree which is not equivalent to the original tree

*We will call two carets consecutive if their child vertices represent consecutive subintervals of the unit interval (regardless of whether or not these subintervals are further subdivided by the children of these carets).*

*We take a collection of consecutive $m$–ary carets in the level which contains $\wedge_j$. If all exposed carets which are consecutive to any caret in that collection are also contained in the collection, we will call the collection complete.*

*There must be a complete collection such that the total number of child vertices in the carets of that collection are divisible by both $p + 1$ and $q + 1$.*

*Proof.* We let $n, m \in \{p + 1, q + 1\}$ such that $n \neq m$. We let the root caret $\wedge_i$ be of type $n$ and the exposed caret $\wedge_j$ be of type $m$. Since we are only considering all consecutive $m$–ary carets in the level of the exposed caret under consideration, we are guaranteed that the number of child vertices contained in any subcollection of these carets will be divisible by $m$, as the number of child vertices on each of these carets is divisible by $m$. So any complete collection will have a number of child vertices which is divisible by $m$. Now suppose that there does not exist a complete collection for which the total number of child vertices of the carets in the collection is divisible by $n$. This would mean that when we switch the $n$–ary root caret with all the $m$–ary carets in a given complete collection, the total number of child vertices in that collection will be different after this switching process. If it increases, then this means that either the total number of leaves in the tree has increased so that the resulting tree cannot be equivalent to $T$ or one or more vertices have moved from a different level to the level of $\wedge_j$ and the valences of any vertices which change levels must be different, so again the resulting tree cannot be equivalent to $T$. If the number of child vertices in the collection decreases, then this means that either the total number of leaves in the tree has decreased so that the resulting tree cannot be equivalent to $T$ or one or more vertices have moved from a different level to the level of $\wedge_j$ and the valences of any vertices which change levels must be different, so again the resulting tree cannot be equivalent to $T$. So in order for the switching to result in equivalent trees, there must be a complete collection such that the total number of child vertices in the carets of that collection are divisible by both $p+1$ and $q+1$. □

**Lemma 7.1.8.** *The pair of trees given in Figure 7.13 is the only pair of distinct, equivalent,* $(p+1, q+1)$*–ary trees with two levels or less.*

*Proof.* Clearly the only equivalent trees with 1 level are carets which are identical. So we need consider only the case of trees with 2 levels. We let $n, m \in \{p+1, q+1\}$ such that $n \neq m$. We let the root caret be of type $n$ and the exposed caret under consideration be of type $m$, and we try to switch these two carets. By Lemma 7.1.7, there must exist a complete collection of $m$–ary carets in the second level for which the total number of leaves is divisible by $n$ and $m$. The smallest number divisible by $n$ and $m$ is $nm$, which is also the maximum number of leaves possible in an $(n, m)$–ary tree. Since all of the carets in this complete collection must be $m$–ary, then we can only perform the switch when the root caret has $n$ $m$–ary carets as its children. □

**Corollary 7.1.9.** *If it is possible to transform a tree into a distinct equivalent tree by switching two carets which are adjacent on the leaf-path of a given leaf on the tree, then it is possible to to obtain this transformation through a substitution of one equivalent subtree given in Figure 7.13 for the other equivalent subtree given in that figure.*

*Proof.* This follows immediately from Lemma 7.1.8. □

**Lemma 7.1.10.** *If it is possible to transform a tree into a distinct equivalent tree by switching the root caret type of the tree with the type of an exposed caret in the $(k+1)$th level, then it is possible to to obtain this transformation through the combination of a sequence of substitutions of the type given in Figure 7.13 and a switch of the type of the root caret of a subtree with the type of an exposed caret on the $k$th level of the subtree, for all $k \geq 2$.*

*Proof.* We will prove this by induction. We begin by proving the case when $k = 2$. We let $n, m \in \{p+1, q+1\}$ such that $n \neq m$. We let the root caret $\wedge_i$ be of type $n$ and the exposed caret under consideration $\wedge_j$ be of type $m$, and we try to switch the types of these two carets. By Lemma 7.1.6, in order for this switching to produce an equivalent tree, all leaves must be children in $m$–ary carets. This implies that the $m$–valence of all leaves will be at least one. So by Theorem 7.1.4, the tree $T$ can be rewritten as a tree $T^1$ with an $m$–ary root caret with $m$ $n$–ary children, where one of these children now has as a descendant the exposed caret in question (although in the process of rewriting, the type of the exposed caret in question may have already been switched to $n$–ary in the tree $T^1$). By looking at the proof of Theorem 7.1.4, we can see that this process of rewriting is done entirely by a sequence of substitutions of those given in Figure 7.13. So the type of the root is switched by replacing the subtree consisting of the $n$–ary root caret with its $n$ $m$–ary children in $T$ by the equivalent subtree consisting of an $m$–ary root with $m$ $n$–ary children to obtain $T^1$, which is also a substitution of the type given in Figure 7.13. Now we consider the exposed caret $\wedge_j$ again. It now has an $n$–ary parent caret. If the exposed caret $\wedge_j$ has already been converted to an $n$–ary caret in the process of rewriting $T$ as $T^1$, then we are done. Then the switching has produced an equivalent tree solely by the use of subtree substitutions of the kind given in Figure 7.0.1. If the exposed caret $\wedge_j$ is still $m$–ary after the transformation of $T$ into $T^1$, then we can convert it to type $n$–ary if and only if it is possible for us to switch this exposed caret $\wedge_j$ with its $n$–ary parent caret, which can be seen as a switch of two carets in a subtree with 2 levels: the subtree consisting of the parent caret of the exposed caret $\wedge_j$ and all its children.

Now we proceed to the induction step. Suppose we have a tree with a root caret $\wedge_i$ and an exposed caret $\wedge_j$ in the $(k+1)$th level of opposite type whose types we would like to switch in order to produce an equivalent tree. Again we let $n, m \in \{p+1, q+1\}$ such that $n \neq m$. We let the root caret $\wedge_i$ be of type $n$ and the exposed caret under consideration $\wedge_j$ be of type $m$, and we try to switch the types of these two carets. By Lemma 7.1.6, in order for this switching to produce an equivalent tree, all leaves must be children in $m$–ary carets. This implies that the $m$–valence of all leaves will be at least one. So by Theorem 7.1.4, the tree $T$ can be rewritten as a tree $T^1$ with an $m$–ary root caret with $m$ $n$–ary children, where one of these children now has as a descendant the exposed caret in question (although in the process of rewriting, the type of the exposed caret in question may have already been switched to $n$–ary in the tree $T^1$). By looking at the proof of Theorem 7.1.4, we can see that this process of rewriting is done entirely by a sequence of substitutions of those given in Figure 7.13. So the type of the root is switched by replacing the subtree consisting of the $n$–ary root caret with its $n$ $m$–ary children in $T$ by the equivalent subtree consisting of an $m$–ary root with $m$ $n$–ary children to obtain $T^1$, which is also a substitution of the type given in Figure 7.13. Now we consider the exposed caret $\wedge_j$ again. It now has an $n$–ary parent caret. If the exposed caret $\wedge_j$ has already been converted to an $n$–ary caret in the process of rewriting $T$ as $T^1$, then we are done. Then the switching has produced an equivalent tree solely by the use of subtree substitutions of the kind given in Figure 7.0.1. If the exposed caret $\wedge_j$ is still $m$–ary after the transformation of $T$ into $T^1$, then we can convert it to type $n$–ary if and only if it is possible for us to switch this exposed caret $\wedge_j$ with its the $n$–ary parent caret of the new subtree consisting of the parent caret of the exposed caret $\wedge_j$ and all its children; the exposed caret $\wedge_j$ is in level $k$ of that subtree. □

**Theorem 7.1.11.** *Any tree can be transformed into any other equivalent tree by a sequence of equivalent subtree replacements of the type indicated by the two equivalent trees given in Figure 7.15.*

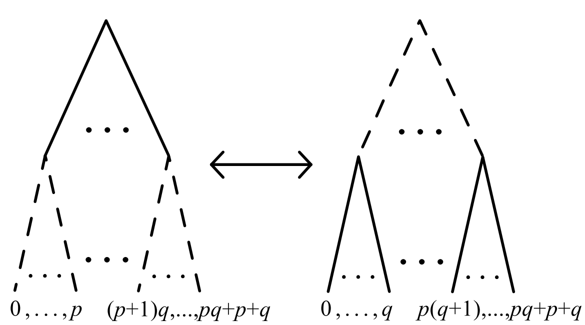


*Figure 7.15: The two equivalent subtrees of Theorem 7.1.11. Here the carets composed of dotted lines are $(p+1)$–ary and the carets composed of solid lines are $(q+1)$–ary.*

*For example, in the case when $p = 1$ and $q = 2$, the equivalent subtree replacement diagram would be of the type given in Figure 7.16.*

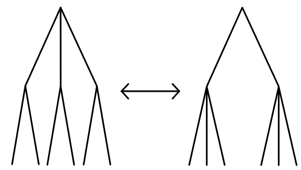

*Figure 7.16: The two equivalent subtrees of Theorem 7.1.11 when $p = 1$ and $q = 2$.*

*Proof.* Suppose we have two equivalent $(p+1, q+1)$–ary trees, $T$ and $S$. From Proposition 7.1.3, we know that we can transform $T$ into $S$ and vice versa by rearranging the order of carets within a given leaf-path, for a certain number of leaf-paths in the tree. So we consider all the ways in which we could switch two carets on a leaf-path. We begin by observing that switching two carets on a leaf-path will only affect the valences of the leaves which are descended from the switched caret in the lower-numbered level (i.e. the caret which is closer to the root in the tree). Therefore switching two carets on a leaf path in a tree $T$ will only change the valence of leaves which are descended from the higher-numbered level switching caret if it would also change the valence of at least one of the child vertexes of that caret. We need only consider the subtree which consists of the switching caret in the lower-numbered level and all its descendants except those descendants which are also descendants of the switching caret in the higher-numbered level.

Now the result follows immediately from Lemma 7.1.10.

□

From Corollary 7.1.2, we can see that the valence of the leaf-paths in a tree-pair diagram is central in determining the equivalence of tree-pair diagrams representing elements of $F(p+1, q+1)$, so we introduce the idea of a *leaf-path chart* for a tree-pair diagram. We will then proceed to try to develop a unique representative up to equivalence for elements of $F(p+1, q+1)$ similar to the way minimal tree-pair diagrams are unique representatives up to equivalence for elements of $F(p+1)$.

**Definition (leaf-path chart) 7.1.12.** *The leaf-path chart for a given tree-pair diagram is a chart which lists the leaf index numbers in the first column, the negative valence of the leaf with that number in the second column (which we will refer to as the negative column) and the positive valence of the leaf with that number in the third column (which we will refer to as the positive column). We will denote the chart of a tree-pair diagram* $(T_-, T_+)$ *by* $C(T)$ *or by* $C(x)$ *if the context is clear, where* $x$ *is the element of* $F(p+1, q+1)$ *which can be represented by the minimal tree-pair diagram* $(T_-, T_+)$*. For an example of a leaf-path chart for a given tree-pair diagram, see Figure 7.17 and Table 7.1.*

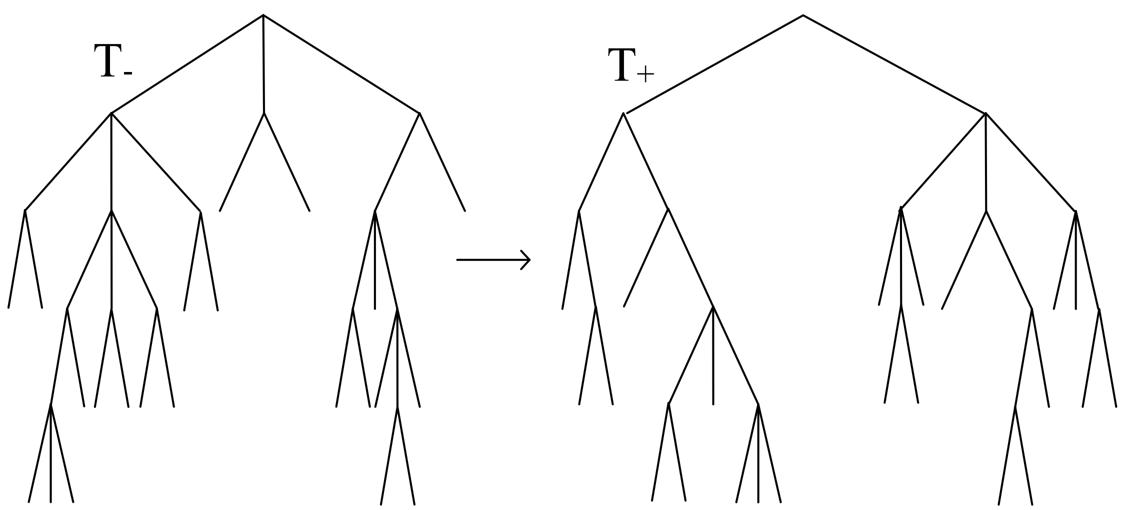


Figure 7.17: A (2,3)–ary tree-pair diagram

We note that a given leaf-path chart may represent more than one distinct tree-pair diagram, although it is clear from Corollary 7.1.2 that any distinct tree-pair diagrams represented by the same chart will be equivalent with respect to the element of $F(p+1, q+1)$ which they represent. For example, in Figure 7.18, we can see two distinct trees which have the same leaf-path chart (see Table 7.2).

Table 7.1: Leaf-path chart for tree-pair diagram given in Figure 7.17

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle 1,2\rangle$ | $\langle 3,0\rangle$ |
| 1 | $\langle 1,2\rangle$ | $\langle 4,0\rangle$ |
| 2 | $\langle 1,4\rangle$ | $\langle 4,0\rangle$ |
| 3 | $\langle 1,4\rangle$ | $\langle 3,0\rangle$ |
| 4 | $\langle 1,4\rangle$ | $\langle 4,1\rangle$ |
| 5 | $\langle 1,3\rangle$ | $\langle 4,1\rangle$ |
| 6 | $\langle 1,3\rangle$ | $\langle 3,1\rangle$ |
| 7 | $\langle 1,3\rangle$ | $\langle 3,2\rangle$ |
| 8 | $\langle 1,3\rangle$ | $\langle 3,2\rangle$ |
| 9 | $\langle 1,3\rangle$ | $\langle 3,2\rangle$ |
| 10 | $\langle 1,2\rangle$ | $\langle 1,2\rangle$ |
| 11 | $\langle 1,2\rangle$ | $\langle 2,2\rangle$ |
| 12 | $\langle 1,1\rangle$ | $\langle 2,2\rangle$ |
| 13 | $\langle 1,1\rangle$ | $\langle 1,2\rangle$ |
| 14 | $\langle 2,2\rangle$ | $\langle 2,1\rangle$ |
| 15 | $\langle 2,2\rangle$ | $\langle 2,1\rangle$ |
| 16 | $\langle 1,2\rangle$ | $\langle 3,2\rangle$ |
| 17 | $\langle 1,3\rangle$ | $\langle 3,2\rangle$ |
| 18 | $\langle 2,3\rangle$ | $\langle 1,2\rangle$ |
| 19 | $\langle 2,3\rangle$ | $\langle 1,2\rangle$ |
| 20 | $\langle 1,3\rangle$ | $\langle 2,2\rangle$ |
| 21 | $\langle 1,1\rangle$ | $\langle 2,2\rangle$ |

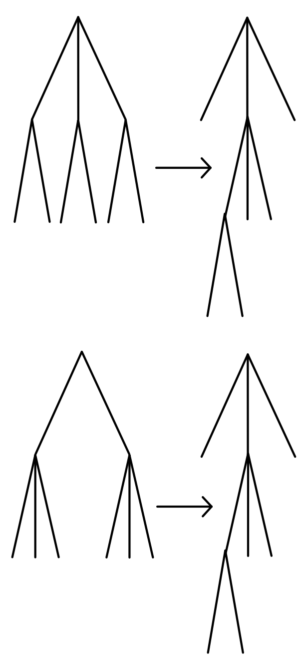

Figure 7.18: Two (2,3)–ary tree-pair diagrams with the same leaf-path chart

Table 7.2: Leaf-path chart for tree-pair diagrams given in Figure 7.18

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle 1, 1\rangle$ | $\langle 0, 1\rangle$ |
| 1 | $\langle 1, 1\rangle$ | $\langle 1, 2\rangle$ |
| 2 | $\langle 1, 1\rangle$ | $\langle 1, 2\rangle$ |
| 3 | $\langle 1, 1\rangle$ | $\langle 0, 2\rangle$ |
| 4 | $\langle 1, 1\rangle$ | $\langle 0, 2\rangle$ |
| 5 | $\langle 1, 1\rangle$ | $\langle 0, 1\rangle$ |

It is clear from Corollary 7.1.2 that two tree-pair diagrams which have identical leaf-path charts will be equivalent.

**Corollary 7.1.13.** *The leaf-path chart of any minimal tree-pair diagram representative of an element of* $F(p+1, q+1)$ *is unique.*

*Proof.* This follows immediately from Corollary 7.1.2. □

However, there may be equivalent tree-pair diagrams which do not have identical leaf-path charts. For example, we consider an element $x$ of $F(2, 3)$ represented by the tree-pair diagram given in Figure 7.17; this tree-pair diagram is equivalent to the tree-pair diagram given in Figure 7.19, which we can see by replacing the subtree of $T_-$ consisting of a 3–ary caret root with 3 2–ary carets as children with a subtree consisting of a 2–ary caret root with 2 3–ary caret children, and then the second 3-ary caret child, which has leaves numbered 7, 8, and 9 in $T_-$ can be canceled with the corresponding exposed 3–ary caret in $T_+$. In this case we will have $C(T) \neq C(S)$ (see Tables 7.1 and 7.3). To observe this, we need only note that the number of leaves in each tree-pair diagram will be different, and therefore the leaf-path charts of each diagram will each have a different number of rows. So we can conclude that there are elements of $F(2, 3)$ which have equivalent tree-pair diagrams which have leaf-path charts which are not identical.

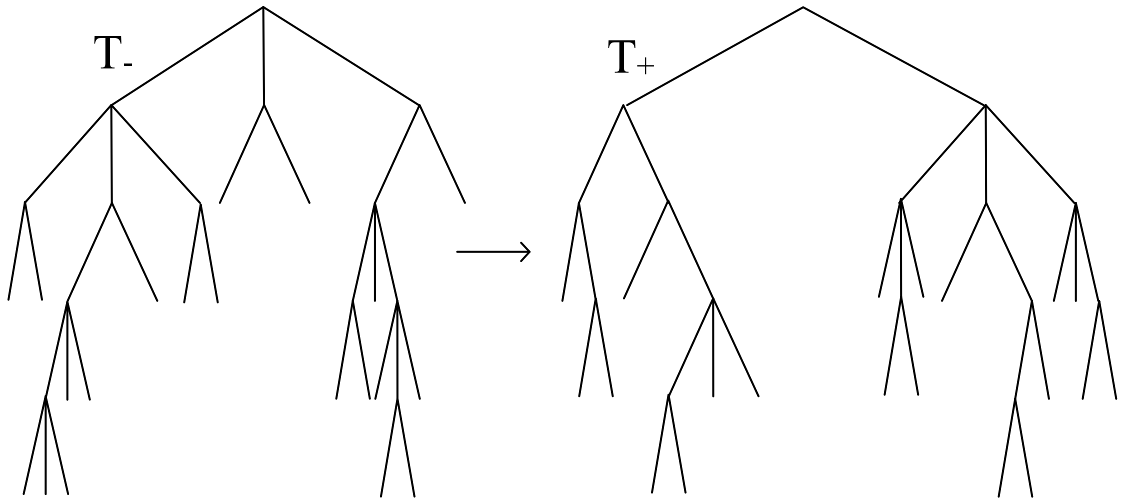


Figure 7.19: A (2,3)–ary tree-pair diagram which is equivalent to the tree-pair diagram given in Figure 7.17

Table 7.3: Leaf-path chart for tree-pair diagram given in Figure 7.19

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle 1,2\rangle$ | $\langle 3,0\rangle$ |
| 1 | $\langle 1,2\rangle$ | $\langle 4,0\rangle$ |
| 2 | $\langle 1,4\rangle$ | $\langle 4,0\rangle$ |
| 3 | $\langle 1,4\rangle$ | $\langle 3,0\rangle$ |
| 4 | $\langle 1,4\rangle$ | $\langle 4,1\rangle$ |
| 5 | $\langle 1,3\rangle$ | $\langle 4,1\rangle$ |
| 6 | $\langle 1,3\rangle$ | $\langle 3,1\rangle$ |
| 7 | $\langle 1,2\rangle$ | $\langle 3,1\rangle$ |
| 8 | $\langle 1,2\rangle$ | $\langle 1,2\rangle$ |
| 9 | $\langle 1,2\rangle$ | $\langle 2,2\rangle$ |
| 10 | $\langle 1,1\rangle$ | $\langle 2,2\rangle$ |
| 11 | $\langle 1,1\rangle$ | $\langle 1,2\rangle$ |
| 12 | $\langle 2,2\rangle$ | $\langle 2,1\rangle$ |
| 13 | $\langle 2,2\rangle$ | $\langle 2,1\rangle$ |
| 14 | $\langle 1,2\rangle$ | $\langle 3,2\rangle$ |
| 15 | $\langle 1,3\rangle$ | $\langle 3,2\rangle$ |
| 16 | $\langle 2,3\rangle$ | $\langle 1,2\rangle$ |
| 17 | $\langle 2,3\rangle$ | $\langle 1,2\rangle$ |
| 18 | $\langle 1,3\rangle$ | $\langle 2,2\rangle$ |
| 19 | $\langle 1,1\rangle$ | $\langle 2,2\rangle$ |

It is clear that adding a caret to the same leaf in both the positive and negative tree of a tree-pair diagram will not change the element which the tree-pair diagram represents. When an $i$–ary caret is added to the leaf with index number $n$ on both the positive and negative trees of a tree-pair diagram, the corresponding change to its leaf-path chart will be for the $i$–valence of $l_n$ to go up by one in both trees, and for $i-1$ extra rows to be added to the chart so that these new rows have index numbers $n+1, ...n+i-1$ and so that each of these new rows has positive and negative valence identical to the new positive and negative valence respectively of $l_n$. Then all rows which used to have index numbers $n+1, ..., m-1$ (where $m$ is the total number of leaves on the original tree) will now have index numbers $n+i, ..., m+i-2$ respectively. We will refer to this procedure on leaf-path charts which is equivalent to adding a pair of carets to a tree-pair diagram as *leaf-path chart addition*. For example, if we look at the leaf-path chart given in Table 7.19, we can get to the leaf-path chart given in Table 7.17 by performing 3–ary leaf-path chart addition on the row with index number 7.

Since we can always add a $(p+1)$–ary or $(q+1)$–ary caret to any leaves with the same index number in both trees of a tree-pair diagram, we can likewise always perform leaf-path chart addition as described in the preceding paragraph. The reverse action however, is not so simple. We may not be able to cancel any $(p+1)$ or any $(q+1)$ consecutive rows in a leaf-path chart which have identical positive and identical negative valences and replace them with a single row with $(p+1)$ (or $(q+1)$) positive and negative valence reduced by one, because there may be no tree-pair diagram represented by the given leaf-path chart which has those $(p+1)$ (or $(q+1)$) leaves located in the same $(p+1)$–ary (or$(q+1)$–ary) caret in both trees. We explore this question, but first we give a formal definition of this new kind of cancelation.

**Definition (leaf-path chart cancelation) 7.1.14.** *The process of replacing* $(p+1)$ *or* $(q+1)$ *consecutive rows with identical valences* $\langle x, y\rangle$ *in a leaf-path chart when* $x \geq 1$ *or* $y \geq 1$ *respectively with a single row with valence* $\langle x-1, y\rangle$ *or* $\langle x, y-1\rangle$ *respectively (which corresponds to removing a pair of exposed carets in a tree-pair diagram) is called leaf-path chart cancelation.*

We may not be able to perform cancelation on any given leaf-path chart and obtain a chart as a result which can be represented by a tree-pair diagram. For example, the leaf-path chart given in Table 7.4 can be represented by the tree-pair diagram given in Figure 7.20. This leaf-path chart has two possible places where cancelation could take place: rows 1 and 2 and/or rows 3 and 4 could be canceled through 2–ary cancelation.

Table 7.4: Leaf-path chart for tree-pair diagrams given in Figure 7.4

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle 1,1\rangle$ | $\langle 1,2\rangle$ |
| 1 | $\langle 2,1\rangle$ | $\langle 1,2\rangle$ |
| 2 | $\langle 2,1\rangle$ | $\langle 1,2\rangle$ |
| 3 | $\langle 2,1\rangle$ | $\langle 1,1\rangle$ |
| 4 | $\langle 2,1\rangle$ | $\langle 1,1\rangle$ |
| 5 | $\langle 1,0\rangle$ | $\langle 1,0\rangle$ |

However, either of these cancelations would result in a positive 2–ary valence of 0; since row 5 already has a positive 3–ary valence of 0, this would mean that if a tree-pair diagram exists which would represent the leaf-path chart resulting from the cancelation, then the positive tree in this tree-pair diagram could have neither a 2–ary caret (because there is no 2–ary caret on the leaf path of the rows in the table resulting from either cancelation) nor a 3–ary caret (because there is no 3–ary caret on the leaf path of row 5) as the root. As this is clearly a contradiction, we can conclude that no tree-pair diagram exists which would represent the leaf-path chart which would result from either or both of the possible cancelations.

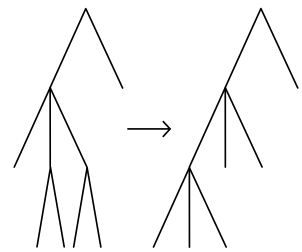

Figure 7.20: The (2,3)–ary tree-pair diagram which is equivalent to the leaf-path chart given in Table 7.4

In general, it may be difficult to tell whether or not we can perform leaf-path chart cancelation on an instance of $(p+1)$ (or $(q+1)$) consecutive rows in a leaf-path chart, because it may be difficult to tell whether or not there is any instance of a tree-pair diagram which can be represented by the given leaf-path chart for which those $(p+1)$ (or $(q+1)$) consecutive leaves which correspond to the canceled rows on the chart are all leaves of the same $(p+1)$–ary (or $(q+1)$–ary) caret.

**Lemma 7.1.15.** *Leaf-path chart cancelation can be performed exactly when there exists at least one tree-pair diagram which can be represented by the chart which would result from the cancelation.*

*Proof.* This becomes apparent when we consider the leaf-path chart as a representation of an element of $F(p+1, q+1)$ viewed as a homeomorphism. Without loss of generality, we will consider $(p+1)$–ary cancelation. Cancelation of this type requires that we have $p+1$ consecutive rows on the chart with identical positive and negative valences, and that the positive and negative $(p+1)$–valences are greater than or equal to 1. Let these be the rows indexed with numbers $i, ..., i+p$. If we let $I_i'$ denote the interval we obtain from canceling $I_i, ..., I_{i+p}$ (and $l_i'$ denote its corresponding leaf), we can see that:

$$L(I_i^{\pm} \cap \cdots \cap I_{i+p}^{\pm}) = \sum_{j=i}^{i+p} \frac{1}{(p+1)^{v_{p+1}^{\pm}(l_j)}(q+1)^{v_{q+1}^{\pm}(l_j)}}$$

Since all of these valences are equal, and since $v(l_i'^{\pm}) = v(l_i^{\pm}) - (1,0)$, this yields:

$$L(I_i^{\pm} \cap \cdots \cap I_{i+p}^{\pm}) = \frac{p+1}{(p+1)^{v_{p+1}^{\pm}(l_j)}(q+1)^{v_{q+1}^{\pm}(l_j)}} = \frac{1}{(p+1)^{v_{p+1}^{\pm}(l_j)-1}(q+1)^{v_{q+1}^{\pm}(l_j)}} = L(I_i')$$

So in both the positive and negative halves of the leaf-path chart, $(p+1)$ intervals of length $\frac{s^{\pm}}{(p+1)}$ are being replaced with one interval of length $s^{\pm}$ (for some $s^{\pm} \in \mathbb{Z}[\frac{1}{(p+1)(q+1)}]$). This means that canceling any consecutive $p+1$ rows which have identical positive and negative valence will not change the element of $F(p+1, q+1)$ which the leaf-path chart represents. So as long as the leaf-path chart resulting from the cancelation has a tree-pair diagram representative, then any tree-pair diagram represented by the leaf-path chart before cancelation will be equivalent to any tree-pair diagram represented by the leaf-path chart after cancelation. (If the leaf-path chart resulting from a cancelation does not correspond to any $(p+1, q+1)$–ary tree-pair diagram, then it will still represent an element of $F(p+1, q+1)$, as we can see from the above discussion, but the leaf-path valences will no longer describe the actual presence of carets in a $(p+1, q+1)$–ary tree-pair diagram, since no $(p+1, q+1)$–ary tree-pair diagram exists with those particular valences.) □

In order to resolve this question of when we can perform leaf-path chart cancelation, we need a theorem which enumerates the conditions under which a $(p+1, q+1)$–ary leaf-path chart can represent at least one possible $(p+1, q+1)$–ary tree-pair diagram.

**Remark 7.1.16.** *There exists a* $(p+1, q+1)$*–ary tree-pair diagram which represents a given leaf-path chart if and only if there exists a tree with the exact valences given in the positive column of the leaf-path chart, and there exists a tree with the exact valences given in the negative column of the leaf-path chart.*

So we can simplify our question by looking at one side of a leaf-path chart at a time. In order to do this, we introduce the following definitions:

**Definition (tree chart) 7.1.17.** *A tree chart is a chart with the leaf index numbers listed in the first column, and the corresponding valences of these leaves listed in the second column.*

When referring to a given leaf-path chart which can be represented by a tree-pair diagram, its *negative* or *positive tree chart* is the tree chart formed by taking its first column and the column containing the valences of the leaves in its negative or positive tree respectively.

We want to be able to talk about whether or not a chart is representable by a tree-pair diagram; since a tree chart by definition will always refer to a chart which can be represented by a tree, we need to develop some terminology to refer to charts which may not be representable by a tree. We begin by defining a concept for leaf-path charts which is analogous to the concept of valence in tree-pair diagrams but which is defined in terms of piecewise-linear homeomorphisms rather than in terms of $(p+1, q+1)$–ary trees.

**Definition (subinterval weight) 7.1.18.** *For fixed $p, q \in \mathbb{N}$ such that $p+1$ and $q+1$ are relatively prime, we consider a subdivision of the closed unit interval such that any given subinterval $I_i$ has length of the form:*

$$\frac{1}{(p+1)^{x_i}(q+1)^{y_i}}$$

*for some $x_i, y_i \in \mathbb{Z}^*$.*

*For such a subdivision, we call the numbers $x_i$ and $y_i$ the $(p+1)$–weight and the $(q+1)$–weight respectively, and we will call the vector $\langle x_i, y_i \rangle$ the subinterval weight. We will denote the weight of an interval $I_i$ by $\mathbf{w}(I_i)$ and the $(p+1)$–weight and $(q+1)$–weight of $I_i$ by $w_{p+1}(I_i)$ and $w_{q+1}(I_i)$ respectively.*

We note that for any leaf-path chart which can be represented by a $(p+1, q+1)$–ary tree-pair diagram, the length of the interval of any given row in the chart will be of the form:

$$\frac{1}{(p+1)^{x_i}(q+1)^{y_i}}$$

where $x_i, y_i \in \mathbb{Z}^*$ because $\mathbf{v}(l_i) = \langle x_i, y_i \rangle$. We also note that performing leaf-path chart cancelation on such a leaf-path chart will not change this fact. This leads us to make the following remark, which obviously follows from the definition of valence and of $(p+1, q+1)$–ary tree-pair diagrams.

**Remark 7.1.19.** *The valence of a leaf $l_i$ in a $(p+1, q+1)$–ary tree is $\langle x_i, y_i \rangle$ if and only if the subinterval $I_i$ (of the closed unit interval) which $l_i$ represents has length*

$$\frac{1}{(p+1)^{x_i}(q+1)^{y_i}}$$

*and therefore weight $\langle x_i, y_i \rangle$.*

By definition, any give subinterval $I_i$ on a leaf-path chart (even if that leaf-path chart results from leaf-path chart cancelation performed on another leaf-path chart) can be written in the form

$$\frac{1}{(p+1)^{x_i}(q+1)^{y_i}}$$

for some $x_i, y_i \in \mathbb{Z}^*$.

So now we define a concept analogous to the concept of a tree chart for trees for which the definition is based on subdivisions of the closed unit interval rather than trees.

**Definition (subinterval chart) 7.1.20.** *A $(p+1, q+1)$–ary subinterval chart is a representation of a certain subdivision of the closed unit interval for which each interval $I_i$ has length of the form*

$$\frac{1}{(p+1)^{x_i}(q+1)^{y_i}}$$

*for some* $x_i, y_i \in \mathbb{Z}^*$*; it is a chart with the subinterval index numbers listed in the first column (the first subinterval is indexed by 0 and the remaining subintervals are numbered in increasing order from left to right), and the corresponding subinterval weights of these subintervals listed in the second column.*

If we are not sure whether or not a leaf-path chart can be represented by a tree-pair diagram, then we call the subinterval chart formed by taking its first column and its negative or positive column its *negative* or *positive subinterval chart* respectively.

So every leaf-path chart can be represented by a pair of subinterval charts with identical indexes, and if and only if that leaf-path chart can be represented by a tree-pair diagram can that leaf-path chart be represented by a pair of tree charts with identical indexes.

So we are interested in addressing the question: When is a subinterval chart a tree chart?

**Definition (exposed clusters) 7.1.21.** *An exposed* $(p+1)$*–cluster or* $(q+1)$*–cluster is an instance of* $p+1$ *or* $q+1$ *consecutive rows respectively in a tree chart with identical valences* $\langle x, y\rangle$ *such that the* $x \geq 1$ *or* $y \geq 1$ *respectively. If the number of rows is clear from the context, then we may simply refer to a cluster or an exposed cluster. Removing an exposed* $n$*–ary cluster from a tree chart will be equivalent to removing an exposed* $n$*–ary caret from a tree; this relationship will become clear soon.*

**Proposition 7.1.22.** *If there exist* $p+1$ *or* $q+1$ *consecutive rows on an interval-chart with identical subinterval weights* $\langle x, y\rangle$ *such that* $x \geq 1$ *or* $y \geq 1$ *respectively, then we can replace these* $p+1$ *or* $q+1$ *consecutive rows with a single row with subinterval weight* $\langle x-1, y\rangle$ *or* $\langle x, y-1\rangle$ *respectively and renumber the subsequent rows on the chart accordingly. The subinterval chart resulting from this process will* *represent the same subdivision of the closed unit interval as the original subinterval chart.*

*Proof.* This follows immediately from Remark 7.1.19 and the fact that a subdivision of the closed unit interval is entirely determined by the lengths of each subinterval. □

**Definition (cluster elimination) 7.1.23.** *We call the process described in Proposition 7.1.22 cluster elimination.*

We can similarly define the process of cluster addition:

**Definition (cluster addition) 7.1.24.** *We can replace a single interval with weight $\langle x, y\rangle$ by $p+1$ intervals with weight $\langle x+1, y\rangle$ or $q+1$ intervals with weight $\langle x, y+1\rangle$. The subinterval chart resulting from this process will represent the same subdivision of the closed unit interval as the original subinterval chart. We call this process cluster addition.*

We say that a $(p+1, q+1)$–ary tree can be represented *exactly* by a subinterval chart if and only if the subinterval weight for each interval in the subinterval chart is exactly the valence of the leaf of the tree with the same index number as the given subinterval.

**Theorem 7.1.25.** *A $(p+1, q+1)$–ary tree can be represented by a given subinterval chart exactly if and only if there exists a sequence of cluster eliminations which can be performed on the subinterval chart such that after the final elimination step in the sequence all interval weights in the chart are $\langle 0, 0\rangle$ and no interval weights in the chart are $\langle 0, 0\rangle$ before this final step.*

*Proof.* First we show that there must exist a sequence of cluster elimination as described in the theorem for any tree chart. We suppose that we have an interval chart which is a tree chart. We note that all trees must have at least one exposed caret. Then we note that any exposed $n$–ary caret (where $n \in \{p+1, q+1\}$) in a given tree will have $n$ leaves $l_{\alpha_1}, ..., l_{\alpha_n}$ such that $\mathbf{v}(l_j) = \mathbf{v}(l_k)$ for all $j, k \in \{\alpha_1, ..., \alpha_n\}$. Therefore, $\mathbf{w}(I_j) = \mathbf{w}(I_k)$ for all $j, k \in \{\alpha_1, ..., \alpha_n\}$ where $I_j, I_k$ represent subintervals present in the subdivision of the unit interval represented by the tree, and we can perform $n$–cluster elimination on rows $\alpha_1$ to $\alpha_n$ of that subinterval chart which corresponds to the tree. In this case, the tree which would result from removing the exposed $n$–ary caret with leaves $l_{\alpha_1}, ..., l_{\alpha_n}$ from the given tree will represent the same subdivision of the closed unit interval as the subinterval chart which results from the cluster elimination of the intervals $I_{\alpha_1}, ..., I_{\alpha_n}$. Therefore, the subinterval chart which results from cluster elimination corresponding to the removal of an exposed caret in a tree is itself a tree chart. It is also clear that the valences of any given leaf remaining in this tree chart after such cluster elimination will be $\langle 0, 0 \rangle$ if and only if the tree which would result from the removal of the exposed caret corresponding to the cluster elimination is the empty tree. Since a tree can always be turned into an empty tree by removing one exposed caret at a time, if a subinterval chart is also a tree chart, then there exists a sequence of cluster elimination steps such that after the final elimination step in the sequence, all valences are $\langle 0, 0 \rangle$ and no valences are $\langle 0, 0 \rangle$ before this final elimination step.

To prove the converse of this statement, we suppose that we have an interval chart which has a cluster elimination sequence as described in this theorem. We consider the given sequence of cluster elimination described in the statement of this theorem and reverse the order of this sequence and replace each step of cluster elimination with its inverse action of cluster addition. In this way we obtain a cluster addition sequence which takes us from a subinterval chart with a single row with weight $\langle 0, 0 \rangle$ to the given subinterval chart under consideration. To avoid confusion with the original cluster elimination sequence, we will call this new cluster addition sequence the *inverse sequence*. Before performing the inverse sequence, we have a subinterval chart with a single row with weight $\langle 0, 0\rangle$; this subinterval chart can be represented by the empty tree. We will call this subinterval chart with a single row with weight $\langle 0, 0\rangle$ the *empty chart*. We note that if a subinterval chart is also a tree chart, then any subinterval chart which results from cluster addition on the original chart will also be a tree chart. Therefore, by induction, any subinterval chart obtained from the empty chart by an inverse sequence whose original cluster elimination sequence satisfies the conditions of this theorem will itself be a tree chart, and the converse is proved.

In other words, once we have a cluster elimination sequence satisfying the conditions of this theorem for a given subinterval chart, we can build the tree which represents the same subdivision of the closed unit interval as the the given subinterval chart by beginning with the empty tree and adding a caret to the leaf in the tree which corresponds to the each cluster addition step in the given subinterval chart's inverse sequence. □

**Theorem 7.1.26.** *A tree-pair diagram representing an element of $F(p+1, q+1)$ exists for a given leaf-path chart if and only if all of the following conditions hold:*

1. *The positive and negative sides of the leaf-path chart have the same number of rows.*

2. *If $n$ denotes the number of rows in the chart, then*

$$\sum_{i=0}^{n-1} \frac{1}{(p+1)^{v^{\pm}_{p+1}(l_i)}(q+1)^{v^{\pm}_{q+1}(l_i)}} = 1$$

3. *For both the positive and negative sides of the chart, there exists at least one cluster elimination sequence which satisfies the conditions of Theorem 7.1.25.*

*Conditions 1 and 2 ensure that the leaf-path chart represents an element of $F(p+1, q+1)$, and Condition 3 ensures that a $(p+1, q+1)$–ary tree-pair diagram exists with the leaf-path valences given in the chart.*

*Proof.* To prove this, we think of each tree-pair diagram as a piecewise-linear orientation-preserving homeomorphism of the closed unit interval with breakpoints in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$ and slopes in $\langle p+1, q+1\rangle$. We note that, given the way we have defined our leaf-path chart, as long as condition 1 of Theorem 7.1.26 is satisfied, a chart will always represent a piecewise-linear orientation-preserving homeomorphism of some closed interval $[0, m]$ for some $m \in \mathbb{N}$. We note again that the length of each closed subinterval $I_i^{\pm}$ of $[0, m]$ represented by the positive or negative side of a row numbered $i$ in the chart will be

$$\frac{1}{(p+1)^{v_{p+1}^{\pm}(l_i)}(q+1)^{v_{q+1}^{\pm}(l_i)}}$$

This means that the first breakpoint will be at

$$\frac{1}{(p+1)^{v_{p+1}^{\pm}(l_0)}(q+1)^{v_{q+1}^{\pm}(l_0)}}$$

and the *ith* breakpoint will be at

$$\sum_{j=0}^{i-1}\frac{1}{(p+1)^{v_{p+1}^{\pm}(l_j)}(q+1)^{v_{q+1}^{\pm}(l_j)}}$$

Since

$$\sum_{j=0}^{i-1}\frac{1}{(p+1)^{v_{p+1}^{\pm}(l_j)}(q+1)^{v_{q+1}^{\pm}(l_j)}}$$

is clearly an element of $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$ for all $i = 0, 1, 2, ..., n-1$, we will clearly have all breakpoints in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$. Since the slope of a given closed subinterval is dependent entirely upon the location of it's endpoints in the domain (negative side of the chart) and range (positive side of the chart), as long as we have all breakpoints and all interval lengths in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$, we will be guaranteed to have all slopes in $\langle p{+}1, q{+}1\rangle$; since the slope of an interval can be calculated by dividing its length in the positive side of the chart by its length in the negative side of the chart, this yields

$$(p+1)^{v^-_{p+1}(l_i)-v^+_{p+1}(l_i)}(q+1)^{v^-_{q+1}(l_i)-v^+_{q+1}(l_i)}$$

for the slope of the interval, which is clearly an element of $\langle p+1, q+1\rangle$. The second condition of Theorem 7.1.26 guarantees that $m = 1$, and therefore that the chart represents a mapping of the closed unit interval onto itself. So it is clear that a leaf-path chart satisfies both of the first two conditions of this theorem, if and only if it represents a piecewise-linear orientation-preserving homeomorphism of the closed unit interval with breakpoints in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$ and slopes in $\langle p{+}1, q{+}1\rangle$. And therefore by Theorem 7.1.25, a leaf-path chart satisfies the third condition of this theorem if and only if at least one tree-pair diagram exists from which the given chart can be derived. □

We introduce a few more definitions now.

**Definition (tree-pair chart and interval-chart pair) 7.1.27.** *If a leaf-path chart can be represented by a tree-pair diagram, we will refer to it as a tree-pair chart. If it cannot be represented by a tree-pair diagram, we will refer to it as an interval-chart pair.*

**Corollary 7.1.28.** *For any two equivalent $(p+1, q+1)$–ary tree-pair diagrams $(T_-, T_+)$ and $(S_-, S_+)$, $(T_-, T_+)$ can be transformed into $(S_-, S_+)$ (or vice versa) through a sequence consisting entirely of transformations from the list below:*

1. *subtree substitution of the type indicated in Figure 7.15*

2. *leaf-path chart cancelation*

3. *leaf-path chart addition*

*Proof.* It is clear that any two $(p+1, q+1)$–ary tree-pair diagrams which can be transformed into one another through a sequence of these transformations will be equivalent. We therefore proceed to prove the converse: that any two equivalent $(p+1, q+1)$–ary tree-pair diagrams can be transformed into one another through a sequence of these transformations.

To prove this, we begin by noting that a $(p+1, q+1)$–ary tree-pair diagram is exactly a visual representation of breakpoints that exist in the homeomorphism of the closed unit interval. Leaves on a tree-pair diagram indicate the subintervals into which the domain and range are divided, and these subintervals are demarcated by their boundary points. Formally we define a *boundary point* as follows: an orientation-preserving piecewise-linear homeomorphism of the closed unit interval can be uniquely represented by $n$ pairs of subintervals of the closed unit interval $([0, a_1], [0, b_1]), ([a_1, a_2], [b_1, b_2]), ..., ([a_{n-1}, 1], [b_{n-1}, 1])$ where if we let $a_0 = b_0 = 0$ and $a_n = b_n = 1$, the map $[a_{i-1}, a_i] \to [b_{i-1}, b_i]$ corresponds to the *ith* linear function (if the functions are numbered from 1 to $n$) in the piecewise-linear function; we then define the ordered pairs $(0, 0), (a_1, b_1), (a_2, b_2), ..., (a_{n-1}, b_{n-1}), (1, 1)$ as the *boundary points* of the homeomorphism. The pairs of points in the domain and range indicated to be boundary points in this way by the trees include those points at which the slope of the homeomorphism changes (including (0,0) and (1,1)) and these points at which the slope changes are referred to specifically as breakpoints; in addition, there may be extra boundary points indicated by the trees at which the slope does not change, but which are needed in order for the trees to remain $(p+1, q+1)$–ary trees. For example, we consider the element of $F(2, 3)$ which can be represented by the tree-pair diagram given in Figure 7.21.

The boundary points given in this tree are:

$$(0, 0), \left(\frac{1}{2}, \frac{1}{2}\right), \left(\frac{3}{4}, \frac{2}{3}\right), \left(\frac{7}{8}, \frac{5}{6}\right), (1, 1)$$

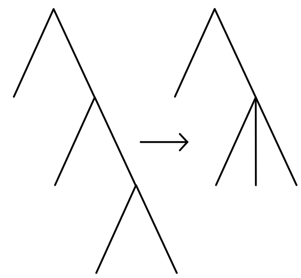

Figure 7.21: A (2,3)–ary tree-pair diagram

However, the point $(\frac{7}{8}, \frac{5}{6})$ is not a breakpoint, as the slope of the homeomorphism does not change at this point. All other points are breakpoints, as the slope does change at those points. So the homeomorphism represented by this tree-pair diagram can be uniquely represented by the list of breakpoints:

$$(0,0), \left(\frac{1}{2}, \frac{1}{2}\right), \left(\frac{3}{4}, \frac{2}{3}\right), (1,1)$$

It is obvious that an element of $F(p+1, q+1)$ seen as a homeomorphism is completely determined by its breakpoints because it is piecewise linear. So all tree-pair diagrams which represent a given homeomorphism will include the breakpoints of that homeomorphism, although each of them may also contain additional boundary points which are unnecessary to determine the homeomorphism which the tree-pair diagram represents. So if two tree-pair diagrams are equivalent, they will have exactly the same breakpoints, but may have different additional boundary points which are not breakpoints.

In fact, because the homeomorphism which represents an element of $F(p+1, q+1)$ is linear in between any two boundary points, to add an additional boundary point which is not a breakpoint in between two breakpoints in the domain of that homeomorphism is possible as long as the boundary point in the range is added at the same relative location between two breakpoints range which correspond to the two breakpoints in the domain between which we have put the new boundary point in the domain; for example, if we add an extra boundary point halfway in between two breakpoints in the domain, we must add its corresponding range boundary point halfway in between the two corresponding breakpoints in the range.

If two tree-pair diagrams are equivalent, they will have the same breakpoints, but they may also each have additional boundary points. Since the process of caret addition is exactly the process of adding extra boundary points which are not breakpoints at the same relative position between breakpoints in the domain and range, we can simply add carets to each tree until all their boundary points are common. This will always be possible because, as we recall from the definition of $F(p+1, q+1)$ and $(p+1, q+1)$–ary tree-pair diagrams - all boundary points (including breakpoints) will be in $\mathbb{Z}[\frac{1}{(p+1)(q+1)}]$. So getting common boundary points in both trees will just be a matter of finding a common denominator. We let $(T_-, T_+)$ and $(S_-, S_+)$ represent the two tree-pair diagrams in question, and we let $(T'_-, T'_+)$ and $(S'_-, S'_+)$ represent these tree-pair diagrams after the described caret addition. The leaves on the tree-pair diagrams will all have identical valences after adding carets in this way, so we will have $T'_\pm \equiv S'_\pm$ (where the $\pm$ is the same in both trees). So by Theorem 7.1.11, we can transform $T_-$ into $S_-$ and $T_+$ into $S_+$ (or vice versa) by a series of substitutions of the kind given in Figure 7.15; we let $(T''_-, T''_+)$ represent $(T'_-, T'_+)$ after these substitutions. But $(T''_-, T''_+) \equiv (S'_-, S'_+)$, so by removing the carets we added to $(S_-, S_+)$ to get $(S'_-, S'_+)$ in the opposite order in which we added them, we can get $(S_-, S_+)$ from $(T''_-, T''_+)$. □

So the closest we can come to a theorem like Theorem 7.1.1 for tree-pair diagrams is the following:

**Theorem 7.1.29.** *If $(T_-, T_+)$ and $(S_-, S_+)$ are $(p+1, q+1)$–ary tree-pair diagrams with identical leaf-path charts, then $(T_-, T_+)$ and $(S_-, S_+)$ are equivalent.*

*If $(T_-, T_+)$ and $(S_-, S_+)$ are equivalent, then either they have identical leaf-path charts, or their leaf-path charts can be transformed into identical leaf-path charts through a series of leaf-path chart additions and/or cancelations.*

*Proof.* This follows immediately from Theorem 7.1.28. □

## 7.2 Finding Minimal Leaf-Path Charts and Tree-Pair Diagrams

In this section we introduce an algorithm for finding minimal or leaf-path charts and tree-pair diagrams for a given element of $F(p+1, q+1)$.

First we define what we mean by a minimal leaf-path chart.

**Definition (minimal leaf-path chart) 7.2.1.** *A minimal leaf-path chart for a given element $x$ of $F(p+1, q+1)$ is a leaf-path chart representative of $x$ (which may or may not be a tree-pair chart) with minimal number of rows with respect to all other leaf-path chart representatives of $x$.*

**Theorem 7.2.2.** *To find a minimal leaf-path chart for a given element $x$ of $F(p+1, q+1)$:*

1. *Define $x$ in terms of the piecewise-linear orientation-preserving homeomorphism of the closed unit interval which it represents.*

2. *Write this homeomorphism as a list of fundamental interval lengths for both the domain (negative) and range (positive): $|I_0^-| \rightarrow |I_0^+|, |I_1^-| \rightarrow |I_1^+|, ..., |I_m^-| \rightarrow |I_m^+|$ where $m$ is the number of interval pairs in the homeomorphism.*

3. *For a given pair of interval lengths $|I_i^-| \rightarrow |I_i^+|$, since both $|I_i^-|$ and $|I_i^+|$ are less than one, they can both be written as proper fractions. Let $n_i^-$ and $n_i^+$ be the numerators of $|I_i^-|$ and $|I_i^+|$ respectively, and let $(p+1)^{a_i}(q+1)^{b_i}$ and*

*$(p+1)^{c_i}(q+1)^{d_i}$ be the denominators of $|I_i^-|$ and $|I_i^+|$ respectively. If $n_i^-$ and $n_i^+$ are both 1, then this pair of intervals $I_i^-$ and $I_i^+$ can be written as a single row in the leaf-path chart with index $i$ and positive and negative valences $\langle a_i, b_i\rangle$ and $\langle c_i, d_i\rangle$ (by Remark 7.1.19). If either $n_i^-$ or $n_i^+$ is not equal to 1, then we let $LCM(n_i^-, n_i^+)$ denote the least common multiple of $n_i^-$ and $n_i^+$. We then subdivide both the positive and negative intervals in to $LCM(n_i^-, n_i^+)$ equally-sized subintervals so that $I_i^-$ now becomes ${I_i^-}' \cup \cdots \cup {I_{i+LCM(n_i^-,n_i^+)-1}^-}'$ where each of the intervals in this union have length $\frac{1}{(p+1)^{a_i}(q+1)^{b_i}n_i^+}$, and $I_i^+$ now becomes ${I_i^+}' \cup \cdots \cup {I_{i+LCM(n_i^-,n_i^+)-1}^+}'$ where each of the intervals in this union have length $\frac{1}{(p+1)^{c_i}(q+1)^{d_i}n_i^-}$. Now we can write $I_i^-$ and $I_i^+$ each as $LCM(n_i^-, n_i^+)$ rows on a leaf-path chart, each with negative valence $\langle a_i, b_i\rangle$ and positive valence $\langle c_i, d_i\rangle$.*

*Once this process is completed for all interval pairs in the homeomorphism, we have a leaf-path chart which represents $x$ (this leaf-path chart may or may not be a tree-pair chart). This leaf-path chart is minimal.*

*Proof.* Because a piecewise-linear orientation-preserving homeomorphism is entirely determined by the lengths of the subintervals in its domain and range, we can determine $x$ completely from the list of subinterval lengths enumerated in step 2. In order for a subdivision of the closed unit interval to be put into leaf-path chart form, the numerator of the length of each subinterval must be 1. The subdivision of a given subinterval pair described in step 3 is the minimal subdivision of $I_i^-$ and $I_i^+$ which will yield a 1 in the numerator of the lengths of all subintervals of $I_i^-$ and $I_i^+$; so this will be the minimal number of subdivisions of $I_i^-$ and $I_i^+$ required to list each subinterval as a row on a leaf-path chart. It only remains to show that $n_i^{\pm} \in \langle p+1, q+1\rangle$ (where $\langle p+1, q+1\rangle$ is the multiplicative group, not a valence).

By definition,

$$S_i = \frac{|I_i^+|}{I_i^-|} = \frac{\frac{n_i^+}{m_i^+}}{\frac{n_i^-}{m_i^-}}$$

where $S_i$ is the slope of the linear map from $I_i^-$ to $I_i^+$ and where $S_i, m_i^\pm \in \langle p + 1, q + 1 \rangle$, and so

$$S_i = \frac{n_i^+}{m_i^+} \cdot \frac{m_i^-}{n_i^-} = \frac{m_i^-}{m_i^+} \cdot \frac{n_i^+}{n_i^-}$$

where $\frac{m_i^-}{m_i^+} \in \langle p+1, q+1 \rangle$, so we must have

$$\frac{n_i^+}{n_i^-} \in \langle p+1, q+1 \rangle$$

which implies that

$$n_i^+, n_i^- \in \langle p+1, q+1 \rangle$$

.

So it is clear from Remark 7.1.19 that this leaf-path chart will represent $x$, and from the definition of least common multiple that such a chart must be minimal.

□

Finding a minimal leaf-path chart will be simpler than finding a minimal tree-pair diagram. If the minimal leaf-path chart obtained from the algorithm given in Theorem 7.2.2 can be represented by a tree-pair diagram, then we have a minimal tree-pair diagram with no extra computation; however, the leaf-path chart obtained from the algorithm given in Theorem 7.2.2 may not be a tree-pair chart. In this case, we would need to continue perform addition on the chart until it is a tree-pair chart, and we would need to be certain that our sequence of additions to turn it into a tree-pair chart were indeed minimal. We can certainly do this by trying all possible combinations, but we can choose which carets to try adding first by looking for certain patterns.

Alternatively, we could begin with a tree-pair diagram for $x$, and perform a sequence of moves described in Theorem 7.1.28 - in order for our set of choices to include all possible tree-pair diagrams with fewer leaves than the current one, we will have to identify those instances in which adding one or more carets to the tree-pair diagram will actually reduce the length because it leads to possible cancelations which were not possible before the carets were added; obviously the resulting number of leaves canceled must be greater than the number of leaves which were added to the tree through the caret addition.

## 7.3 Composition of Tree-Pair Diagrams

Our results from previous sections make it relatively easy to perform composition on pairs of tree-pair diagrams representing elements of $F(p+1,q+1)$. All we need do is use Theorem 7.1.1; to find $xy$ for $x, y \in F(p+1, q+1)$ and $x = (T_-, T_+)$, $y = (S_-, S_+)$, where the operation is composition just as in the case of $F(p+1)$, we need to make $S_+$ identical to $T_-$. This can be accomplished by adding carets to $T_-$ and $S_+$ (and therefore to the leaves with the same index numbers in $T_+$ and $S_-$ respectively) until the valence in all leaf-paths of both $T_-$ and $S_+$ are the same. If we then let $T'_-, T'_+, S'_-, S'_+$ denote $T_-, T_+, S_-, S_+$ respectively after this addition of carets, then the product can be taken to be $(S'_-, T'_+)$ as in the case for $F(p+1)$.

For example, to compose the two (2,3)–ary tree-pair diagrams given in Figure 7.22, we must make the positive tree of $x_0$ equivalent to the negative tree of $y_0^{-1}$. In order to do this, we need only be certain that the leaf-path valences are the same for every leaf on both trees. The leaf numbered 0 on the positive tree of $x_0$ has valence $\langle 2, 0\rangle$, and the leaf numbered 0 on the negative tree of $y_0^{-1}$ has valence $\langle 0, 1\rangle$, so in order to give both leaves the same valence, we must add a 3–ary caret to the leaf numbered 0 in the positive tree of $x_0$ and two 2–ary carets to the leaf

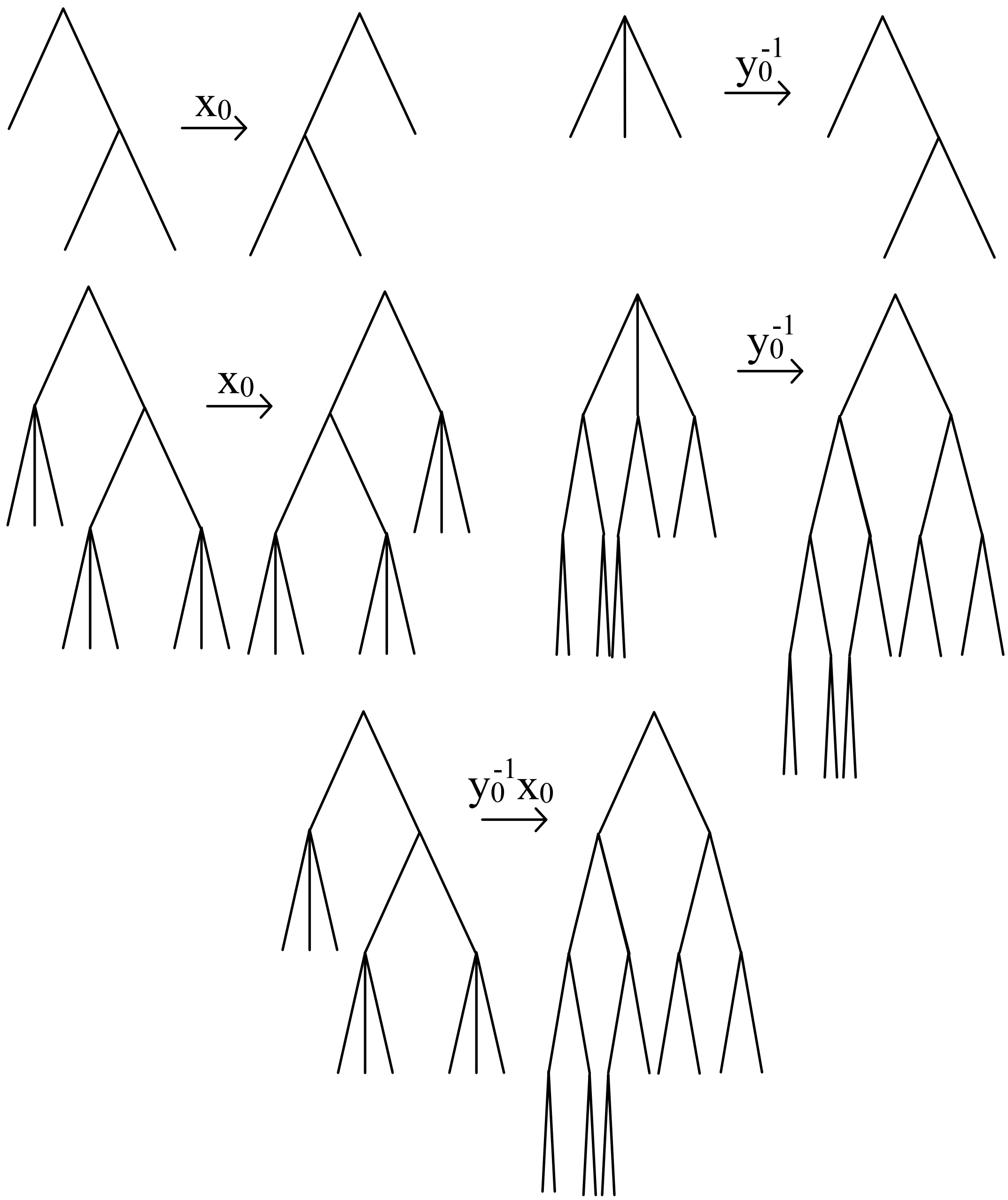


Figure 7.22: Composition of two elements of $F(2,3)$

numbered 0 in the negative tree of $y_0^{-1}$. Now the leaves with index numbers 0 and 1 in the resulting trees will have the same valences. So we consider the leaf with index number 2 in the resulting trees: it has valence $\langle 2, 1 \rangle$, and the leaf numbered 0 on the negative tree of $y_0^{-1}$ has valence $\langle 1, 1 \rangle$, so in order to give both leaves the same valence, we must add a 2–ary caret to the leaf numbered 2 in the negative tree of $y_0^{-1}$. Now the leaves with index numbers 2 and 3 in the resulting trees will have the same valences. So we consider the leaf with index number 4. Continuing in this fashion, we will end up with the tree-pair diagrams given below the initial ones in Figure 7.22. Taking the resulting negative tree of $x_0$ after all carets have been added and taking the resulting positive tree of $y_0^{-1}$ after all carets have been added gives us the tree-pair diagram which represents the product.

**Theorem 7.3.1.** *Let $x, y \in F(p+1, q+1)$ and let $(T_-, T_+)$ and $(S_-, S_+)$ be minimal tree-pair diagram representatives of $x$ and $y$ respectively. Then we add carets to $T_-$ and $S_+$ (and therefore to $T_+$ and $S_-$ respectively) until both trees have the same number of leaves and each of these leaves has identical valences. This process will eventually terminate, and once it terminates, the resulting trees $T'_-$ and $S'_+$ will be equivalent.*

*Proof.* We need to make the negative column of $C(x)$ identical to the positive column of $C(y)$. We will let $\langle a_i, b_i \rangle$ represent the valence of interval $I_i$ in the negative column of $C(x)$, and we let $\langle c_j, d_j \rangle$ represent the valence of interval $J_j$ in the positive column of $C(y)$. Let $M_{p+1} = \max_{I_i \in C(x), J_j \in C(y)} \{a_i, c_j\}$ and $M_{q+1} = \max_{I_i \in C(x), J_j \in C(y)} \{b_i, d_j\}$.

Now we need only perform $(p+1)$–ary and $(q+1)$–ary addition until the negative valence of all rows in $C(x)$ is the same as the positive valence of the corresponding rows in $C(y)$. For the sake of simplicity throughout this proof, we will do this by performing addition on the leaf-path charts until all negative valences in $C(x)$ and all positive valences in $C(y)$ are equal to $\langle M_{p+1}, M_{q+1} \rangle$. As long as this process terminates, it is clear that the resulting trees $T'_-$ and $S'_+$ and their corresponding tree charts will be identical as long as the two tree charts have the same number of rows. So to complete this proof, all we need do is show that this process will always terminate, and that when it does, the number of rows in each chart will be identical.

First we show that this process will terminate. We begin with the first row in each leaf-path chart; the one representing $I_0$ and $J_0$ respectively in $C(x)$ and $C(y)$. We perform repeated $(p+1)$–ary addition on $I_0$ and $J_0$ as needed until both intervals reach a negative and positive $(p+1)$–valence of $M_{p+1}$ respectively. It is clear that $(p+1)$–ary addition will have to be performed $M_{p+1}-a_0$ times on $I_0$ and $M_{p+1}-c_0$ times on $J_0$. Each time we perform $(p+1)$–ary addition on an interval, we will actually be replacing that interval with $p+1$ intervals with identical valences, so we must also perform $(p+1)$–ary addition on these intervals as well until all new intervals produced by the addition have $(p+1)$–valence $M_{p+1}$ as well. So the total number of $(p+1)$–ary additions we will have to perform on $C(x)$ to give $I_0$ (and all the intervals added to the chart by performing $(p+1)$–ary addition on $I_0$) negative $(p+1)$–valences of $M_{p+1}$ will be:

$$(p+1)^{M_{p+1}-a_0-1}$$

and the number of intervals which will be taking the place of the original $I_0$ interval after this $(p+1)$–ary addition is performed will be:

$$(p+1)^{M_{p+1}-a_0}$$

Each of these intervals will now have negative valence $\langle M_{p+1}, b_0\rangle$. Similarly, the total number of $(p+1)$–ary additions we will have to perform on $C(y)$ to give $J_0$ (and all the intervals added to the chart by performing $(p+1)$–ary addition on $J_0$) positive $(p+1)$–valences of $M_{p+1}$ will be:

$$(p+1)^{M_{p+1}-c_0-1}$$

and the number of intervals which will be taking the place of the original $J_0$ interval after this $(p+1)$–ary addition is performed will be:

$$(p+1)^{M_{p+1}-c_0}$$

Each of these intervals will now have positive valence $\langle M_{p+1}, d_0 \rangle$.

Once we have finished performing $(p+1)$–ary addition on these intervals, we will need to perform $(q+1)$–ary addition on each of the $(p+1)^{M_{p+1}-a_0}$ new intervals which have replaced $I_0$ and on each of the $(p+1)^{M_{p+1}-c_0}$ new intervals which have replaced $J_0$, until all of these intervals have negative and positive $(q+1)$–valence $M_{q+1}$ respectively. As in the case for $(p+1)$–ary addition, the total number of $(q+1)$–ary additions we will have to perform on $C(x)$ to give a single interval (and all the intervals added to the chart by performing $(q+1)$–ary addition on that single interval) negative $(q+1)$–valences of $M_{q+1}$ will be:

$$(q+1)^{M_{q+1}-b_0-1}$$

and the number of intervals which will be taking the place of the original single interval in question after this $(q+1)$–ary addition is performed will be:

$$(q+1)^{M_{q+1}-b_0}$$

But if we consider that the process of $(p+1)$–ary addition replaced $I_0$ with $(p+1)^{M_{p+1}-a_0}$ intervals, we have to perform $(q+1)$–ary addition this many times on each of these $(p+1)^{M_{p+1}-a_0}$ intervals, so that the total number of $(q+1)$–ary additions we will have to perform on $C(x)$ to give all the intervals which have replaced $I_0$ negative $(q+1)$–valences of $M_{q+1}$ will be:

$$(p+1)^{M_{p+1}-a_0-1}(q+1)^{M_{q+1}-b_0-1}$$

and the number of intervals which will be taking the place of $I_0$ after $(p+1)$–ary and $(q+1)$–ary addition is performed will be:

$$(p+1)^{M_{p+1}-a_0}(q+1)^{M_{q+1}-b_0}$$

Each of these intervals will now have negative valence $\langle M_{p+1}, M_{q+1}\rangle$. Similarly, the total number of $(q+1)$–ary additions we will have to perform on $C(y)$ to give all the intervals which have replaced $J_0$ positive $(q+1)$–valences of $M_{q+1}$ will be:

$$(p+1)^{M_{p+1}-c_0-1}(q+1)^{M_{q+1}-d_0-1}$$

and the number of intervals which will be taking the place of $J_0$ after $(p+1)$–ary and $(q+1)$–ary addition is performed will be:

$$(p+1)^{M_{p+1}-c_0}(q+1)^{M_{q+1}-d_0}$$

Each of these intervals will now have positive valence $\langle M_{p+1}, M_{q+1}\rangle$.

If we let $L_x$ and $L_y$ denote the number of leaves in the minimal-tree-pair diagram for $x$ and $y$ respectively, then we will have to repeat the above process which we have just described for $I_0$ and $J_0$ for intervals $I_1, ..., I_{L_x}$ and for intervals $J_1, ..., J_{L_y}$ as well. So, the total number of intervals which will be in the leaf-path chart for $x$ after all necessary $(p+1)$–ary and $(q+1)$–ary addition is performed will be:

$$\sum_{i=0}^{L_x}(p+1)^{M_{p+1}-a_i}(q+1)^{M_{q+1}-b_i}$$

Each of these intervals will now have negative valence $\langle M_{p+1}, M_{q+1}\rangle$.

Similarly, the total number of intervals which will be in the leaf-path chart for $y$ after all necessary $(p+1)$–ary and $(q+1)$–ary addition is performed will be:

$$\sum_{i=0}^{L_y}(p+1)^{M_{p+1}-c_i}(q+1)^{M_{q+1}-d_i}$$

Each of these intervals will now have positive valence $\langle M_{p+1}, M_{q+1}\rangle$.

Since both of these values are clearly finite, we can be certain that this process of composition through leaf-path chart addition always terminates. Now we need to prove that the number of rows in the resulting leaf-path chart for $x$ and the resulting leaf-path chart for $y$ are equal. To do this we consider the following equation for the number of rows in the resulting leaf-path chart for $x$:

$$\sum_{i=0}^{L_x}(p+1)^{M_{p+1}-a_i}(q+1)^{M_{q+1}-b_i}$$

This can be rewritten:

$$(p+1)^{M_{p+1}}(q+1)^{M_{q+1}}\sum_{i=0}^{L_x}\frac{1}{(p+1)^{a_i}(q+1)^{b_i}}$$

where

$$\sum_{i=0}^{L_x}\frac{1}{(p+1)^{a_i}(q+1)^{b_i}}$$

is just the sum of the length of all the intervals in the leaf-path chart, which is by definition equal to 1. So this yields:

$$\sum_{i=0}^{L_x}(p+1)^{M_{p+1}-a_i}(q+1)^{M_{q+1}-b_i} = (p+1)^{M_{p+1}}(q+1)^{M_{q+1}}$$

Similarly, for the number of rows in the resulting leaf-path chart for $y$:

$$\sum_{i=0}^{L_y}(p+1)^{M_{p+1}-c_i}(q+1)^{M_{q+1}-d_i} = (p+1)^{M_{p+1}}(q+1)^{M_{q+1}}$$

So the total number of rows in each resulting leaf-path chart will be the same, and therefore composition can take place. □

We should note that in practice it will not be necessary to perform leaf-path chart addition until every leaf in $C(x)$ and $C(y)$ has negative and positive valences of $\langle M_{p+1}, M_{q+1}\rangle$. In principle, most of the time the total number of additions required will be much smaller, because we do not need the valence of every row to be equal - we only need the valences of rows sharing the same index to be equal, so we can halt our addition once this is result is obtained, which will generally happen in far fewer steps than those required to make all valences equal to $\langle M_{p+1}, M_{q+1}\rangle$. For example, in the composition given in Figure 7.22, this process clearly halted before all valences were equal.

# Chapter 8

# Actions of Generators on Tree-Pair-Diagrams

## 8.1 Presentations of $F(2,3)$

For the duration of this paper, we will restrict our discussion to the group $F(2,3)$ specifically.

In [17], Melanie Stein gave infinite and finite presentations for $F(2,3)$, which we will use throughout this paper. We will refer to these as the standard infinite and finite presentations.

The infinite presentation has the following generators:

$$\{x_0, x_1, ..., y_0, y_1, ..., z_0, z_1, ...\}$$

where the $x_i$ generators are simply the same generators as in $F(2)$. We can see the tree-pair diagrams for these generators in Figure 8.1. (The generators we refer to as $z_i$ are the same as those which Stein refers to as $y_i^+$ in [17]).

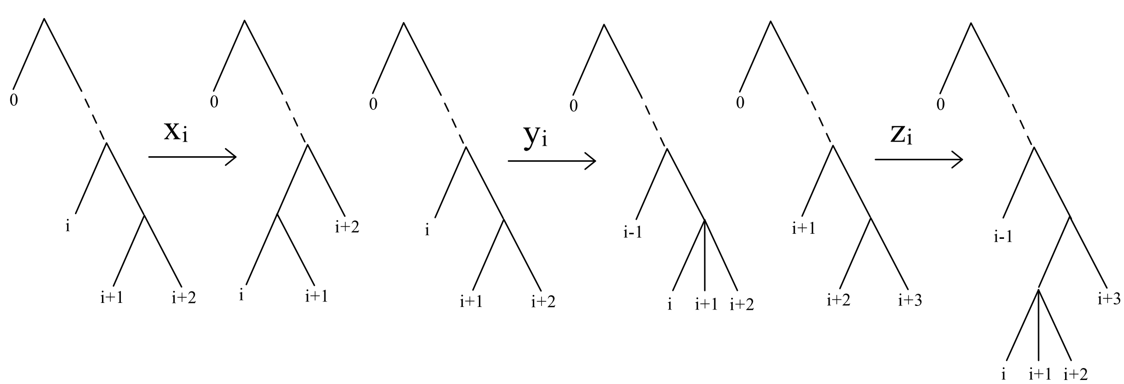


Figure 8.1: The generators for the standard infinite presentation of $F(2,3)$

The relators for this presentation are:

1. $x_j x_i = x_i x_{j+1}$

2. $y_j x_i = x_i y_{j+1}$

3. $z_j x_i = x_i z_{j+1}$

4. $x_j z_i = z_i x_{j+2}$

5. $y_j z_i = z_i y_{j+2}$

6. $z_j z_i = z_i z_{j+2}$

for $i < j$ and

1. $y_{i+1} z_i = y_i x_{i+1} x_i$

2. $x_i z_{i+1} z_i = z_i x_{i+2} x_{i+1} x_i$

for all $i$.

Essentially, the first set of relators are the same relators we encounter in $F(p+1)$, and the second set of relators are the ones that provide for the substitution of the type we see in Figure 7.16.

Clearly this infinite presentation is not minimal. Using the relator $y_{i+1} z_i = y_i x_{i+1} x_i$ to replace all instances of $z_i$ with an expression in the $x_i$ and $y_i$ generators, we could reduce this infinite presentation to one containing only $x_i$ and $y_i$ generators; however, because the presence of the $z_i$ generators will be essential in determining the normal form, we will employ this infinite presentation instead.

The finite presentation for $F(2, 3)$ introduced by Stein in [17], and which we will refer to as the standard finite presentation has the generators:

$$\{x_0, x_1, y_0, y_1\}$$

which we can see depicted as tree-pair diagrams in Figure 8.2.

The relators for this presentation are:

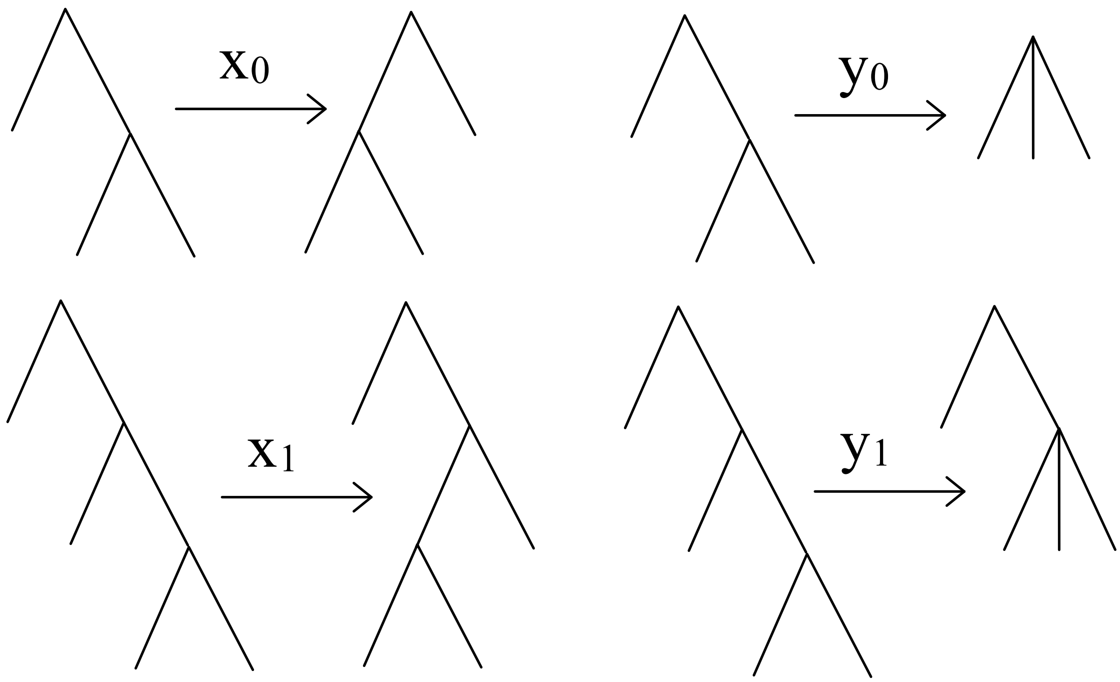


Figure 8.2: The generators for the standard finite presentation of $F(2, 3)$

1. $x_2x_0 = x_0x_3$
2. $x_3x_1 = x_1x_4$
3. $y_2x_0 = x_0y_3$
4. $y_3x_1 = x_1y_4$
5. $x_1z_0 = z_0x_3$
6. $x_2z_1 = z_1x_4$
7. $y_1z_0 = z_0y_3$
8. $y_2z_1 = z_1y_4$
9. $x_0z_1z_0 = z_0x_2x_1x_0$

10. $x_1 z_2 z_1 = z_1 x_3 x_2 x_1$

where $x_3 = x_1^{-1} x_2 x_1$, $x_4 = x_2^{-1} x_3 x_2$, $y_3 = x_1^{-1} y_2 x_1$, $y_4 = x_2^{-1} y_3 x_2$, $z_0 = y_1^{-1} y_0 x_1 x_0$, $z_1 = y_2^{-1} y_1 x_2 x_1$, and $z_2 = y_3^{-1} y_2 x_3 x_2$.

The standard infinite presentation can be derived from the standard finite presentation by induction.

## 8.2 Actions of Finite Generators on (2, 3)–ary Tree-Pair Diagrams

In order to multiply an arbitrary word $w \in F(2,3)$ by a generator in the standard finite presentation, we may need to add carets to the tree-pair diagram for $w$. We let $(T_-, T_+)$ denote the tree-pair diagram for $w$ once all carets have been added so that multiplication can take place. Just as in the case for $F(p+1)$, once all necessary carets have been added, multiplying on the right by a generator will only affect the negative tree. We can see the action of each generator on $T_-$ in Figure 8.3.

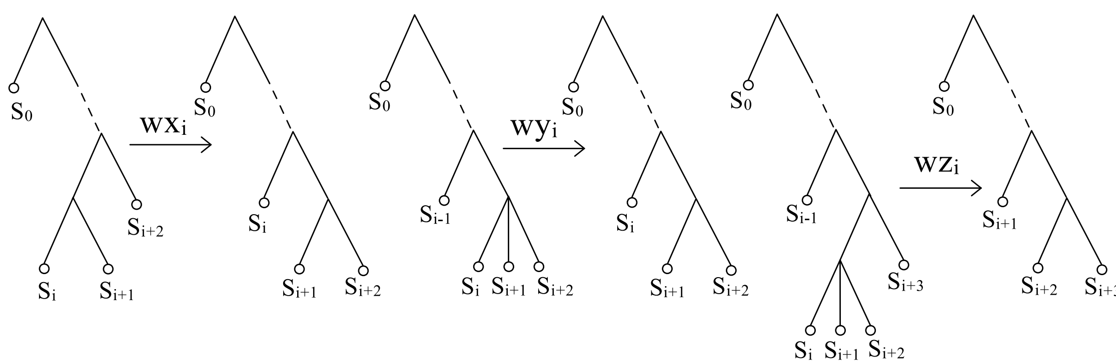


Figure 8.3: The action of the generators on the negative tree of an arbitrary element of $F(2,3)$ once the tree-pair diagram for that element has had all necessary carets added so that multiplication can take place

If $T_-$ cannot be written in the form of the tree on the left of the depiction of the action of $g$ (where $g \in \{x_0, ..., y_0, ..., z_0, ...\}$) on $w$ given in Figure 8.3, then carets must be added to $T_-$ (and by extension $T_+$) before we can find the minimal tree-pair diagram of $wg$.

# Chapter 9

## A Normal Form for Elements of $F(2,3)$

We introduce a normal form in the infinite generating set for elements of $F(2,3)$. Any time we refer to a normal form in this section, we mean with respect to the standard infinite generating set.

Also, for the duration of this paper, the notation $L(T)$ or $L(T_-,T_+)$ or $L(w)$ denotes the number of leaves in the tree $T$ or in the tree-pair diagram $(T_-,T_+)$ or in the minimal tree-pair diagram representative for $w$ respectively.

**Lemma 9.1.** *If every word $w$ in our group can be uniquely factored into clearly identifiable subwords $w_1$,$w_2$,...,$w_n$ so that the choice of each subword and the number of subwords in the factorization is unique for each $w$, then if we can define a normal form for each subword $NF(w_i)$, we can define a normal form for $w$ by:*

$$NF(w) = NF(w_1)NF(w_2)\cdots NF(w_n)$$

For example, the normal form for Thompson's group $F(p+1)$ involves factoring $w$ into a positive and negative word. If we let $x_+$ (and likewise $x_-$) be the element whose minimal tree-pair diagram representative has a tree composed entirely of right carets as the negative (positive) tree and whose positive (negative) tree is the same as the positive (negative) tree for $x$, then we can always factor $x$ uniquely as: $x = x_+x_-^{-1}$. This fact was used to find the normal form for elements of $F(p+1)$ in [4], and we shall use it here to find the normal form for elements of $F(2,3)$.

**Definition (positive word) 9.1.** *A positive word in $F(2,3)$ is a word whose minimal tree-pair diagram representative has a negative tree which consists entirely of right 2–ary carets and therefore its normal form consists entirely of generators with powers which are positive. Similarly, a negative word is a word whose minimal tree-pair diagram representative has a positive tree which consists entirely of right 2–ary carets and therefore its normal form consists entirely of generators with powers which are negative.*

Throughout this section, we will continue the use of this notation $x_+$ and $x_-$, and we will also use the notation $R^n$ to represent the tree composed entirely of right 2–ary carets, containing $n$ carets total.

Throughout this section, we will also use the notation $NF(x)$ to denote the normal form of $x$ in $F(2,3)$.

**Corollary 9.2.** *If a normal form exists for positive elements of $F(2,3)$, then a normal form exists for any element $w$ of $F(2,3)$. This normal form can be obtained by writing*

$$NF(w) = NF(w_+)NF(w_-^{-1})$$

*Proof.* Since any word $w$ can always be factored uniquely into $w_+w_-^{-1}$, and by definition $w_-$ and $w_+$ are positive elements, this follows immediately from Lemma 9.1. □

So we seek a normal form for positive words. to begin our search, we first write a minimal tree-pair diagram representative for an arbitrary positive word in a general way.

**Remark 9.3.** *Any positive element $w$ of $F(2,3)$ has a minimal tree-pair diagram representative which can be written in the form given by Figure 9.1, and it is always possible to uniquely identify each of the subtrees $A_0, B_0, ..., A_m, B_m, C_1, ..., C_n$.*

*Proof.* This is obvious; all we have done is written a general $F(2,3)$ tree-pair diagram for a positive element by symbolizing any middle carets and their children by using subtree labels. □

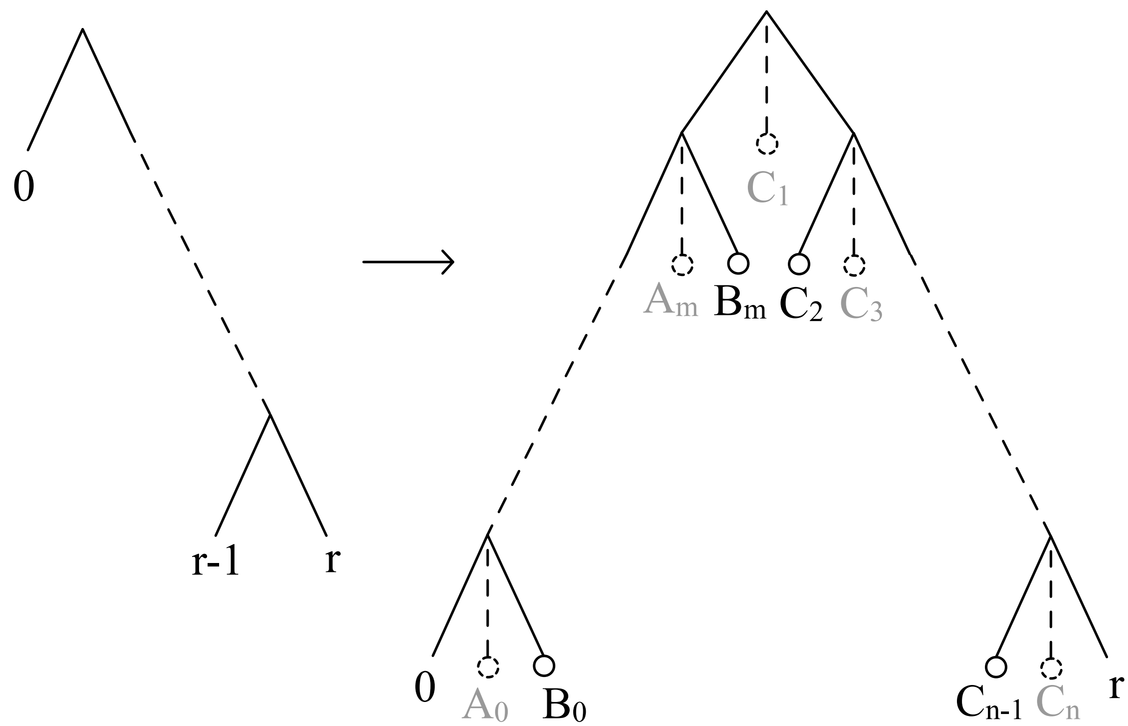


Figure 9.1: General Form of Positive Element in $F(2,3)$. Dotted middle edge and subtrees indicate that each caret in the positive tree may be 2–ary or 3–ary.

We build our normal form in parts; first we show that we can obtain the normal form of a positive word by looking at the right and left "halves" of the positive tree in the minimal tree-pair diagram representative and finding the normal form for each of these subtrees separately. Or, more formally:

**Corollary (to Lemma 9.1) 9.4.** *If a normal form exists for words $w_l$ and $w_r$ which have minimal tree-pair diagram representatives of the form given in Figures 9.2 and 9.3 respectively, then a normal form exists for the word $W_{lr}$ which has minimal tree-pair diagram representative of the form $w_{lr}$ given in Figure 9.4.*

$$NF(w_{lr}) = NF(w_l)NF(w_r)$$

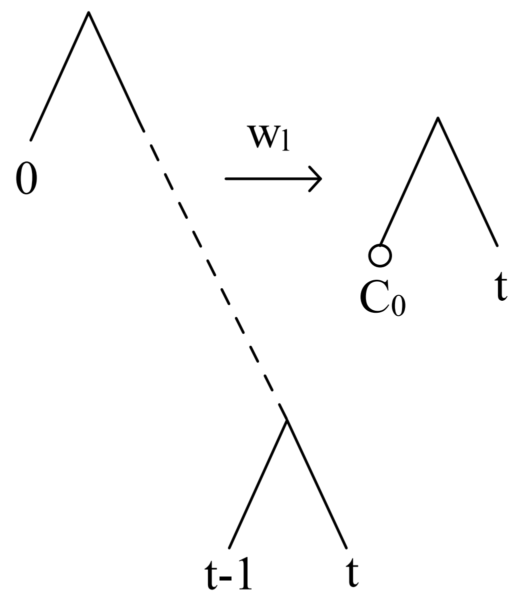


Figure 9.2: Minimal tree-pair diagram representative of $w_l$ in $F(2,3)$. Here $L(C_0) = t$.

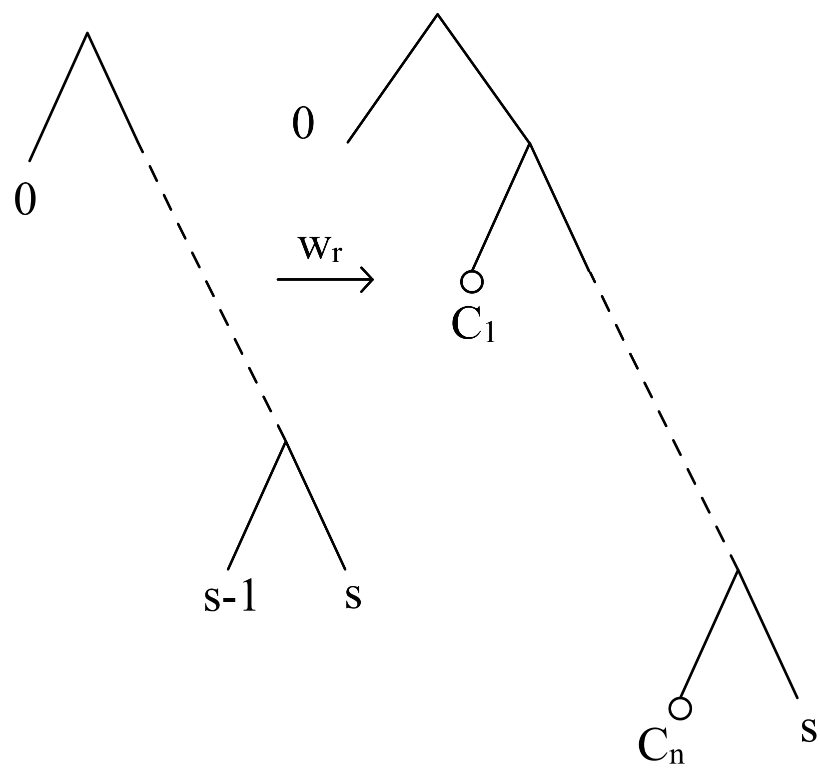


Figure 9.3: Minimal tree-pair diagram representative of $w_r$ in $F(2,3)$. Here $L(C_1) + \cdots + L(C_n) = s - 1$.

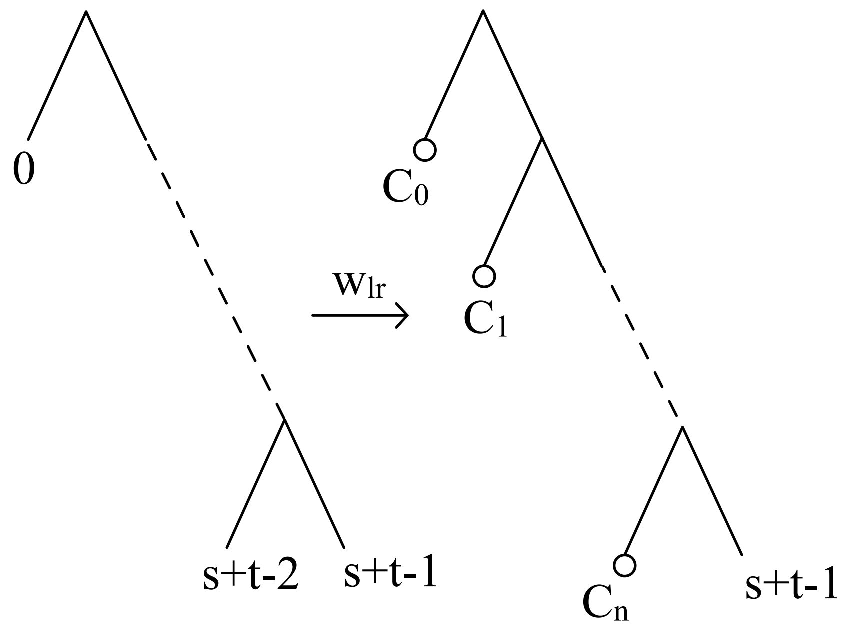


Figure 9.4: Minimal tree-pair diagram representative of $w_{lr}$ in $F(2,3)$

*Proof.* To prove this, we need only observe that $w_{lr} = w_l w_r$, and that $w_l$ and $w_r$ can be uniquely determined for any word $w_{lr}$ if the subtrees $C_0, ..., C_n$ of the minimal tree-pair diagram representative for $w_{lr}$ can all be uniquely determined. The Corollary then follows directly from Lemma 9.1. □

Now we seek a normal form for all elements whose positive tree in their minimal tree-pair diagram representative is only a "left half," i.e. one which has the form given in Figure 9.7, and we show that this normal form only depends on the subtrees which contain only carets which are descended from carets on the left side of the positive tree.

**Corollary (to Lemma 9.1) 9.5.** *We let $w_i$, $\gamma_0^i w_i$,, and $W_i$ be the elements of $F(2,3)$ with minimal tree-pair diagram representative given in Figures 9.5, 9.6, and 9.7 respectively. If a normal form exists for $w_i$, then the normal form for $W_m$*

*is:*

$$NF(W_m) = \prod_{i=m}^{0} \gamma_0^i NF(w_i)$$

*where* $\gamma_0^i = x_0$ *if and only if* $\wedge_{\mathcal{C}_i}$ *in Figure 9.7 is 2–ary and* $\gamma_0^i = z_0$ *if and only if* $\wedge_{\mathcal{C}_i}$ *in Figure 9.7 is 3–ary.*

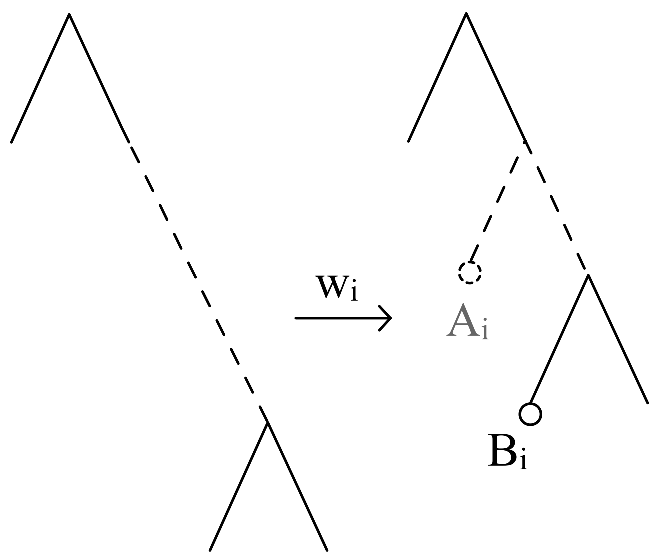


Figure 9.5: Minimal tree-pair diagram representative of $w_i$ in $F(2,3)$. Dotted edges and subtree indicate that the first right caret in the positive tree and the subtree $A_i$ may not exist.

*Proof.* To prove this, we show that

$$W_m = \prod_{m}^{i+1} \gamma_0^i w_i$$

where $\gamma_0^i$ and $w_i$ are as defined in the theorem, and we show that this factorization is unique if the tree-pair diagram representative for $W_m$ is unique. We will prove this by induction. We suppose that a normal form exists for any element of $F(2,3)$ with minimal tree-pair diagram representative of the form given in Figure 9.5. We also note that if we multiply an element with minimal tree-pair diagram representative of the form given in Figure 9.5 by $x_0$ if the dotted caret in the figure does not exist

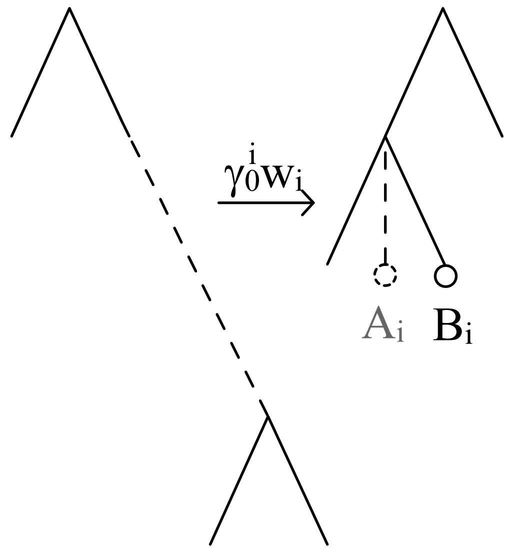


Figure 9.6: Minimal tree-pair diagram representative of $\gamma_0^i w_i$ in $F(2,3)$. Dotted middle edge and subtree indicate that the left child of the root $\wedge_0$ in the positive tree may be 2–ary or 3–ary, so that the subtree $A_i$ may or may not exist; $\wedge_0$ is 3–ary if and only if $\gamma_0^i = x_0$ and the subtree $A_i$ existed in the minimal tree-pair diagram for $w_i$ (Figure 9.5).

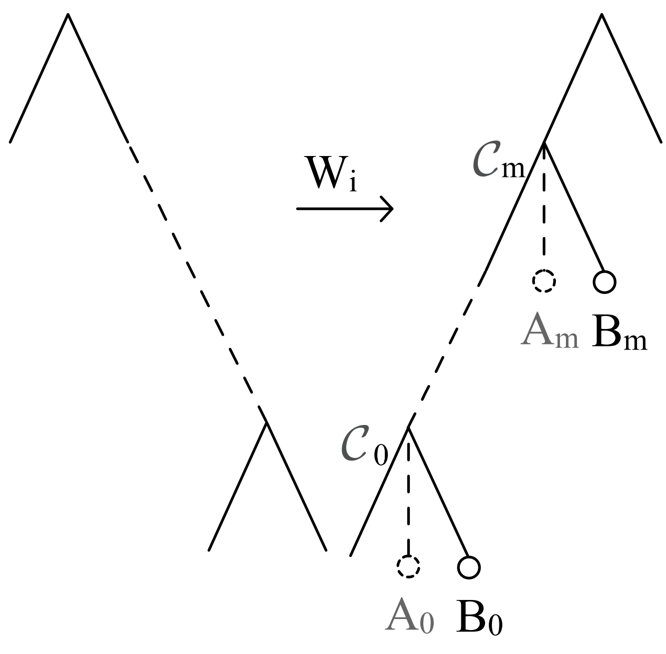


Figure 9.7: Minimal tree-pair diagram representative of $\gamma_0^i w_i$ in $F(2,3)$. Dotted middle edges and subtrees indicate that each of the carets $\wedge_{\mathcal{C}_0}, ..., \wedge_{\mathcal{C}_m}$ in the positive tree may be 2–ary or 3–ary, so that each of the subtrees $A_0, ...A_m$ may or may not exist.

and by $z_0$ otherwise, then the product will have minimal tree-pair diagram of the form given in Figure 9.6.

We begin the induction by noting that the tree-pair diagram given in Figure 9.7 for $W_1$ will be the minimal tree-pair diagram representative of the product $\gamma_0^1 w_1 \gamma_0^0 w_0$ where $w_0$ and $w_1$ are words with minimal tree-pair diagram representatives of the form given by Figure 9.5 and 9.6 respectively. We note that this factorization of $W_1$ is unique as long as our choice of subtrees in 9.5 is unique. Now we suppose that $W_i = \gamma_0^i w_i \cdots \gamma_0^0 w_0$ where $W_i$ and $\gamma_0^j w_j$ (for $j = 1, ..., i$) are as in Figures 9.7 (when $m = i$) and 9.6 respectively. It is then easy to see that $\gamma_0^{i+1} w_{i+1} W_i$ has minimal tree-pair diagram representative of the form given in Figure 9.7 when $m = i + 1$. So, by induction, any word $W_m$ with minimal tree-pair diagram of the form given in Figure 9.7 can be written as the product:

$$W_m = \prod_{i=m}^{0} \gamma_0^i w_i$$

where

$$\gamma_0^i = \begin{cases} x_0, & \text{if } \wedge_{\mathcal{C}_i} \text{ in Figure 9.7 is 2–ary} \\ z_0, & \text{if } \wedge_{\mathcal{C}_i} \text{ in Figure 9.7 is 3–ary} \end{cases}$$

and where this product is unique if the minimal tree-pair diagram representative given in Figure 9.7 is unique. □

Now we seek a normal form for all elements whose positive tree in the minimal tree-pair diagram representative is only a "right half," i.e. has the form given in Figure 9.3, and we show that this normal form only depends on the subtrees which consist entirely of carets which are descended from carets on the right side of the positive tree and the type of the carets on the right side of the positive tree.

**Corollary (to Lemma 9.1) 9.6.** *If a normal form exists for elements with minimal tree-pair diagram representatives of the form given in Figure 9.4, then a* *normal form exists for elements whose minimal tree-pair diagram representative is of the form given in Figure 9.8.*

*In fact, if $w$ is of the form given in Figure 9.8*

$$NF(w) = y_{\beta_1} \cdots y_{\beta_u} NF(w_{lr})$$

*where $\beta_i < \beta_{i+1} + 1$ for all $i \in \{1, ..., u-1\}$; $k \in \{\beta_1, ..., \beta_u\}$ if and only if the leaf with index number $k$ (in the subtree of Figure 9.1 consisting entirely of right carets and the root caret) is the leftmost child of a 3–ary caret.*

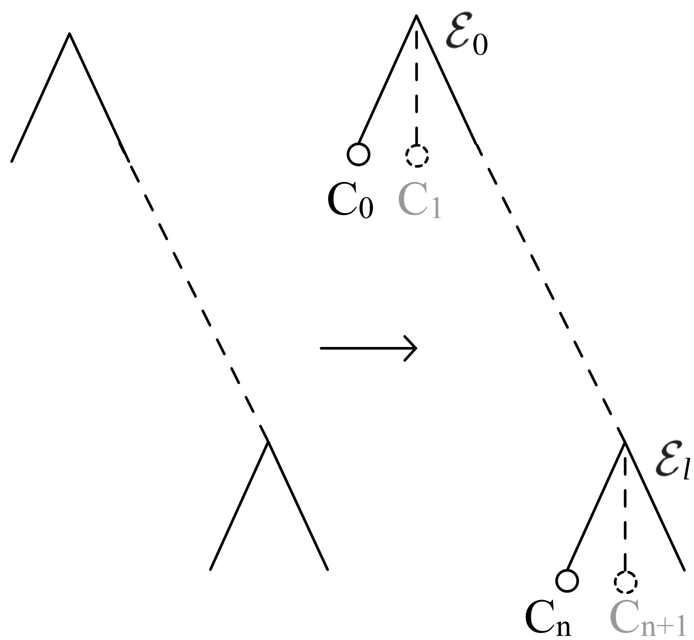


Figure 9.8: Minimal tree-pair diagram representative of a positive element in $F(2,3)$. Dotted middle edges and subtrees indicate that each of the carets $\wedge_{\mathcal{E}_0}, ..., \wedge_{\mathcal{E}_l}$ in the positive tree may be 2–ary or 3–ary, so that each of the subtrees $C_1, C_3, C_5, ...C_{n+1}$ may or may not exist.

*Proof.* We begin by observing that if we have $u_i v_i \in F(2,3)$ so that the minimal tree-pair diagram representatives of $u_i$ and $v_i$ can be represented by Figures 9.9 and 9.10 respectively, then $v_i = y_i u_i$.

We let $u$ denote the number of 3–ary carets on the right side of the positive tree in the minimal tree-pair diagram representative of $w$. We let $N_3(\wedge_{\mathcal{E}_i})$ denote the number of 3–ary carets on the leaf path from the parent node of $\wedge_{\mathcal{E}_i}$ to the root node in the positive tree of the minimal tree-pair diagram representative of $w$. Then $\mathcal{E}_i = \mathcal{C}_i - N_3(\wedge_{\mathcal{E}_i})$. Let $S = \{x| \wedge_{\mathcal{E}_x}$ is 3–ary$\}$ and denote the elements of $S$ by $a_1, a_2, ..., a_u$ where $a_i > a_j$ if and only if $i > j$.

Consider $w_{lr}$ and $w$. The caret $\wedge_{\mathcal{E}_{a_u}}$ is the last 3–ary caret on the right side of the positive tree of the minimal tree-pair diagram representative of $w$. We can see that the minimal tree-pair diagram representative of $w_{lr}$ is of the form given in Figure 9.9 if we let $i = \mathcal{D}_{a_u}$. We take $u_{\mathcal{D}_{a_u}} = w_{lr}$ and get

$$v_{\mathcal{D}_{a_u}} = y_{\mathcal{D}_{a_u}} u_{\mathcal{D}_{a_u}} = y_{\mathcal{D}_{a_{u-1}}} w_{lr}$$

Now we can see that the minimal tree-pair diagram representative of $v_{\mathcal{D}_{a_{u-1}}}$ is of the form given in Figure 9.9 if we let $i = \mathcal{D}_{a_{u-1}}$. So we let $u_{\mathcal{D}_{a_{u-1}}} = v_{\mathcal{D}_{a_u}}$ and take

$$v_{\mathcal{D}_{a_{u-1}}} = y_{\mathcal{D}_{a_{u-1}}} u_{\mathcal{D}_{a_{u-1}}}$$

Now we proceed to the general induction step. We suppose that

$$u_{\mathcal{D}_{a_j}} = y_{\mathcal{D}_{a_{j+1}}} y_{\mathcal{D}_{a_{j+2}}} \cdots y_{\mathcal{D}_{a_u}} w_{lr}$$

Then

$$v_{\mathcal{D}_{a_j}} = y_{\mathcal{D}_{a_j}} u_{\mathcal{D}_{a_j}}$$

Then we can see that the minimal tree-pair diagram representative of $v_{\mathcal{D}_{a_j}}$ is of the form given in Figure 9.9 if we let $i = \mathcal{D}_{a_{j-1}}$. So we define

$$\begin{aligned} u_{\mathcal{D}_{a_{j-1}}} &= v_{\mathcal{D}_{a_j}} \\ &= y_{\mathcal{D}_{a_j}} u_{\mathcal{D}_{a_j}} \\ &= y_{\mathcal{D}_{a_j}} y_{\mathcal{D}_{a_{j+1}}} \cdots y_{\mathcal{D}_{a_u}} w_{lr} \end{aligned}$$

So by induction,

$$w = y_{\beta_1} \cdots y_{\beta_u} w_{lr}$$

where $\beta_i < \beta_{i+1} + 1$ for all $i \in \{1, ..., u-1\}$ ensures that any expression of this form will be representable by the minimal tree-pair diagram given in Figure 9.8. This factorization is unique if the choice of subtrees and caret type in the positive tree of Figure 9.8 is unique.

$\square$

Now we seek a general algorithm to find the normal form of an element with minimal tree-pair diagram representative of the form given in Figure 9.3 because our normal form of both the "left and right halves" of the positive tree in the minimal tree-pair diagram representative of a positive element depends upon having the normal form of an element with minimal tree-pair diagram of the form given in Figure 9.3. Before we can do this, we will need to develop a concept analogous to the idea of leaf exponents, which were used to obtain the normal form for elements of $F(p+1)$ from their minimal tree-pair diagram representatives.

**Definition (leaf exponent matrix) 9.7.** *We consider the leaf path for $l_i$ in a tree. Now we define a subpath of the leaf path which we will call the exponent path: the subtree consisting of all carets $\wedge_j$ on the leaf path of $l_i$ such that $\wedge_j$ has its left edge on the directed edge from the root node to the leaf and $\wedge_j$ is not a right caret or the root caret. The number of carets on the exponent path is the leaf exponent of $l_i$, just as in the case of $F(p+1)$.*

*The leaf exponent matrix generalizes the leaf exponent in the case of $F(2,3)$ to indicate the order of 2–ary and 3–ary carets on this path. The exponent path can be broken into subpaths which contain only one valence of carets and which contain all consecutive carets of that valence on that part of the exponent path. If we let $\lambda_{i,1}$ denote the subpath closest to the root and if we let $\lambda_{i,j}$ denote the adjacent subpath*

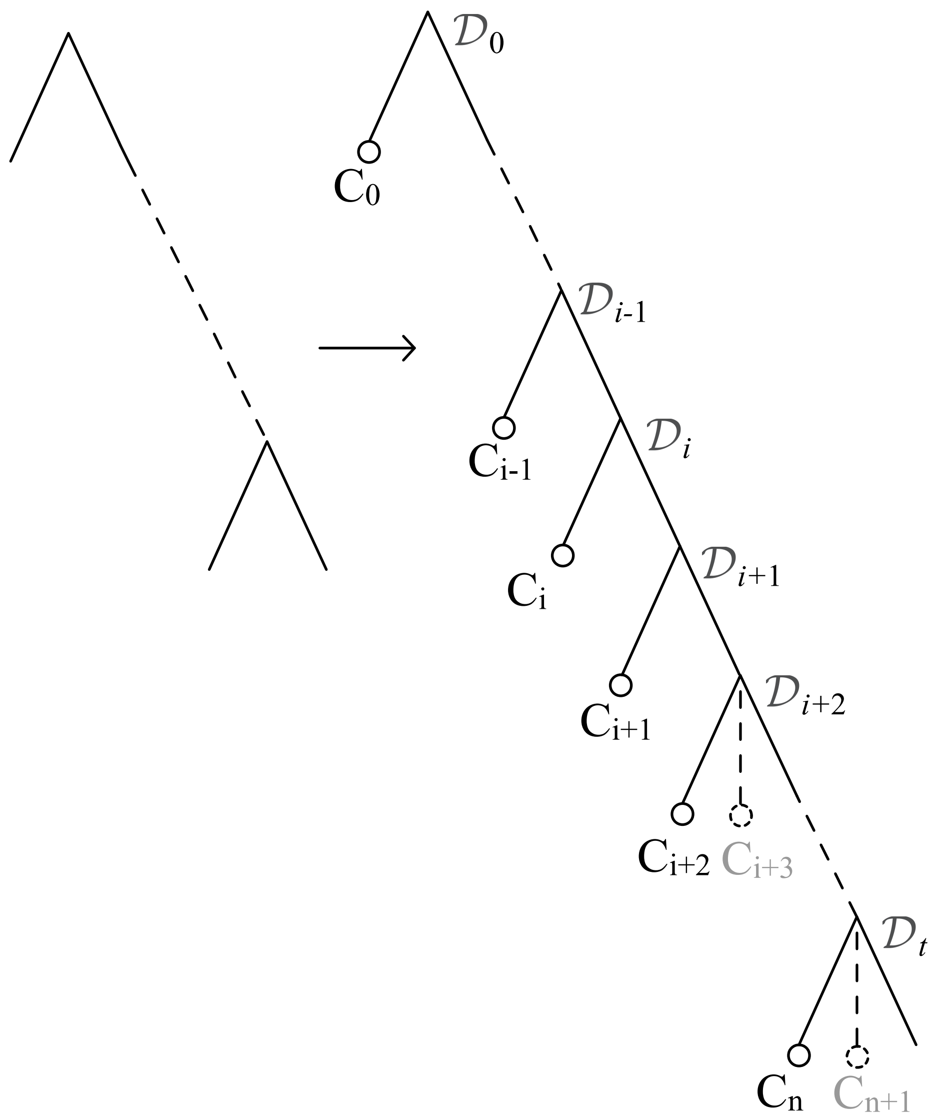


Figure 9.9: Minimal tree-pair diagram representative of a positive element in $F(2,3)$ where the root and the first $i+1$ right carets in the positive tree are 2–ary. Dotted middle edges and subtrees indicate that each of the carets $\wedge_{\mathcal{D}_{i+2}}, ..., \wedge_{\mathcal{D}_t}$ in the positive tree may be 2–ary or 3–ary, so that each of the subtrees $C_{i+3}, C_{i+5}, C_{i+7}, ...C_{n+1}$ may or may not exist.

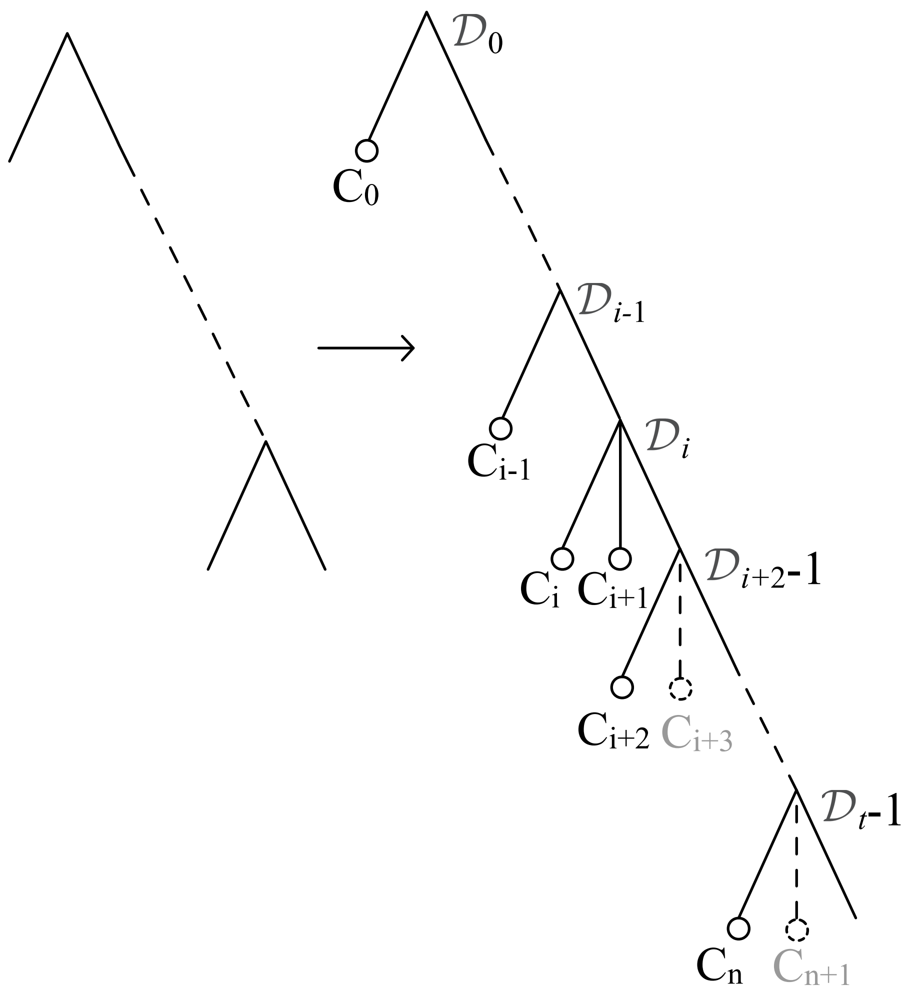


Figure 9.10: Minimal tree-pair diagram representative of a positive element in $F(2,3)$ where the root and the first $i-1$ right carets in the positive tree are 2–ary, the $ith$ right caret is 3–ary. Dotted middle edges and subtrees indicate that each of the carets $\wedge_{\mathcal{D}_{i+2}-1}, ..., \wedge_{\mathcal{D}_t-1}$ in the positive tree may be 2–ary or 3–ary, so that each of the subtrees $C_{i+3}, C_{i+5}, C_{i+7}, ...C_{n+1}$ may or may not exist.

*to $\lambda_{i,j-1}$ which is closer to the leaf $l_i$, then the leaf exponent matrix for $l_i$ is*

$$E_i = \begin{pmatrix} \epsilon_{i,1} & \cdots & \epsilon_{i,m} \\ d_{i,1} & \cdots & d_{i,m} \end{pmatrix}$$

*where*

$$\epsilon_{i,j} = \begin{cases} 2 \text{ if carets in the subpath } \lambda_{i,j} \text{ are 2–ary} \\ 3 \text{ if carets in the subpath } \lambda_{i,j} \text{ are 3–ary} \end{cases}$$

*and where $d_{i,j}$ is the number of carets on the subpath $\lambda_{i,j}$. Clearly the leaf exponent of $l_i$ will be equal to*

$$\sum_{j=1}^{m} d_{i,j}$$

**Theorem 9.8.** *Any element $x$ of $F(2,3)$ whose minimal tree-pair diagram representative can be written in the form given in Figure 9.3 has normal form:*

$$NF(x) = (\gamma^1_{\alpha_1})^{e_1} \cdots (\gamma^s_{\alpha_s})^{e_s}$$

*where $\gamma^i_{\alpha_i}$ is either $x_{\alpha_i}$ or $z_{\alpha_i}$, and $\alpha_i \leq \alpha_j$ whenever $i < j$, and where a subword*

$$(\gamma^j_{\alpha_j})^{e_j} \cdots (\gamma^{j+n}_{\alpha_{j+n}})^{e_{j+n}}$$

*such that $\alpha_k = \alpha_j$ for all $k \in \{j, ..., j+n\}$ and $\alpha_{j-1} \neq \alpha_j \neq \alpha_{j+n+1}$ exists if and only if the leaf exponent matrix for $l_{\alpha_j}$ in the positive tree of the minimal tree-pair diagram representative for $x$ is such that $\epsilon_{\alpha_j,l} = \omega(\gamma^l_{\alpha_l})$ and $d_{\alpha_i,l} = e_l$ for all $l \in \{j, ..., j+n\}$ where $\omega(\gamma_i)$ is defined such that*

$$\omega(\gamma_i) = \begin{cases} 2 \text{ if } \gamma_i = x_i \\ 3 \text{ if } \gamma_i = z_i \end{cases}$$

*and where $\epsilon_{i,j}$ and $d_{i,j}$ refer to entries in the leaf exponent matrix.*

*Proof.* We begin by considering Figure 9.11. We let $v$ be an element of $F(2,3)$ with this minimal tree-pair diagram representative. We note that both $x_i$ and

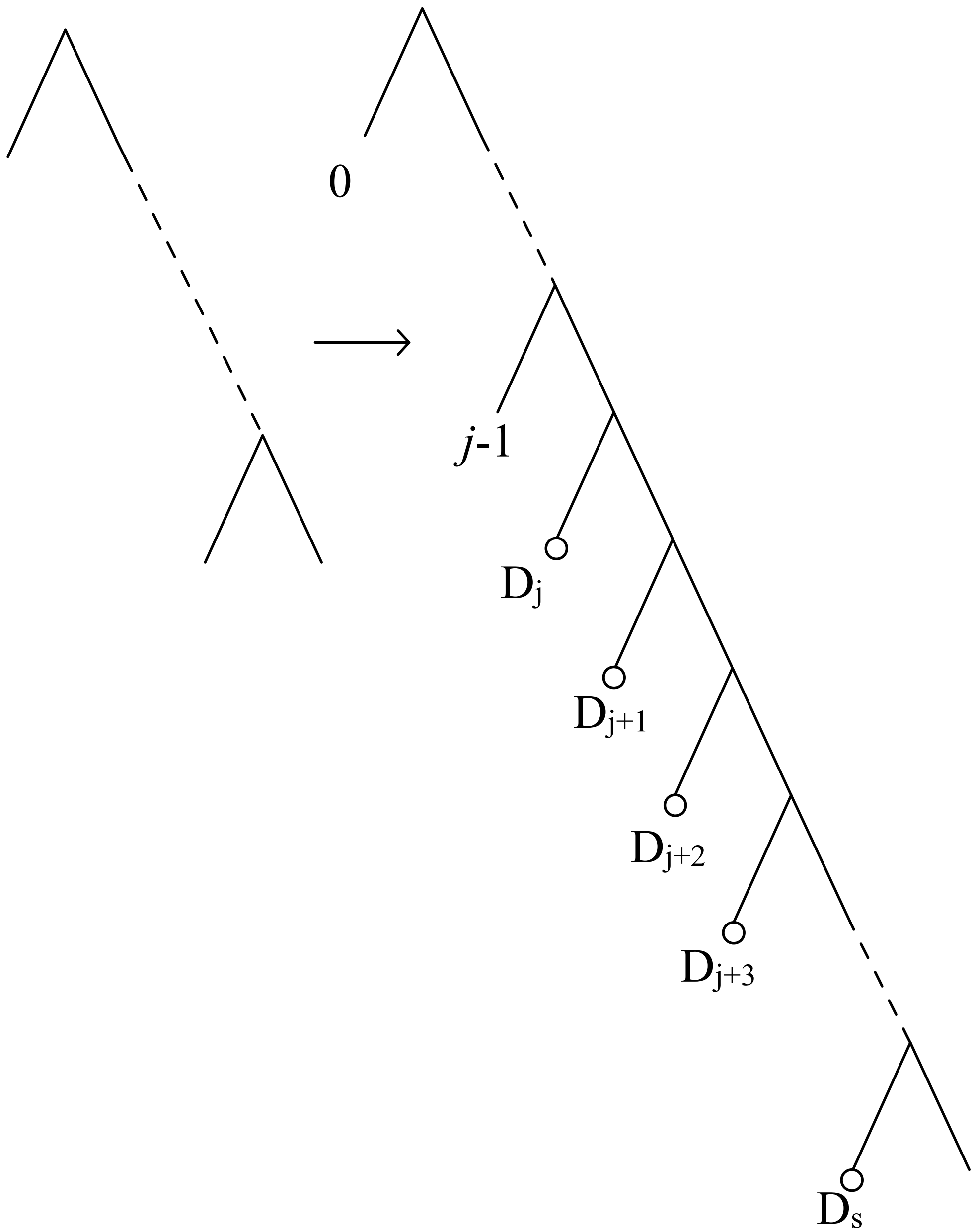


Figure 9.11: Minimal tree-pair diagram representative of an element of $F(2,3)$ where the root and all right carets in the positive tree are 2–ary and the root and the first $j-1$ right carets each have no left child. $D_j, ..., D_s$ are (possibly empty) subtrees.

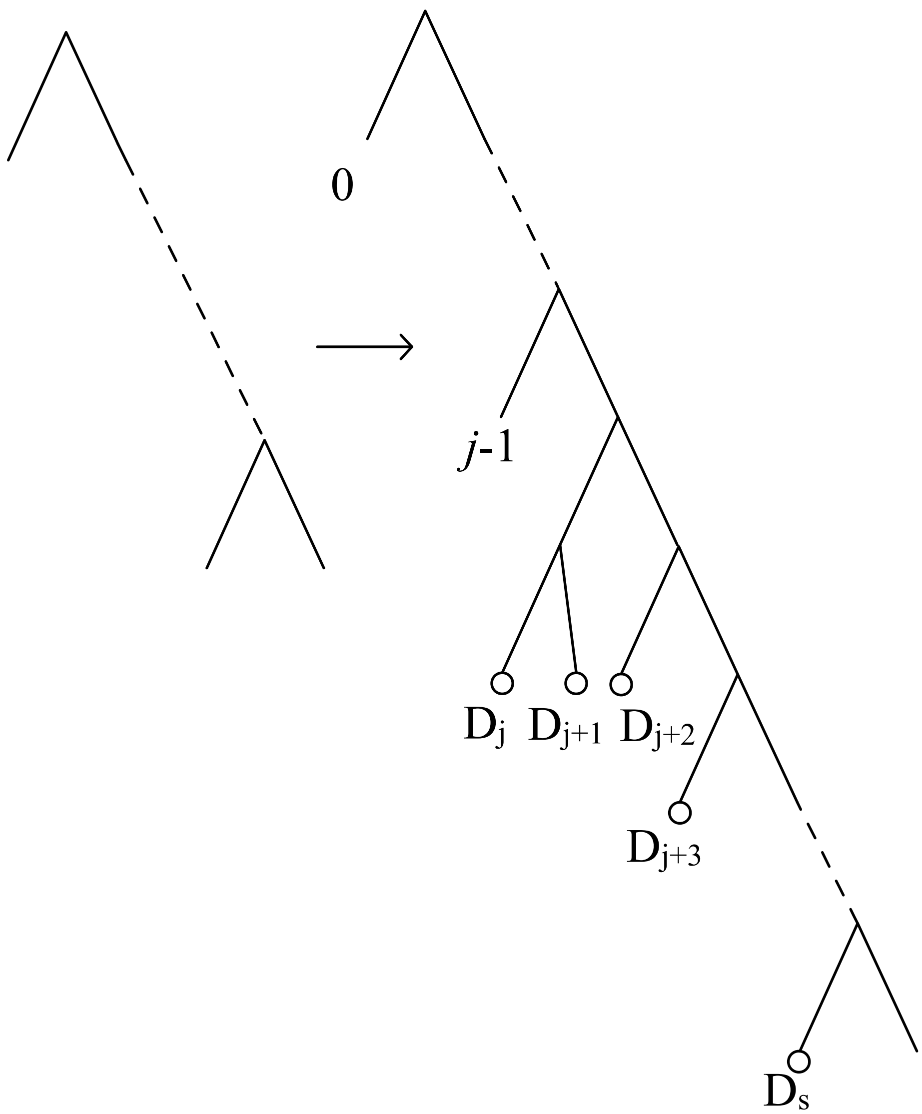


Figure 9.12: Minimal tree-pair diagram representative of an element of $F(2,3)$ obtained by multiplying $x_j$ by an element with minimal tree-pair diagram represented by Figure 9.11. $D_j, ..., D_s$ are (possibly empty) subtrees.

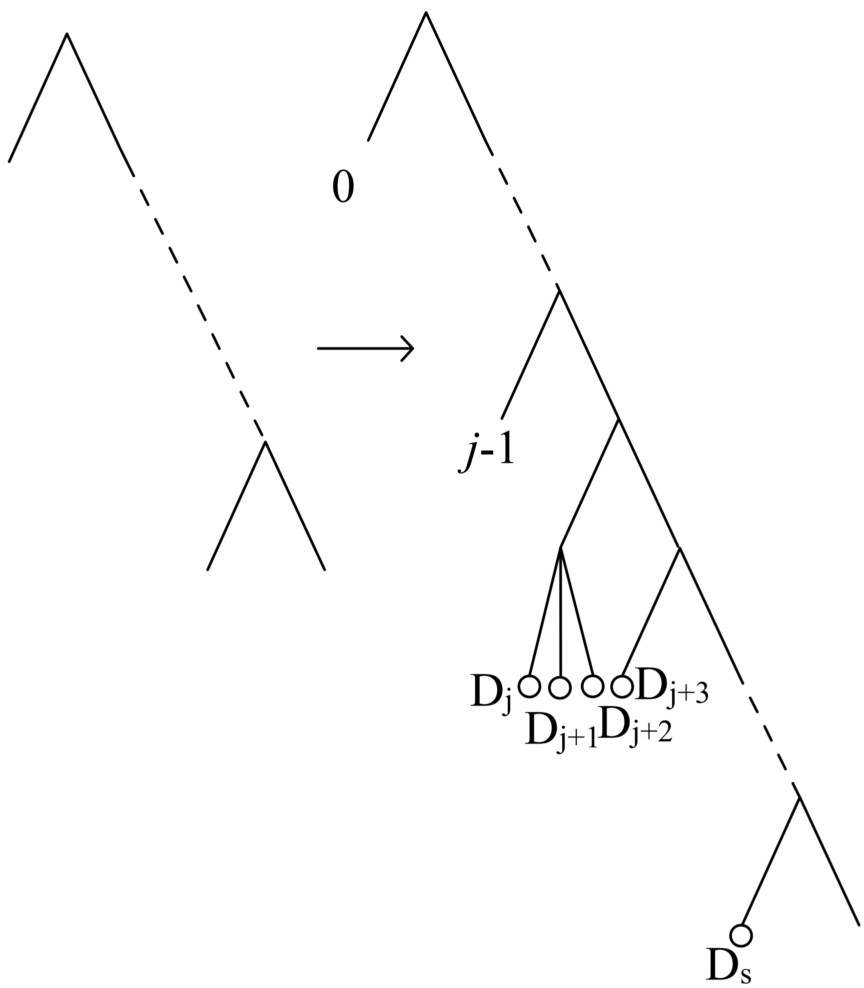


Figure 9.13: Minimal tree-pair diagram representative of an element of $F(2,3)$ obtained by multiplying $z_j$ by an element with minimal tree-pair diagram represented by Figure 9.11. $D_j, ..., D_s$ are (possibly empty) subtrees.

$z_i$ for all $i \in \mathbb{Z}^*$ will have a minimal tree-pair diagram representative with the form given in this figure. Clearly $x_j v$ will have minimal tree-pair diagram given in Figure 9.13 and computing $z_j w$ will produce the tree-pair diagram given in Figure 9.12. So by induction, we can build any tree of the form shown in Figure 9.3 by multiplying together products of $x_i$ and $z_i$ for $i \in \mathbb{Z}^*$, as long as the indexes of any two multiplicands in the product $\gamma_i \gamma_j$ (where $\gamma_k \in \{x_k, z_k\}$) are such that $i \leq j$ and the order of all multiplicands with the same index, (i.e. $(\gamma_{\alpha_j}) \cdots (\gamma_{\alpha_{j+n}})$ such that $\alpha_k = \alpha_j$ for all $k \in \{j, ..., j+n\}$ and $\alpha_{j-1} \neq \alpha_j \neq \alpha_{j+n+1}$) is the same as the order of the 3–ary and 2–ary carets on the leaf exponent path from the direction of the root or right side to the direction of the leaf. So we can conclude that:

$$NF(x) = (\gamma^1_{\alpha_1})^{e_1} \cdots (\gamma^s_{\alpha_s})^{e_s}$$

where $\gamma^i_{\alpha_i}$ is either $x_{\alpha_i}$ or $z_{\alpha_i}$, and $\alpha_i \leq \alpha_j$ whenever $i < j$, and where a subword

$$(\gamma^j_{\alpha_j})^{e_j} \cdots (\gamma^{j+n}_{\alpha_{j+n}})^{e_{j+n}}$$

such that $\alpha_k = \alpha_j$ for all $k \in \{j, ..., j+n\}$ and $\alpha_{j-1} \neq \alpha_j \neq \alpha_{j+n+1}$ exists if and only if the leaf exponent matrix for $l_{\alpha_j}$ in the positive tree of the minimal tree-pair diagram representative for $x$ is such that $\epsilon_{\alpha_j,l} = \omega(\gamma^l_{\alpha_l})$ and $d_{\alpha_i,l} = e_l$ for all $l \in \{j, ..., j+n\}$. It is clear that this factorization is unique if all minimal tree-pair diagram representatives are unique. □

**Lemma 9.9.** *Now we have a method for finding the normal form of elements with minimal tree-pair diagram representative given in Figure 9.5, so by Corollary 9.5 we have a normal form for the element $x$ with minimal tree-pair diagram representative given in Figure 9.7. We let*

$$\mathcal{B}_i = \gamma_0^{m-i+2} \gamma^{m-i+2}_{\alpha_{m-i+2,1}} \cdots \gamma^{m-i+2}_{\alpha_{m-i+2,t_{m-i+2}}}$$

*for* $2 \leq i \leq m+2$ *where* $\gamma_0^{m-i+2}$ *and* $w_{m-i+2}$ *are as defined in Corollary 9.5 and where* $NF(w_{m-i+2})$ *is written*

$$\gamma_{\alpha_{m-i+2,1}}^{m-i+2} \cdots \gamma_{\alpha_{m-i+2,t_{m-i+2}}}^{m-i+2}$$

*where for a given* $l$, $\alpha_{l,i} \leq \alpha_{l,j}$ *if and only if* $i < j$.

*Then*

$$NF(x) = \mathcal{B}_2 \cdots \mathcal{B}_{m+2}$$

*and for a given* $\mathcal{B}_i$ *such that* $i \in \{2, ..., m+2\}$ *and any given* $j \in \{1, ..., t_{m-i+2}\}$,

$$\alpha_{m-i+2,j} \leq \omega(\gamma_0^{m-i+2}) + \sum_{k=1}^{j-1} (\omega(\gamma_{\alpha_{m-i+2,k}}^{m-i+2,k}) - 1)$$

*Proof.* The first part of this Lemma follows immediately from Corollary 9.5 and Theorem 9.8. To see why

$$\alpha_{m-i+2,j} \leq \omega(\gamma_0^{m-i+2}) + \sum_{k=1}^{j-1} (\omega(\gamma_{\alpha_{m-i+2,k}}^{m-i+2,k}) - 1)$$

for all $j \in \{1, ..., t_{m-i+2}\}$ for a given $\mathcal{B}_i \in NF(x)$, we need only note that $\mathcal{B}_i$ will be the normal form for an element with minimal tree-pair diagram representative given by Figure 9.6. This tree-pair diagram has a positive tree with no right carets. If we had

$$\alpha_{m-i+2,j} > \omega(\gamma_0^{m-i+2}) + \sum_{k=1}^{j-1} (\omega(\gamma_{\alpha_{m-i+2,k}}^{m-i+2,k}) - 1)$$

for some $j \in \{1, ..., t_{m-i+2}\}$ then this would mean that the caret containing the leaf numbered $\alpha_j$ could not be a descendent of the left child of the root; so such a leaf would have to be a descendent of the right side of the tree, and the caret which contains such a leaf cannot be the root caret, since this leaf does not have leaf exponent zero. Therefore, there must be a right caret on the tree, which is a contradiction. □

Now we find the normal form of an arbitrary positive element by taking the normal forms given in detail for the "left and right halves" of the positive tree in the minimal tree-pair diagram representative of our positive element, and we put these together using Corollary 9.4.

**Theorem 9.10.** *For a positive element $x$ of $F(2,3)$ whose minimal tree-pair diagram representative can be written in the form given in Figure 9.1, the normal form is:*

$$NF(x) = \mathcal{B}_0 \cdots \mathcal{B}_k$$

*where*

$$\mathcal{B}_0 = y_{\beta_1} \cdots y_{\beta_u}$$

*where $\beta_i < \beta_{i+1} + 1$ for all $i \in \{1, ..., u-1\}$ and $k \in \{\beta_1, ..., \beta_u\}$ if and only if the leaf $l_k$ is the leftmost child of a 3–ary caret in the subtree of Figure 9.1 consisting entirely of right carets and the root caret*

*and where*

$$\mathcal{B}_1 = (\gamma^1_{\alpha_1})^{e_1} \cdots (\gamma^s_{\alpha_s})^{e_s}$$

*where $\gamma^i_{\alpha_i}$ is either $x_{\alpha_i}$ or $z_{\alpha_i}$, and $\alpha_i \leq \alpha_j$ whenever $i < j$, and where a subword*

$$(\gamma^j_{\alpha_j})^{e_j} \cdots (\gamma^{j+n}_{\alpha_{j+n}})^{e_{j+n}}$$

*such that $\alpha_k = \alpha_j$ for all $k \in \{j, ..., j+n\}$ and $\alpha_{j-1} \neq \alpha_j \neq \alpha_{j+n+1}$ exists if and only if the leaf exponent matrix for $l_{\alpha_j}$ in the subtree of the positive tree of the minimal tree-pair diagram representative for $x$ which consists solely of the right side of the tree and all carets descended from right carets or from the right or middle edge of the root (if it exists) is such that $\epsilon_{\alpha_j,l} = \omega_1(l)$ and $d_{\alpha_i,l} = e_l$ for all $l \in \{j, ..., j+n\}$*

*and where*

$$\mathcal{B}_i = \gamma^{k-i}_0 \gamma^{k-i}_{\alpha_{k-i,1}} \cdots \gamma^{k-i}_{\alpha_{k-i,t_{k-i}}}$$

*for* $2 \leq i \leq k = m+2$ *where* $\gamma_0^{k-i}$ *and* $w_{k-i}$ *are as defined in Corollary 9.5 and where* $NF(w_{k-i})$ *is written*

$$\gamma^{k-i}_{\alpha_{k-i,1}} \cdots \gamma^{k-i}_{\alpha_{k-i,t_{k-i}}}$$

*where for a given* $l$*,* $\alpha_{l,i} \leq \alpha_{l,j}$ *if and only if* $i < j$*, and for a given* $\mathcal{B}_i$ *such that* $i \in \{2, ..., k\}$ *and any given* $j \in \{1, ..., t_{k-i}\}$*,*

$$\alpha_{k-i,j} \leq \omega(\gamma_0^{k-i}) + \sum_{l=1}^{j-1} (\omega(\gamma^{k-i,l}_{\alpha_{k-i,l}}) - 1)$$

*In this way we can write an algebraic expression in normal form for any word in* $F(2,3)$ *just by looking at the placement of carets in its minimal tree-pair diagram, and we can draw a minimal tree-pair diagram for any word by putting that word into its normal form and using the generators in the normal form to determine the placement of carets in the tree.*

*Proof.* This follows immediately from Lemma 9.1, Corollaries 9.4, 9.5 and 9.6, and Theorem 9.8, and Lemma 9.9. □

We note that the normal form we have developed in Theorem 9.10 is not unique. If we look carefully at the proof of this theorem and the lemmas, etc. leading up to it, we will see that this normal form will be unique only when the choice of minimal tree-pair diagram representative is unique. However, we have already seen that elements of $F(2,3)$ may have more than one distinct minimal tree-pair diagram representative. In this sense, the normal form given in Theorem 9.10 could more accurately be called the unique normal form of a particular minimal tree-pair diagram representative, rather than the normal form of the element of $F(2,3)$ represented by that minimal tree-pair diagram.

However, if we place further restrictions on our choice of minimal tree-pair diagram representative, we can produce a unique normal form for elements of $F(2,3)$. The reason why some minimal tree-pair diagrams will not be unique is the presence of subtrees of the form given in Figure 7.16 which can be exchanged for one another to produce an equivalent but distinct tree-pair diagram with the same number of leaves. If we simply establish a rule which determines which of these equivalent subtrees we choose for a given minimal tree-pair diagram representative, then we can establish a method for choosing a unique minimal tree-pair diagram representative for any given element of $F(2,3)$.

**Theorem 9.11.** *For a given set of all minimal tree-pair diagram representatives of the element $x$ in $F(2,3)$, we choose the minimal tree-pair diagram representative for which all instances of any subtree of the form given in Figure 7.16 have the form given in Figure 9.14. We will call a minimal tree-pair diagram which has been uniquely chosen in this way the normal tree-pair diagram representative.*

*This choice of minimal tree-pair diagram representative will have a normal form which is equivalent to the normal form obtained by taking the normal form of any minimal tree-pair diagram representative of $x$ and replacing all occurrences of the subwords $(y_{i+1}z_i)^{\pm 1}$ and $(x_iz_{i+1}z_i)^{\pm 1}$ with $(y_ix_{i+1}x_i)^{\pm 1}$ and $(z_ix_{i+2}x_{i+1}x_i)^{\pm 1}$ respectively (although some rearrangement of the normal form may be necessary in order to locate these occurrences and this intermediate rearrangement may put the expression into a form which is not normal - the expression can be put back into normal form once all substitutions have been carried out to yield a new normal form expression).*

*If the normal form of $x$ is denoted by*

$$NF(x) = \mathcal{B}_0\mathcal{B}_1 \cdots \mathcal{B}_k\mathcal{C}_l^{-1} \cdots \mathcal{C}_1^{-1}\mathcal{C}_0^{-1}$$

*where $NF(x_+) = \mathcal{B}_0\mathcal{B}_1 \cdots \mathcal{B}_k$ and $NF(x_-) = \mathcal{C}_0\mathcal{C}_1 \cdots \mathcal{C}_l$, then any single instance of one of the subwords $(y_{i+1}z_i)^{\pm 1}$ or $(x_iz_{i+1}z_i)^{\pm 1}$ will be present in no more than two blocks of the normal form. If this instance is present in the positive part of the normal form, a single instance of the subwords $y_{i+1}z_i$ and $x_iz_{i+1}z_i$ will be present* *in one of the blocks*

$$\mathcal{B}_1, ..., \mathcal{B}_k$$

*or one of the following subwords:*

$$\mathcal{B}_0\mathcal{B}_1, \mathcal{B}_2\mathcal{B}_3, \mathcal{B}_3\mathcal{B}_4, ..., \mathcal{B}_{k-1}\mathcal{B}_k, \mathcal{B}_0\mathcal{B}_1\mathcal{B}_2$$

*where if the subword* $y_{i+1}z_i$ *or* $x_iz_{i+1}z_i$ *is present in* $\mathcal{B}_0\mathcal{B}_1\mathcal{B}_2$*, all parts of that subword will be contained entirely in* $\mathcal{B}_0$ *and* $\mathcal{B}_2$ *(although in order for these two parts to be united, one of these two parts must move through the block* $\mathcal{B}_1$*). If this instance is present in the negative part of the normal form, a single instance of the subword* $(y_{i+1}z_i)^{-1}$ *or* $(x_iz_{i+1}z_i)^{-1}$ *will be present in one of the blocks*

$$\mathcal{C}_l^{-1}, ..., \mathcal{C}_1^{-1}$$

*or one of the following subwords:*

$$\mathcal{C}_1^{-1}\mathcal{C}_0^{-1}, \mathcal{C}_3^{-1}\mathcal{C}_2^{-1}, \mathcal{C}_4^{-1}\mathcal{C}_3^{-1}, ..., \mathcal{C}_l^{-1}\mathcal{C}_{l-1}^{-1}, \mathcal{C}_2^{-1}\mathcal{C}_1^{-1}\mathcal{C}_0^{-1}$$

*where if the subword* $(y_{i+1}z_i)^{-1}$ *or* $(x_iz_{i+1}z_i)^{-1}$ *is present in* $\mathcal{C}_2^{-1}\mathcal{C}_1^{-1}\mathcal{C}_0^{-1}$*, all parts of that subword will be contained entirely in* $\mathcal{C}_2^{-1}$ *and* $\mathcal{C}_0^{-1}$ *(although in order for these two parts to be united, one of these two parts must move through the block* $\mathcal{C}_1^{-1}$*).*

*We call this new unique choice of normal form for* $x$ *the unique normal form, and we denote it by* $UNF(x)$*.*

*Proof.* First we show that it is always possible to replace all instances of subtrees of the form given in Figure 7.16 within either tree of a minimal tree-pair diagram representative with the subtree given in Figure 9.14 in a unique way. We will show that this is true for any instances in the positive tree; our argument then clearly holds for any instances in the negative tree. We begin by considering all subtrees of the form given in Figure 9.15 which are present in the tree.

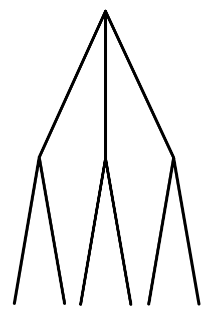

Figure 9.14: Choice of the two equivalent subtrees given in Figure 7.16 for the normal tree-pair diagram representative of an element $x$ of $F(2,3)$

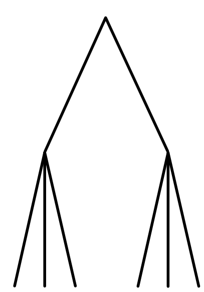

Figure 9.15: We replace all instances of this subtree in a minimal tree-pair diagram representative of an element of $F(2,3)$ with the equivalent subtree given in Figure 7.16 in order to obtain the normal tree-pair diagram representative

We consider a single instance of a subtree of the type given in Figure 9.15. If this single subtree does not share any carets with any other subtrees of the type given in Figure 9.15, then we can replace the subtree under consideration with the equivalent subtree given in Figure 9.14 without affecting our ability to perform further future replacements of this type. Next we consider what will happen if the subtree given in Figure 9.15 does share carets with another subtree of the type given in Figure 9.15: we see that this is not possible, because any subtree with only two levels which shares carets with the given subtree must have either a 3–ary root or a 2–ary child. So replacing a subtree of the type given in Figure 9.15 with the equivalent subtree given in Figure 9.14 will always be possible without affecting any future ability to perform a further substitution.

Now we show that these subtree substitutions in the minimal tree-pair diagram representative are equivalent to the algebraic substitutions given in this theorem. We will show that this is true for any instances in the positive tree; our argument then clearly holds for any instances in the negative tree. First we consider the minimal tree-pair diagram representatives of the subwords $y_i x_{i+1} x_i$, $z_i x_{i+2} x_{i+1} x_i$, $y_{i+1} z_i$, and $x_i z_{i+1} z_i$. The easiest way to do this is to put each of these words into normal form. Using the relators to do this yields $y_i x_i x_{i+2}$, $z_i x_i x_{i+2} x_{i+4}$, $y_{i+1} z_i$, and $x_i z_i z_{i+3}$ respectively (these normal forms actually only apply whenever $i \neq 0$ - whenever $i = 0$, each of these is already in normal form except $z_i x_{i+2} x_{i+1} x_i$, which will have normal form $z_i x_{i+1} x_{i+3} x_i$ when $i = 0$). From these normal forms, we can easily see that the minimal tree-pair diagram representative of each of these words will have the form given in Figures 9.16, 9.17 9.18, and 9.19 respectively.

From our calculations and proofs leading up to the normal form given in Theorem 9.10, we can see that an instance of $y_i x_{i+1} x_i$, $z_i x_{i+2} x_{i+1} x_i$, $y_{i+1} z_i$, or $x_i z_{i+1} z_i$ is present in the normal form derived from a given minimal tree-pair diagram representative of a word $x$ if and only if the positive tree of that minimal tree-pair

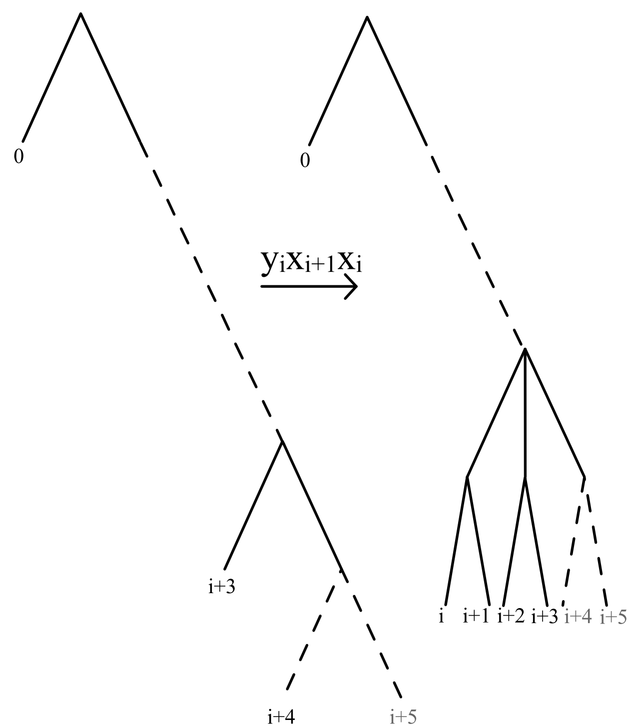


Figure 9.16: Minimal tree-pair diagram representative of $y_i x_{i+1} x_i = y_i x_i x_{i+2}$ of $F(2,3)$, where the dotted caret pair shows a pair of carets that could be added to the diagram but which are not present in the minimal representative.

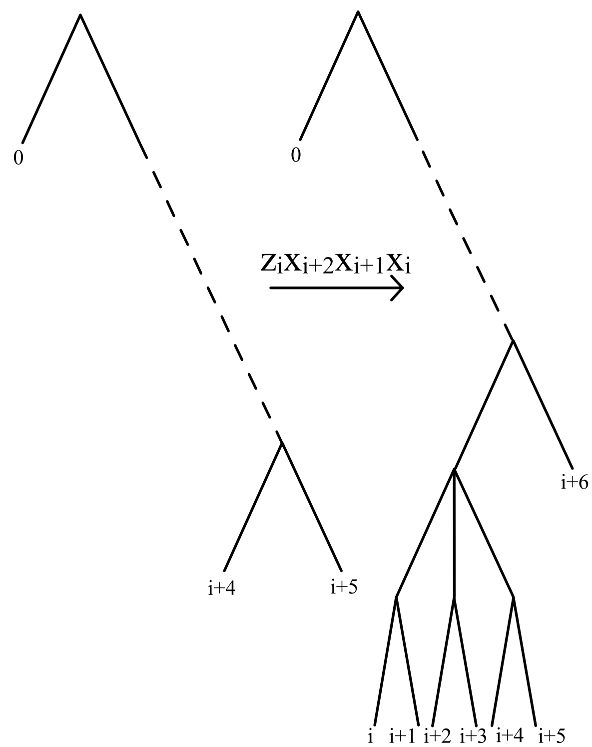


Figure 9.17: Minimal tree-pair diagram representative of $z_i x_{i+2} x_{i+1} x_i = z_i x_i x_{i+2} x_{i+4}$ of $F(2,3)$

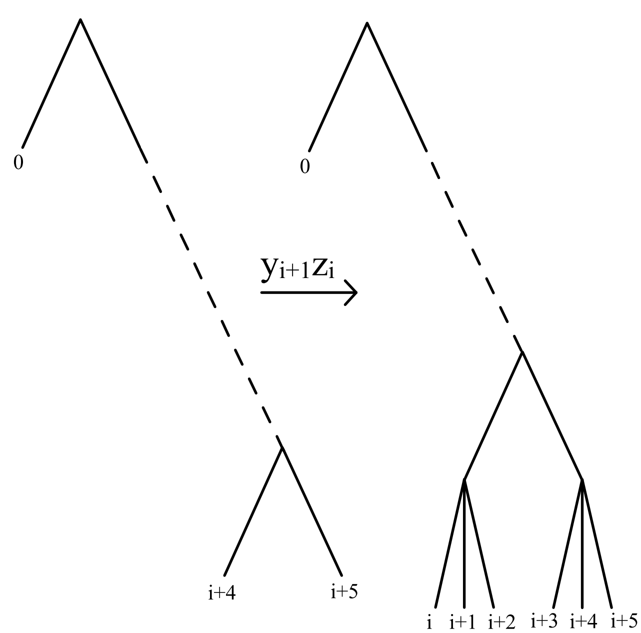


Figure 9.18: Minimal tree-pair diagram representative of $y_{i+1} z_i$ of $F(2,3)$

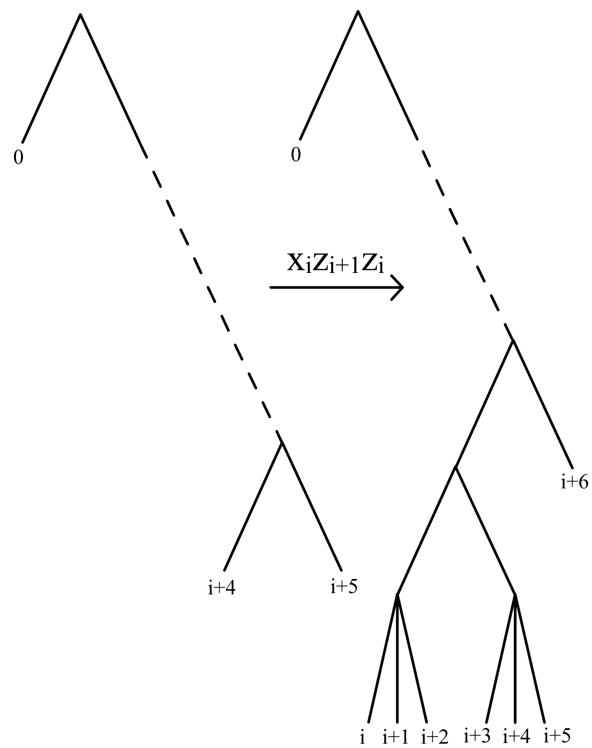


Figure 9.19: Minimal tree-pair diagram representative of $x_i z_{i+1} z_i = x_i z_i z_{i+3}$ of $F(2,3)$

diagram representative has as a subtree the tree given in Figure 9.20, 9.21 9.22, or 9.23 respectively.

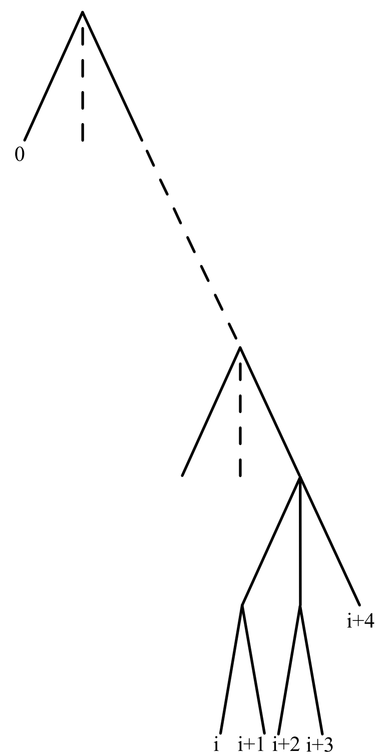


Figure 9.20: Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $y_i x_{i+1} x_i$ is present in the normal form; this subtree shares the same root as the larger tree. The dotted edges within a caret indicate that that caret may be 2–ary or 3–ary.

From looking at these subtrees, it becomes clear that the algebraic substitution will be equivalent to the substitution of the subtrees in the minimal tree-pair diagram representative.

Now we show that any instances of the subword $(y_{i+1}z_i)^{\pm 1}$ or $(x_i z_{i+1} z_i)^{\pm 1}$ in the normal form of a minimal tree-pair diagram representative will be contained entirely in one of the subwords as described in this theorem. (Again we prove this for the positive tree only, as it will then also clearly be true in the case of the negative tree.) This becomes obvious when we look at the subtrees given in Figures 9.20, 9.21 9.22, and 9.23; the subtrees given in Figures 9.20 and 9.22 will

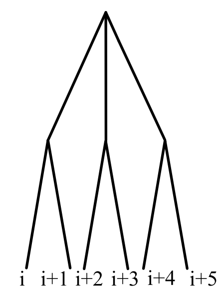


Figure 9.21: Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $z_i x_{i+2} x_{i+1} x_i$ is present in the normal form; the numbering in this figure refers to the leaf numbering in a larger subtree which contains the subtree given in this figure and all its ancestor carets in the larger tree. The subtree given in this figure will contain only carets which are not on the right side of the larger tree.

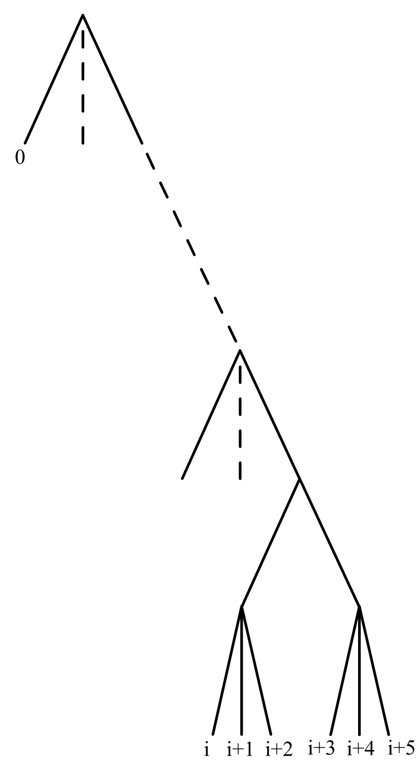


Figure 9.22: Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $y_{i+1} z_i$ is present in the normal form; this subtree shares the same root as the larger tree. The dotted edges within a caret indicate that that caret may be 2–ary or 3–ary.

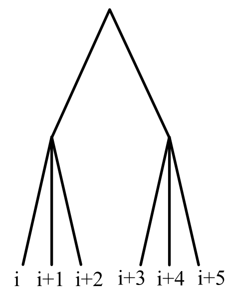


Figure 9.23: Subtree present in the minimal tree-pair diagram representative of $x$ of $F(2,3)$ if $x_i z_{i+1} z_i$ is present in the normal form; the numbering in this figure refers to the leaf numbering in a larger subtree which contains the subtree given in this figure and all its ancestor carets in the larger tree. The subtree given in this figure will contain only carets which are not on the right side of the larger tree.

be contained entirely in $\mathcal{B}_0\mathcal{B}_1$ if $i \neq 0$, and will be contained in blocks $\mathcal{B}_0$ and $\mathcal{B}_2$ if $i = 0$ (although in order to obtain the subword we want, we will have to move some generators from $\mathcal{B}_0$ or $\mathcal{B}_2$ through block $\mathcal{B}_1$). The subtrees given in Figures 9.21 and 9.23 will be contained entirely in $\mathcal{B}_{k-j}\mathcal{B}_{k-j+1}$ if the root caret in the given subtree is the $jth$ left caret on the left side of the larger tree (where the numbering begins with 0 and we note that $k - j \geq 2$. It will be contained in the block $\mathcal{B}_{k-j}$ if the first left ancestor caret of the given subtree (on the directed path from the root of the given subtree to the root of the larger tree) is the $jth$ left caret on the left side of the larger tree (where the numbering begins with 0). It will be contained in the block $\mathcal{B}_1$ if all the ancestor carets of the given subtree in the larger tree are middle or right carets except the root. □

For illustrative purposes, we find the normal form of an element $w$ of $F(2,3)$. We calculate the normal form of the element given in Figure 9.24.

We first find the normal form of the positive element with the positive tree of Figure 9.24 as its positive tree, $w_+$. To obtain the normal form, we first consider

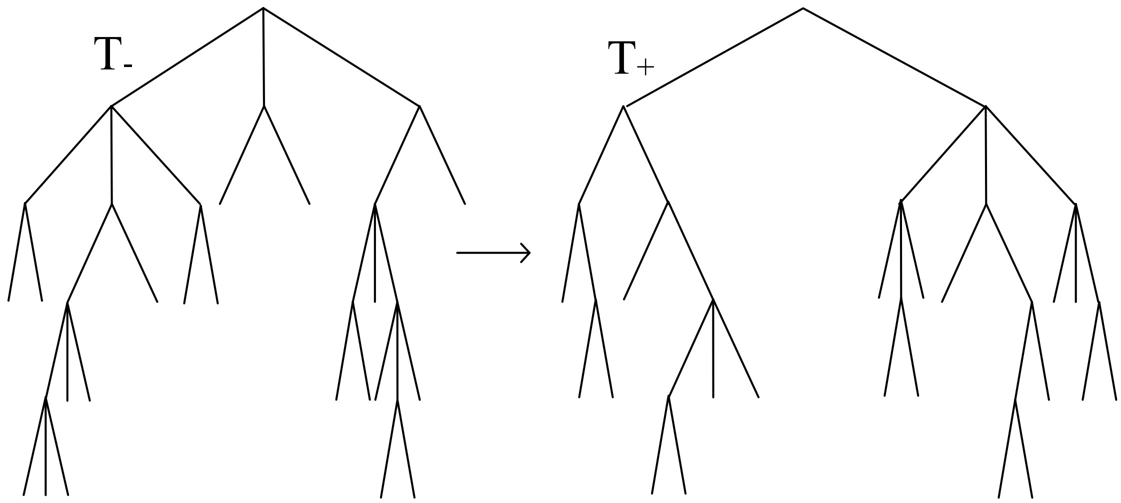


Figure 9.24: Tree-pair diagram representative of an element $w$ of $F(2,3)$

the subtree of the positive tree consisting of all the carets on the right side of the tree. In this subtree, the leaf index numbers of the first leaf in each 3–ary caret are 1 and 3. So this gives us

$$\mathcal{B}_0 = y_1 y_3$$

Next we consider the subtree of the positive tree which consists of the root caret and any carets descended from its right child node. Numbering the leaves on this subtree gives us the following leaf exponent matrices:

$$E_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, E_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_5 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_6 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

So this yields

$$\mathcal{B}_1 = z_1 x_2 x_5 x_6^2$$

Now we consider the tree formed from the subtree of the positive tree which consists of the left child of the root and all of the carets descended from the right child node of the left child of the root by adding a 2–ary root caret so that the left child of the root in $T_+$ is the left child of the 2–ary root caret in this new tree. Numbering the leaves on this subtree gives us the following leaf exponent matrices:

$$E_0 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_2 = \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix}$$

So this yields

$$\mathcal{B}_2 = x_0 x_1 z_2 x_2$$

Now we consider the tree formed from the subtree of the positive tree which consists of the left child of the left child of the root and all of the carets descended from the right child node of the left child of the left child of the root by adding a 2–ary root caret so that the left child of the left child of the root in $T_+$ is the left child of the 2–ary root caret in this new tree. Numbering the leaves on this subtree gives us the following leaf exponent matrices:

$$E_0 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

So this yields

$$\mathcal{B}_3 = x_0 x_1$$

Putting all of these blocks together yields

$$NF(w_+) = \mathcal{B}_0\mathcal{B}_1\mathcal{B}_2\mathcal{B}_3 = y_1 y_3 z_1 x_2 x_5 x_6^2 x_0 x_1 z_2 x_2 x_0 x_1$$

Now we find the normal form of the positive element with the negative tree of Figure 9.24 as its positive tree, $w_-$. To avoid confusion with the blocks in the normal form of the positive word we just calculated, we will replace all the block labels $\mathcal{B}_i$ for this word with the block labels $\mathcal{C}_i$. To obtain the normal form, we first consider the subtree of the negative tree consisting of all the carets on the right side of the tree. In this subtree, the leaf index number of the first leaf in the only 3–ary caret is 0. So this gives us

$$\mathcal{C}_0 = y_0$$

Next we consider the subtree of the negative tree which consists of the root caret and any carets descended from its right child node. Numbering the leaves on this subtree gives us the following leaf exponent matrices:

$$E_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, E_3 = \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix}, E_6 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, E_7 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

So this yields

$$\mathcal{C}_1 = x_1 z_3 x_3 z_6 x_7$$

Now we consider the tree formed from the subtree of the negative tree which consists of the left child of the root and all of the carets descended from the right child node of the left child of the root by adding a 2–ary root caret so that the left child of the root in $T_-$ is the left child of the 2–ary root caret in this new tree. Numbering the leaves on this subtree gives us the following leaf exponent matrices:

$$E_0 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, E_1 = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix}, E_7 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

So this yields

$$\mathcal{C}_2 = z_0 x_1 z_1^2 x_7$$

Now we consider the tree formed from the subtree of the negative tree which consists of the left child of the left child of the root and all of the carets descended from the right child node of the left child of the left child of the root by adding a 2–ary root caret so that the left child of the left child of the root in $T_-$ is the left child of the 2–ary root caret in this new tree. Numbering the leaves on this subtree gives us the following leaf exponent matrix:

$$E_0 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

So this yields

$$\mathcal{C}_3 = x_0$$

Putting all of these blocks together yields

$$NF(w_-) = \mathcal{C}_0\mathcal{C}_1\mathcal{C}_2\mathcal{C}_3 = y_0x_1z_3x_3z_6x_7z_0x_1z_1^2x_7x_0$$

Putting the normal forms of these two elements together yields:

$$\begin{aligned} NF(w) &= NF(w_+)NF(w_-^{-1}) \\ &= y_1y_3z_1x_2x_5x_6^2x_0x_1z_2x_2x_0x_1(y_0x_1z_3x_3z_6x_7z_0x_1z_1^2x_7x_0)^{-1} \\ &= y_1y_3z_1x_2x_5x_6^2x_0x_1z_2x_2x_0x_1x_0^{-1}x_7^{-1}z_1^{-2}x_1^{-1}z_0^{-1}x_7^{-1}z_6^{-1}x_3^{-1}z_3^{-1}x_1^{-1}y_0^{-1} \end{aligned}$$

We note that this normal form is unique because the minimal tree-pair diagram representative given in Figure 9.24 is normal. So we could write

$$UNF(w) = y_1y_3z_1x_2x_5x_6^2x_0x_1z_2x_2x_0x_1x_0^{-1}x_7^{-1}z_1^{-2}x_1^{-1}z_0^{-1}x_7^{-1}z_6^{-1}x_3^{-1}z_3^{-1}x_1^{-1}y_0^{-1}$$

To compare this unique normal form with with another normal form for a minimal tree-pair diagram which is not normal, we consider the minimal tree-pair diagram given in Figure 9.25, which can be derived from Figure 9.24 by replacing two instances of subtrees of the type given in Figure 9.14 with the equivalent type given in Figure 9.15.

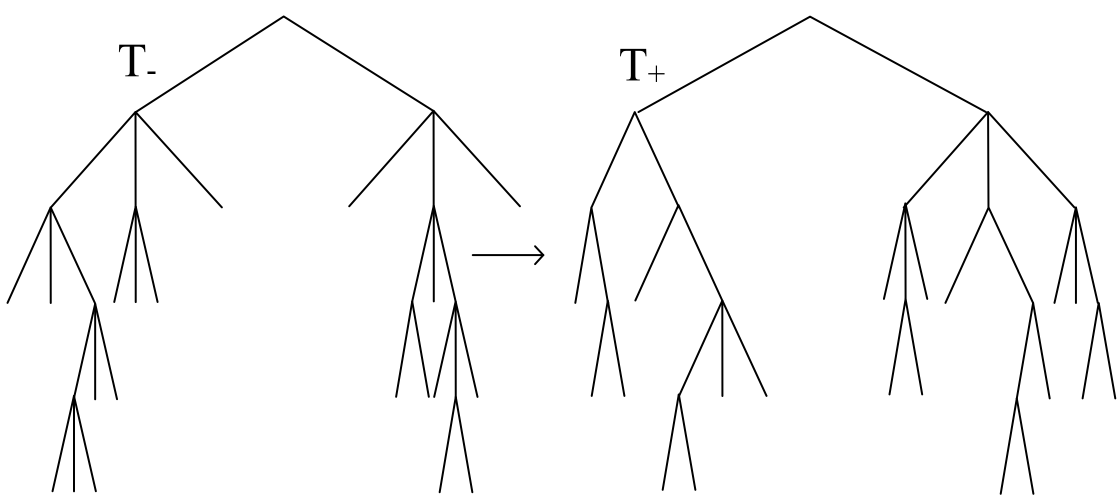


Figure 9.25: Minimal tree-pair diagram representative equivalent to the normal tree-pair diagram representative of the element of $F(2,3)$ given by Figure 9.24

Then, using the same procedure as before, we obtain the normal form for the element whose minimal tree-pair diagram representative is depicted in 9.25:

$$NF(w) = y_1 y_3 z_1 x_2 x_5 x_6^2 x_0 x_1 z_2 x_2 x_0 x_1 z_2^{-2} z_0^{-1} z_1^{-1} z_0^{-1} x_6^{-1} z_5^{-1} x_2^{-1} z_2^{-1} y_1^{-1}$$

Since both the normal form given here and the unique normal form derived above represent the same element $w$, we should be able to see that these two algebraic expressions are equal by performing the kinds of replacements described in Theorem 9.11. Since only the negative tree changes, we concern ourselves only with $NF(w_-)$:

$$\begin{aligned} NF(w_-) &= y_1 z_2 x_2 z_5 x_6 z_0 z_1 z_0 z_2^2 \\ &= y_1 z_0 z_4 x_4 z_7 x_8 z_0 z_1 z_0 z_2^2 \\ &= y_0 x_1 x_0 z_4 x_4 z_7 x_8 z_0 z_1 z_0 z_2^2 \\ &= y_0 x_1 z_3 x_3 z_6 x_7 x_0 z_1 z_0 z_2^2 \\ &= y_0 x_1 z_3 x_3 z_6 x_7 z_0 x_2 x_1 x_0 z_2^2 \\ &= y_0 x_1 z_3 x_3 z_6 x_7 z_0 x_1 z_1^2 x_7 x_0 \end{aligned}$$

We see that there exists a $y_1$ in $\mathcal{C}_0$ and a $z_0$ in $\mathcal{C}_2$. Moving $z_0$ to the front yields line 2 of the equation above. Substituting $y_0 x_1 x_0$ for $y_1 z_0$ then yields the next line. We note that lines 2 and 3 above are not in normal form. To obtain a normal form again, we move $x_0$ as far to the right as possible to obtain line 4 above; this line is again in normal form, and we will denote the blocks present in line 4 by $\mathcal{D}_0\mathcal{D}_1\mathcal{D}_2\mathcal{D}3$. Now we notice that we have $x_0 z_1$ present in block $\mathcal{D}_2$ and $z_0$ present in block $\mathcal{D}_3$; these are already next to each other, so we can replace this expression $x_0 z_1 z_0$ immediately with $z_0 x_2 x_1 x_0$ to obtain line 5. The expression on line 5 is now again not in normal form. So we rearrange the generators in the last two blocks to obtain line 6, which is again in normal form; in fact, line 6 is in unique normal form.

We can see how different the unique normal form is from the second normal form we calculated for $w$ which was not unique. This illustrates how difficult it may be to identify the instances of a relator in the algebraic version of the normal form which will allow us to write out the unique normal form from the algebraic expression alone. However, getting the unique normal form from a tree-pair diagram representative is relatively easy, because in the tree-pair diagram representative it is much more intuitive to see instances of subtrees of the types given in Figure 7.16, so even when given an algebraic expression in normal form rather than a tree-pair diagram, it may be more efficient to use that normal form to draw a minimal tree-pair diagram representative on which we can then perform subtree substitutions of the kind given in Theorem 9.11 to put it in normal from; we can then use the normal tree-pair diagram representative to write down the unique normal form of $w$.

The normal form expression for an element which is given in Theorem 9.10 is not minimal, even in the infinite generators. For example, the unique normal form of the minimal tree-pair diagram for $y_0^2$ is $y_0 y_2 z_1 z_0 x_0^{-1} x_1^{-1} x_0^{-1} x_1^{-2}$.

More generally we will see that the number of generators listed in the normal form for the negative tree of $y_0^n$ will be equal to the number of carets in the tree minus the number of carets on the right edge of the tree (including the root caret), which is equal to $3^n - n - 2$. Similarly, the number of generators listed in the normal form for the positive tree of $y_0^n$ will be equal to the number of carets in the tree, which will be $\frac{3^n-1}{2}$. So the total number of generators in the normal form of $y_0^n$ will be $\frac{3^{n+1}-2n-5}{2}$, and we can see that the number of generators in this normal form can be a poor approximation for the metric.

Also, we may be interested in expressing some of the generators of $F(3)$ in terms of the generators of $F(2,3)$. We can now do that with the use of this normal form. We will let $x_i'$ denote the generator $x_i$ in $F(3)$ so that it is not confused with

$x_i$ in $F(2,3)$, which denotes the generator $x_i$ of $F(2)$. Using our construction of the normal form, the unique normal form for $x_i'$ is:

$$NF(x_i') = y_0 y_2 y_4 \cdots y_{n-3} z_i y_{n-5}^{-1} \cdots y_4^{-1} y_2^{-1} y_0^{-1}$$

where $n$ denotes the number of leaves in the minimal tree-pair diagram representative and

$$n = i + \begin{cases} 4 \text{ if } i \text{ is odd} \\ 5 \text{ if } i \text{ is even} \end{cases}$$

# Chapter 10

## The Metric on $F(2,3)$

For Thompson's group $F(p+1)$, Burillo, Cleary and Stein have shown in [4] that the metric is quasi-isometric to the number of carets in the minimal 2–ary tree-pair diagram representative of the element of the group (we note that this also follows from Fordham's method for finding the metric on $F(p+1)$, which was developed after [4]). This however, is not the case for $F(2,3)$. The simplest way to illustrate this is to take powers of the generator $y_0$ and to show that the number of carets in the minimal tree-pair diagram representative for $y_0^n$ grows exponentially as $n$ (and therefore $|y_0^n|_{\{x_0,x_1,y_0,y_1\}}$) increases linearly.

**Remark 10.1.** *The number of carets in a minimal tree-pair diagram representative of an element $w$ of $F(2,3)$ is quasi-isometric to the number of leaves in the minimal tree-pair diagram representative.*

*Proof.* Since the number of leaves in a 2–ary tree containing $C$ carets will be $C+1$, and the number of leaves in a 3–ary tree containing $C$ carets will be $2C+1$, the number of leaves, $L$, in a $(2,3)$–ary tree will be such that $C+1 \leq L \leq 2C+1$. This implies that $\frac{1}{3}C \leq L \leq 3C$, so the number of leaves in a tree or tree-pair diagram is quasi-isometric to the number of carets. □

Because of Remark 10.1, we will often refer to the number of leaves in a minimal tree-pair diagram and the number of carets in a minimal tree-pair diagram somewhat interchangeably. The convention in research with respect to $F(p+1)$

has been to talk about the number of carets in the minimal tree-pair diagram representative, but since the number of carets in a (2,3)–ary tree-pair diagram may not be the same in both the positive and negative trees, in the case of $F(2,3)$, it makes more sense to refer to the number of leaves in the minimal tree-pair diagram representative instead.

**Theorem 10.2.** *The metric on $F(2,3)$ is not quasi-isometric to the number of carets or leaves in the reduced tree-pair diagram representatives of elements of $F(2,3)$.*

*Proof.* We prove this by showing that the number of leaves in the minimal tree-pair diagram representative for $y_0^n$ grows exponentially as $|y_0^n|_{\{x_0,x_1,y_0,y_1\}} = n$ increases linearly where $n \in \mathbb{N}$. We will let $(T_-, T_+)$ denote the reduced tree-pair diagram for $y_0$, and we will let $(T_-^n, T_+^n)$ denote the reduced tree-pair diagram for $y_0^n$.

We begin with the following inductive hypothesis: The leaf-path chart for $y_0^n$ in $F(2,3)$ (for $n \in \mathbb{N}$) has $3^n$ rows, and $\mathbf{v}(l_i \in T_+^n) = \langle 0, n\rangle$ for all $i$, $v_3(l_i \in T_-^n) = 0$ and

$$v_2(l_i \in T_-^n) = \begin{cases} v_2(l_{\lfloor \frac{i}{2} \rfloor} \in T_-^{n-1}) + 1 & \text{for } i \in \{0, ..., 3^{n-1} - 1\} \\ v_2(l_{\lfloor \frac{i}{2} \rfloor} \in T_-^{n-1}) + 2 & \text{for } i = 3^{n-1} \\ v_2(l_{\lfloor \frac{i}{2} \rfloor - 1} \in T_-^{n-1}) + 2 & \text{for } i = 3^{n-1} + 1 \\ v_2(l_{\lfloor \frac{i-(3^{n-1}+2)}{4} \rfloor + \lfloor \frac{3^{n-1}+1}{2} \rfloor} \in T_-^{n-1}) + 2 & \text{for } i \in \{3^{n-1} + 2, ..., 3^n - 1\} \end{cases}$$

We can see by looking at the leaf-path chart for $y_0$ in Chart 10.1 that this is true for $n = 1$, if we recall that a leaf-path chart for the identity element will have only 1 row with all valences 0.

Now we proceed with the induction by considering the product $y_0^{n+1} = y_0 y_0^n$. We suppose that the inductive hypothesis holds for arbitrary $n$. This produces the leaf-path chart for $y_0^n$ given in Chart 10.2.

Table 10.1: Leaf-path chart for minimal tree-pair diagram representative of $y_0$ in $F(2,3)$

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle 1, 0\rangle$ | $\langle 0, 1\rangle$ |
| 1 | $\langle 2, 0\rangle$ | $\langle 0, 1\rangle$ |
| 2 | $\langle 2, 0\rangle$ | $\langle 0, 1\rangle$ |

Table 10.2: Leaf-path chart for minimal tree-pair diagram representative of $y_0^n$ in $F(2,3)$

| leaf index number | negative valence | positive valence |
|---|---|---|
| 0 | $\langle v_2(l_0 \in T_-^{n-1}) + 1, 0\rangle$ | $\langle 0, n\rangle$ |
| . | . | . |
| . | . | . |
| . | . | . |
| $3^{n-1} - 1$ | $\langle v_2(l_{\lfloor \frac{3^{n-1}-1}{2} \rfloor} \in T_-^{n-1}) + 1, 0\rangle$ | $\langle 0, n\rangle$ |
| $3^{n-1}$ | $\langle v_2(l_{\lfloor \frac{3^{n-1}}{2} \rfloor} \in T_-^{n-1}) + 2, 0\rangle$ | $\langle 0, n\rangle$ |
| $3^{n-1} + 1$ | $\langle v_2(l_{\lfloor \frac{3^{n-1}+1}{2} \rfloor - 1} \in T_-^{n-1}) + 2, 0\rangle$ | $\langle 0, n\rangle$ |
| $3^{n-1} + 2$ | $\langle v_2(l_{\lfloor \frac{3^{n-1}+1}{2} \rfloor} \in T_-^{n-1}) + 2, 0\rangle$ | $\langle 0, n\rangle$ |
| . | . | . |
| . | . | . |
| . | . | . |
| $3^n - 1$ | $\langle v_2(l_{3^{n-1}-1} \in T_-^{n-1}) + 2, 0\rangle$ | $\langle 0, n\rangle$ |

We multiply $y_0$ by $y_0^n$; in order to perform this multiplication, we must make the negative column of the chart for $y_0$ identical to the positive column of $y_0^n$ through leaf-path chart addition, keeping in mind that any leaf-path chart addition which we perform on a row present in one column of a leaf-path chart must be performed on that same row of all columns of that chart simultaneously. First, we notice that all the positive 3–valences of $y_0^n$ are $n$, whereas all the negative 3–valences of $y_0$ are 0. So we must perform 3–ary addition on every row in the leaf-path chart of $y_0^n$ $n$ times; this will multiply the number of rows in $y_0$ by $3^n$, producing $3^{n+1}$ rows total in the resulting chart for $y_0$, which we will now refer to as $C(y_0)'$. Clearly

$$\mathbf{v}(l_i^{\pm} \in C(y_0)') = \langle v_2(l_i^{\pm} \in C(y_0)), n + v_3(l_i^{\pm} \in C(y_0))\rangle \text{ for all } i \in \{0, ..., 3^{n+1}\}$$

so that by the inductive hypothesis

$$\mathbf{v}(l_i^{+} \in C(y_0)') = \langle 0, n+1\rangle$$

Now we notice that all the negative 2–valences in $C(y_0)'$ are at least 1, whereas all of the positive 2–ary valences of $C(y_0^n)$ are 0, so we perform 2–ary addition to every row of $C(y_0^n)$, yielding a chart with $2 \cdot 3^n$ rows which we will refer to as $C(y_0^n)_1$. Now we notice that while the first $3^n$ rows (those numbered $0, ..., 3^n - 1$) of $C(y_0^n)_1$ have the same positive valences as the first $3^n$ rows of $C(y_0)'$, rows $3^n$ through $2 \cdot 3^n - 1$ have a positive 2–valence which is one less than the negative 2–valences of the rows numbered $3^n$ through $2 \cdot 3^n - 1$ in $C(y_0)'$, so we must perform 2–ary addition on these rows in $C(y_0^n)_1$; this results in a tree with $2 \cdot 3^n + 3^n = 3^{n+1}$ rows which we will refer to as $C(y_0^n)'$. Clearly

$$v_2(l_i^{\pm} \in C(y_0^n)') = \begin{cases} v_2(l^{\pm}_{\lfloor \frac{i}{2} \rfloor} \in C(y_0^n)) + 1 & \text{for } i \in \{0, ..., 3^n - 1\} \\ v_2(l^{\pm}_{\lfloor \frac{i}{2} \rfloor} \in C(y_0^n)) + 2 & \text{for } i = 3^n \\ v_2(l^{\pm}_{\lfloor \frac{i}{2} \rfloor - 1} \in C(y_0^n)) + 2 & \text{for } i = 3^n + 1 \\ v_2(l^{\pm}_{\lfloor \frac{i-(3^n+2)}{4} \rfloor + \lfloor \frac{3^n+1}{2} \rfloor} \in C(y_0^n)) + 2 & \text{for } i \in \{3^n + 2, ..., 3^{n+1} - 1\} \end{cases}$$

and

$$v_3(l_i^{\pm} \in C(y_0^n)') = v_3(l_i^{\pm} \in C(y_0^n)) \text{ for } i \in \{0, ..., 3^{n+1} - 1\}$$

so by the inductive hypothesis

$$v_2(l_i^- \in C(y_0^n)') = \begin{cases} v_2(l_{\lfloor \frac{i}{2} \rfloor} \in T_-^n) + 1 & \text{for } i \in \{0, ..., 3^n - 1\} \\ v_2(l_{\lfloor \frac{i}{2} \rfloor} \in T_-^n) + 2 & \text{for } i = 3^n \\ v_2(l_{\lfloor \frac{i}{2} \rfloor - 1} \in T_-^n) + 2 & \text{for } i = 3^n + 1 \\ v_2(l_{\lfloor \frac{i-(3^n+2)}{4} \rfloor + \lfloor \frac{3^n+1}{2} \rfloor} \in T_-^n) + 2 & \text{for } i \in \{3^n + 2, ..., 3^{n+1} - 1\} \end{cases}$$

and

$$v_3(l_i^- \in C(y_0^n)') = 0$$

Since the negative column of $C(y_0)'$ is equal to the positive column of $C(y_0^n)'$, we can form the product of $y_0$ and $y_0^n$ by taking the negative column of $C(y_0^n)'$ as the negative column of $C(y_0^{n+1})$ and the positive column of $C(y_0)'$ as the positive column of $C(y_0^{n+1})$. So our inductive hypothesis holds for all $n \in \mathbb{N}$.

Since the number of rows in our leaf-path chart for $y_0^{n+1}$ will be $3^{n+1}$, this means that the number of leaves in the corresponding reduced tree-pair diagram will be $3^{n+1}$. (It is clear that this tree-pair diagram will be reduced, because all of the carets in its negative tree will be 2–ary and all of the carets in its positive tree will be 3-ary.) So our inductive hypothesis holds for all $n \in \mathbb{N}$.

Since the number of leaves in the reduced tree-pair diagram for $y_0^n$ grows exponentially as $n$ grows linearly and $|y_0^n|_{\{x_0, x_1, y_0, y_1\}} = n$, the metric on $F(2, 3)$ cannot be quasi-isometric to the number of leaves in the minimal tree-pair diagram representatives of elements of $F(2, 3)$.

If we let   denote a quasi-isometry between two metrics, then clearly if $d_X$ $d_Y$ and $d_Y$ $d_Z$ for some metrics $d_X$, $d_Y$ and $d_Z$ on the same metric space, then $d_X$ $d_Z$. This property of quasi-isometry, and the fact that the metric is not quasi-isometric with respect to the number of leaves in minimal tree-pair diagram representatives of elements of $F(2,3)$, along with Remark 10.1, implies that the metric is not quasi-isometric with respect to the number of carets in minimal tree-pair diagram representatives of elements of $F(2,3)$. □

We now generalize this idea to produce a lower bound for the metric on $F(2,3)$. We begin with a lemma about the maximum number of leaves possible of a minimal tree-pair diagram of a representative of an element of $F(2,3)$ which may be obtained as a product of another element and a generator, because the lower bound which we will obtain for the length of a given element $x$ will be given in terms of the number of leaves in the minimal tree-pair diagram representative of $x$.

**Lemma 10.3.** *Let $M(L_x)$ denote the maximum possible value of $L_{xg} - L_x$ (where $L_x$ denotes the number of leaves in the minimal tree-pair diagram representative of $x$) when $g^{\pm 1} \in \{x_0, x_1, y_0, y_1\}$ for arbitrary $x$ in $F(2,3)$. Suppose $M(L_x) \leq aL_x + b$ for some constants $a \in \mathbb{N}$, $b \in \mathbb{Z}$. Now consider $xg_1 \cdots g_n$ where $g_1^{\pm 1}, ..., g_n^{\pm 1} \in \{x_0, x_1, y_0, y_1\}$. Then*

$$L_{xg_1 \cdots g_n} \leq (1+a)^n L_x + \frac{b}{a}((1+a)^n - 1)$$

*Proof.* We prove this lemma by induction. By definition,

$$L_{xg_1 \cdots g_n} = L_{xg_1 \cdots g_{n-1}} + M(L_{xg_1 \cdots g_{n-1}})$$

Our inductive hypothesis is

$$L_{xg_1 \cdots g_n} \leq (1+a)^n L_x + \frac{b}{a}((1+a)^n - 1)$$

We have $M(L_x) = aL_x + b$, so

$$L_{xg_1} = L_x + M(L_x) = L_x + aL_x + b = (a+1)L_x + b$$

So the inductive hypothesis holds for $n = 1$. Now we suppose that it holds for arbitrary $n - 1$:

$$L_{xg_1 \cdots g_{n-1}} \leq (1+a)^{n-1} L_x + \frac{b}{a}((1+a)^{n-1} - 1)$$

Then

$$\begin{aligned}
L_{xg_1\cdots g_n} &\leq L_{xg_1\cdots g_{n-1}} + M(L_{xg_1\cdots g_{n-1}}) \\
&\leq (1+a)^{n-1}L_x + \frac{b}{a}((1+a)^{n-1}-1) \\
&\quad +a((1+a)^{n-1}L_x + \frac{b}{a}((1+a)^{n-1}-1)) + b \\
&= (1+a)((1+a)^{n-1}L_x + \frac{b}{a}((1+a)^{n-1}-1)) + b \\
&= (1+a)^n(L_x + \frac{b}{a}((1+a)^n - (1+a)) + b \\
&= (1+a)^nL_x + \frac{b}{a}((1+a)^n - 1) - \frac{b}{a}\cdot a + b \\
&= (1+a)^nL_x + \frac{b}{a}((1+a)^n - 1)
\end{aligned}$$

And therefore the inductive hypothesis holds for arbitrary $n$. □

**Theorem 10.4.** *For all elements $x \in F(2,3)$ where $|x|$ denotes the length of $x$ in $F(2,3)$ with respect to the standard finite presentation and $L_x$ denotes the number of leaves in the minimal tree-pair diagram representative of $x$*

$$|x|_{\{x_0,x_1,y_0,y_1\}} \geq \log_8 L_x + \left(\log_8 \frac{7}{39} + 1\right)$$

*Proof.* In order to construct the proof for this lower bound on the metric, we consider what could happen when we multiply an arbitrary non-trivial element $x \in F(2,3)$ by a generator in the standard finite presentation. We let $(T_-, T_+)$ denote a minimal tree-pair diagram representative of $x$. When computing the product $xg$ for $g^{\pm} \in \{x_0, x_1, y_0, y_1\}$, we will have to make the positive tree of the minimal tree-pair diagram for the generator identical to $T_-$ by adding carets to both tree-pair diagrams. For the rest of this proof, we will let $L$ denote the number of leaves in $T_-$. We note that for non-trivial $x$, $L \geq 3$.

1. If $g = x_0$, then we categorize $T_-$ as one of the 4 forms given in Figure 10.1. $T_-$ can always be written in one of the 4 forms given in Figure 10.1 without adding any carets.

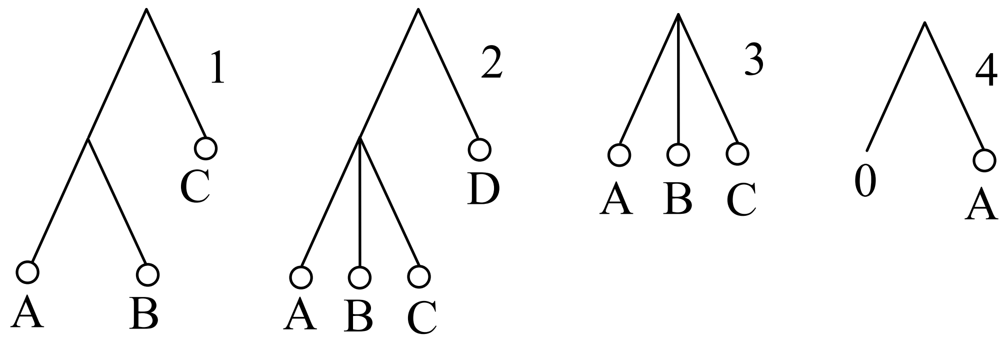


Figure 10.1: The four possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_0$

(a) If $T_-$ can be written in form 1, then no carets need be added to $(T_-, T_+)$ in order for the multiplication with $x_0$ to take place.

(b) If $T_-$ cannot be written in form 1 but can be written in form 2, then there is at least one leaf in subtree $A$, $B$, or $C$ with 2–valence equal to 1 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1). Any leaf in subtrees $A$, $B$, or $C$ with 2–valence equal to 1 must have 2–ary addition performed on it, because by Theorem 7.1.4 the root caret of the subtree whose root is the left child of the root of the tree can be written as a 2–ary caret if and only if every leaf in that subtree has 2–valence at least 1 within that subtree (i.e. 2–valence at least 2 in $T_-$); the left child of the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves in subtrees $A$, $B$, or $C$ with 2–valence 1, $T'_-$, the tree resulting from the addition, can be written in form 1 without adding any further carets. We let $M$ be the number of leaves in subtrees $A$, $B$, or $C$ of $T_-$

with 2–valence equal to 1, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $M$, and $1 \leq M \leq L-1$.

(c) If $T_-$ cannot be written in form 1 or 2 but can be written in form 3, then there is at least one leaf in subtree $A$, $B$, or $C$ with 2–valence equal to 0 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1 or 2). Any leaf in subtrees $A$, $B$, or $C$ with 2–valence equal to 0 must have 2–ary addition performed on it, because by Theorem 7.1.4 the root caret of the tree can be written as a 2–ary caret if and only if every leaf in the tree has 2–valence at least 1; the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves with 2–valence 0, $T'_-$, the tree resulting from the addition, can be written in form 1 or 2 without adding further carets. We let $N$ symbolize the number of leaves in subtrees $A$, $B$, or $C$ with 2–valence equal to 0, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $N$, and $1 \leq N \leq L$. Then, depending upon whether $T'_-$ is of form 1 or 2, the number of leaves added in total to $(T_-, T_+)$ will be $N+M$ where $M$ is the number of leaves in subtrees $A$, $B$, or $C$ with 2–valence equal to 1 once $T'_-$ has been rewritten in form 2, and $0 \leq M \leq (L+N)-1$ ($M = 0$ if $T'_-$ is in form 1 and $1 \leq M \leq (L+N)-1$ if $T'_-$ is in form 2). So the total number of leaves added to $(T_-, T_+)$ will be $N+M$ where $1 \leq N+M \leq 2L+N-1 \leq 3L-1$.

(d) If $T_-$ cannot be written in form 1, 2, or 3, then it must be able to be written in form 4. In this case, only one caret needs to be added to $(T_-, T_+)$ in order for composition to be performed; a 2–ary caret added as the left child of the root. So the total number of leaves added to $(T_-, T_+)$ will be 1.

So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xx_0$ to be performed must be less than or equal to $3L - 1$.

2. If $g = x_1$, then we categorize $T_-$ as one of the 6 forms given in Figure 10.2. $T_-$ can always be written in one of the 6 forms given in Figure 10.2 without adding any carets.

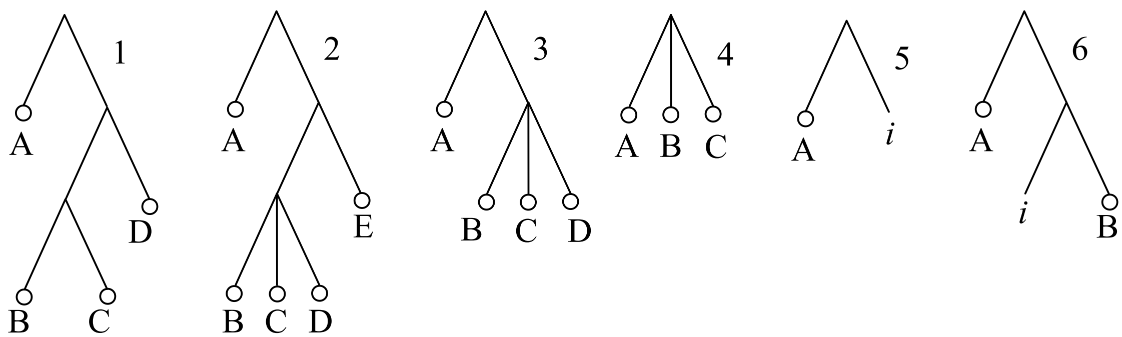


Figure 10.2: The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_1$

(a) If $T_-$ can be written in form 1, then no carets need be added to $(T_-, T_+)$ in order for the multiplication with $x_1$ to take place.

If $T_-$ cannot be written in form 1 but can be written in form 2, then there is at least one leaf in subtree $B$, $C$ or $D$ with 2–valence equal to 2 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1). Any leaf in subtrees $B$, $C$ or $D$ with 2–valence equal to 2 must have 2–ary addition performed on it, because by Theorem 7.1.4 the root caret of the subtree whose root is the left child of the right child of the root of $T_-$ can be written as a 2–ary caret if and only if every leaf in that subtree has 2–valence at least 1 within that subtree (i.e. 2–valence at least 3 in $T_-$); the left child of the right child of the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves in subtrees $B$, $C$, or $D$ with 2–valence 2 within $T_-$, $T'_-$, the tree resulting from the addition, can be written in form 1 without adding further carets. We let $M$ be the number of leaves in subtrees $B$, $C$ or $D$ with 2–valence equal to 2, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $M$, and $1 \leq M \leq L - 2$.

(b) If $T_-$ cannot be written in form 1 or 2 but can be written in form 3, then there is at least one leaf in subtree $B$, $C$, or $D$ with 2–valence equal to 1 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1 or 2). Any leaf in subtrees $B$, $C$, or $D$ with 2–valence equal to 1 must have 2–ary addition performed on it, because by Theorem 7.1.4 the right child of the root caret of $T_-$ can be written as a 2–ary caret if and only if every leaf in the subtree whose root is the right child of the root of $T_-$ has 2–valence at least 1 within that subtree (i.e. 2–valence at least 2 within $T_-$); the right child of the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves with valence 1 in subtrees $B$, $C$, or $D$, $T^1_-$, the tree resulting from the addition, can be written in form 1 or 2 without adding further carets. We let $N$ symbolize the number of leaves in subtrees $B$, $C$, or $D$ with 2–valence equal to 1, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $N$, and $1 \leq N \leq L-1$. Then, depending upon whether $T^1_-$ is of form 1 or 2, the number of leaves added in total to $(T_-, T_+)$ will be $N+M$ where $M$ is the number of leaves in subtrees $B$, $C$, or $D$ with 2–valence equal to 1 once $T^1_-$ has been rewritten in form 2, and $0 \leq M \leq (L + N) - 2$

(where $M = 0$ if $T'_-$ is in form 1 and $1 \leq M \leq L + N - 2$ if $T'_-$ is in form 2). So the total number of leaves added to $(T_-, T_+)$ will be $N + M$ where $1 \leq N + M \leq 2L + N - 3 \leq 3L - 4$.

(c) If $T_-$ cannot be written in form 1, 2, or 3 but can be written in form 4, then there is at least one leaf in subtree $A$, $B$, or $C$ with 2–valence equal to 0 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1, 2 or 3). Any leaf in subtrees $A$, $B$, or $C$ with 2–valence equal to 0 must have 2–ary addition performed on it, because by Theorem 7.1.4 the root caret of the tree can be written as a 2–ary caret if and only if every leaf in the tree has 2–valence at least 1; the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves with valence 0 in subtrees $A$, $B$, or $C$, $T^1_-$, the tree resulting from the addition, can be written in form 1, 2 or 3 without adding further carets. We let $N$ symbolize the number of leaves in subtrees $A$, $B$, or $C$ with 2–valence equal to 1, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $N$, and $1 \leq N \leq L$. Then, depending upon whether $T^1_-$ is of form 1, 2 or 3, the number of leaves added in total to $(T_-, T_+)$ will be $N + M + P$ where $M$ is the number of leaves in subtrees $B$, $C$, or $D$ with 2–valence equal to 1 once $T^1_-$ has been rewritten in form 3 and $0 \leq M \leq (L+N)-1$ (where $M+0$ if $T^1_-$ is in form 1 or 2 and $1 \leq M \leq (L+N)-1$ if $T^1_-$ is in form 3) and where $T^2_-$ will denote the result of any addition performed on $T^1_-$ when it is in form 3, and similarly, $P$ is the number of leaves in subtrees $B$, $C$, or $D$ with 2–valence equal to 2 once $T^1_-$ has been rewritten in form 2 and $0 \leq P \leq (L + N + M) - 2$ (where $P = 0$ if $T^1_-$ or $T^2_-$ is in form 1 and $0 \leq P \leq (L + N + M) - 2$ if $T^1_-$ or $T^2_-$ is in form

2). So the total number of leaves added to $(T_-, T_+)$ will be $N + M + P$ where $1 \leq N + M + P \leq 3L + 2N + M - 3 \leq 7L - 4$.

(d) If $T_-$ cannot be written in form 1, 2, 3, or 4, then it must be able to be written in form 5 or 6. In this case, only 2 or 1 carets respectively need to be added to $(T_-, T_+)$ in order for composition to be performed; a 2–ary caret added as the right child of the root in form 5 and a 2–ary caret added as the left child of the right child of the root in both form 5 and form 6. So the total number of leaves added to $(T_-, T_+)$ will be 1 or 2.

So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xx_1$ to be performed must be less than or equal to $7L - 4$.

3. If $g = y_0$, then we categorize $T_-$ as one of the 2 forms given in Figure 10.3. $T_-$ can always be written in one of the 2 forms given in Figure 10.3 without adding any carets.

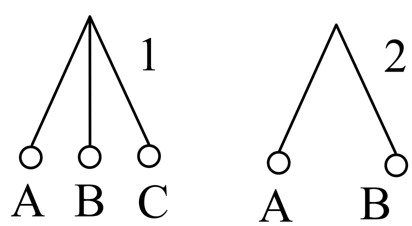


Figure 10.3: The two possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xy_0$

(a) If $T_-$ can be written in form 1, then no carets need be added to $(T_-, T_+)$ in order for the multiplication with $y_0$ to take place.

(b) If $T_-$ cannot be written in form 1, then it must be able to be written in form 2, and there is at least one leaf in subtree $A$ or $B$ with 3–valence equal to 0 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1). Any leaf in subtrees $A$ or $B$ with 3–valence equal to 0 must have 3–ary addition performed on it, because by Theorem 7.1.4 the root of the tree can be written as a 3–ary caret if and only if every leaf in that subtree has 3–valence at least 1; the root must be rewritten as a 3–ary caret in order for composition to take place. Once this is done for all leaves in subtrees $A$ and $B$ with 3–valence 0, $T'_-$, the tree resulting from the addition, can be written in form 1 without adding any further carets. We let $M$ be the number of leaves in subtrees $A$ or $B$ of $T_-$ with 3–valence equal to 0, so that the total number of leaves added to $(T_-, T_+)$ during this process of 3–ary addition is $2M$, and $1 \leq M \leq L$ which implies $2 \leq 2M \leq 2L$.

So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xy_0$ to be performed must be less than or equal to $2L$.

4. If $g = y_1$, then we categorize $T_-$ as one of the 6 forms given in Figure 10.4. $T_-$ can always be written in one of the 6 forms given in Figure 10.4 without adding any carets.

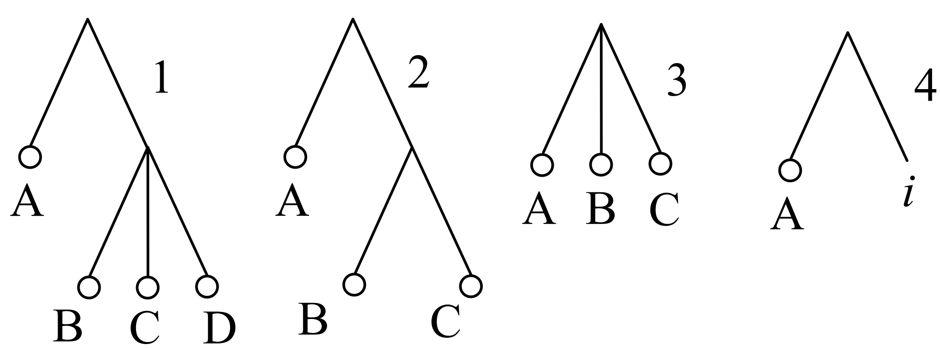


Figure 10.4: The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xy_1$

(a) If $T_-$ can be written in form 1, then no carets need be added to $(T_-, T_+)$ in order for the multiplication with $y_1$ to take place.

(b) If $T_-$ cannot be written in form 1 but can be written in form 2, then there is at least one leaf in subtree $B$ or $C$ with 3–valence equal to 0 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1). Any leaf in subtrees $B$ or $C$ with 3–valence equal to 0 must have 3–ary addition performed on it, because by Theorem 7.1.4 the root caret of the subtree whose root is the right child of the root of $T_-$ can be written as a 3–ary caret if and only if every leaf in that subtree has 3–valence at least 1; the right child of the root must be rewritten as a 3–ary caret in order for composition to take place. Once this is done for all leaves in subtrees $B$ or $C$ with 3–valence 0, $T'_-$, the tree resulting from this addition, can be written in form 1 without adding further carets. We let $M$ be the number of leaves in subtrees $B$ or $C$ with 3–valence equal to 0 so that the total number of leaves added to $(T_-, T_+)$ during this process of 3–ary addition is $2M$, and $1 \leq M \leq L-1$ so that $2 \leq 2M \leq 2L-2$.

(c) If $T_-$ cannot be written in form 1 or 2 but can be written in form 3, then there is at least one leaf in subtree $A$, $B$, or $C$ with 2–valence equal to 0 (because if there were not, by Theorem 7.1.4, we would be able to write $T_-$ in form 1 or 2). Any leaf in subtrees $A$, $B$, or $C$ with 2–valence equal to 0 must have 2–ary addition performed on it, because by Theorem 7.1.4 the root caret of the tree can be written as a 2–ary caret if and only if every leaf in the tree has 2–valence at least 1; the root must be rewritten as a 2–ary caret in order for composition to take place. Once this is done for all leaves with valence 0 in subtrees $A$, $B$, or $C$, $T^1_-$, the tree resulting from the addition, can be written in form 1 or 2 without adding further carets. We let $N$ symbolize the number of leaves in subtrees $A$, $B$, or $C$ with 2–valence equal to 0, so that the total number of leaves added to $(T_-, T_+)$ during this process of 2–ary addition is $N$, and $1 \leq N \leq L$. Then, depending upon whether $T_-^1$ is of form 1 or 2, the number of leaves added in total to $(T_-, T_+)$ will be $N + 2M$ where $M$ is the number of leaves in subtrees $B$ or $C$ with 3–valence equal to 0 once $T_-^1$ has been rewritten in form 2 and $0 \leq M \leq (L+N) - 1$ (where $M = 0$ if $T_-^1$ is in form 1 and $1 \leq M \leq (L+N) - 1$ if $T_-^1$ is in form 2). So the total number of leaves added to $(T_-, T_+)$ will be $N + 2M$ where $1 \leq N + 2M \leq 3L - 2N - 2 \leq 5L - 2$.

(d) If $T_-$ cannot be written in form 1, 2, or 3, then it must be able to be written in form 4. In this case, only 1 caret needs to be added to $(T_-, T_+)$ in order for composition to be performed; a 3–ary caret added as the right child of the root. So the total number of leaves added to $(T_-, T_+)$ will be 2.

So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xy_1$ to be performed must be less than or equal to $5L - 2$.

If $g = x_0^{-1}$, then we categorize $T_-$ as on of the 4 forms given in Figure 10.1. $T_-$ can always be written in one of the 4 forms given in Figure 10.5 without adding any carets. The argument for this case is almost identical to that for $x_0$: the only change is that the location of all leaves from the left to the right is reversed, as we can see by looking at the 4 possible forms for both cases $x_0$ and $x_0^{-1}$, so we omit the details here. So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xx_0^{-1}$ to be performed must be less than or equal to $3L - 1$, just as in the case of the multiplication $xx_0$.

5. If $g = x_1^{-1}$, then we categorize $T_-$ as one of the 6 forms given in Figure 10.6. $T_-$ can always be written in one of the 6 forms given in Figure 10.6 without

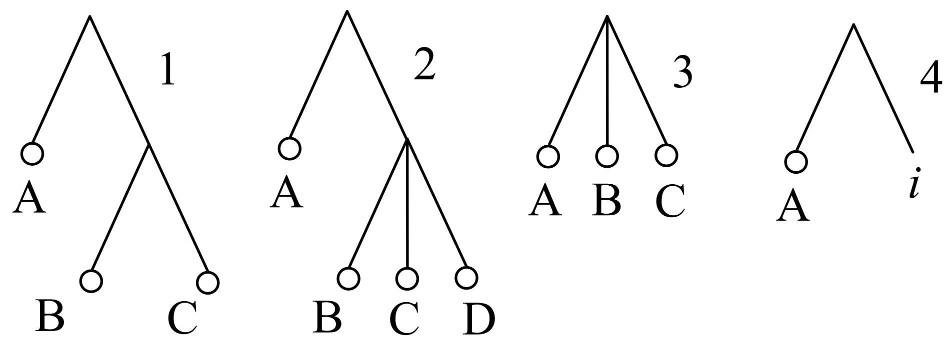


Figure 10.5: The four possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_0^{-1}$ or $xy_0^{-1}$

adding any carets. The argument for this case is almost identical to that for $x_1$: the only diffeence is a shift to the right or left of the bottommost set of subtrees in some cases, as we can see by looking at the 6 possible forms for both cases $x_1$ and $x_1^{-1}$, so we omit the details here. So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xx_1^{-1}$ to be performed must be less than or equal to $7L - 4$, just as in the case of the multiplication $xx_1$.

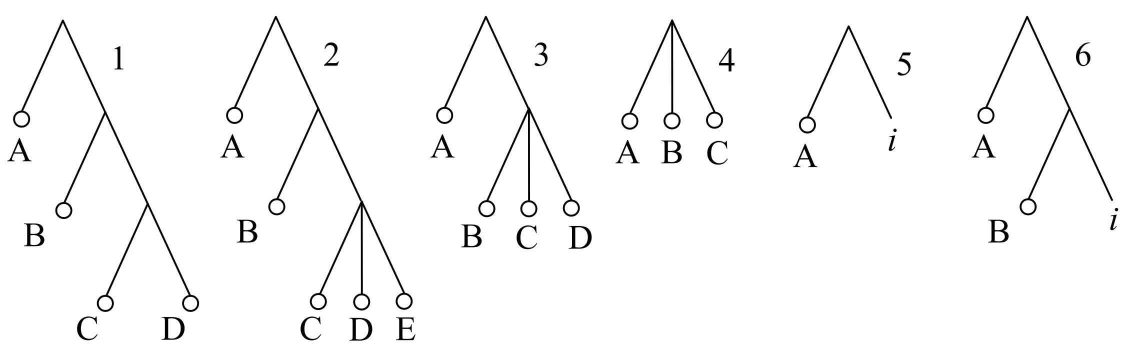


Figure 10.6: The six possible forms of $T_-$ in the minimal tree-pair diagram representative of $x = (T_-, T_+)$, where $x$ is in the product $xx_1^{-1}$ or $xy_1^{-1}$

6. If $g = y_0^{-1}$, then then we categorize $T_-$ as one of the 6 forms given in Figure 10.5. $T_-$ can always be written in one of the 4 forms given in Figure 10.5

without adding any carets. This is exactly the same case as for $x_0^{-1}$ because the minimal tree-pair diagram representatives for $x_0^{-1}$ and $y_0^{-1}$ have identical positive trees, so we omit the details here. So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xy_0^{-1}$ to be performed must be less than or equal to $3L-1$, just as in the case of the multiplication $xx_0^{-1}$.

7. If $g = y_1^{-1}$, then then we categorize $T_-$ as one of the 6 forms given in Figure 10.6. $T_-$ can always be written in one of the 7 forms given in Figure 10.6 without adding any carets. This is exactly the same case as for $x_1^{-1}$ because the minimal tree-pair diagram representatives for $x_1^{-1}$ and $y_1^{-1}$ have identical positive trees, so we omit the details here. So the number of leaves that must be added to $(T_-, T_+)$ in order for the multiplication $xy_1^{-1}$ to be performed must be less than or equal to $7L-4$, just as in the case of the multiplication $xx_1^{-1}$.

We have now enumerated all possible alterations to the number of leaves in an arbitrary non-trivial element $x \in F(2,3)$ when it is multiplied by a generator or its inverse. So we have enumerated all possible alterations to the number of leaves in an arbitrary non-trivial element $x \in F(2,3)$ when its length is increased by one. Therefore, the maximal possible number of leaves added to $x$ when its length increases by 1 is bounded from above by $7L-4$. So, by Lemma 10.3, the maximal possible number of leaves in the reduced product $xg_1 \cdots g_n$ if $|x|_{\{x_0,x_1,y_0,y_1\}} = 1$ (where $g_i^{\pm} \in \{x_0, x_1, y_0, y_1\}$ for all $i \in \{1, ..., n\}$) would be bounded from above by $8^n L_x + \frac{4}{7}(8^n - 1)$. Since $L = 3, 4$, or $5$ for all $x$ such that $|x|_{\{x_0,x_1,y_0,y_1\}} = 1$, we can write this upper bound even more generally as $\frac{39}{7} \cdot 8^n - \frac{4}{7}$. So for any $x$ in $F(2,3)$,

$$L_x \leq \frac{39}{7} \cdot 8^n - \frac{4}{7}$$

where $|x|_{\{x_0,x_1,y_0,y_1\}} = n + 1$

Rewriting this inequality yields

$$\log_8\left(\frac{7}{39}L_x + \frac{4}{39}\right) \leq \log_8(8^{|x|_{\{x_0,x_1,y_0,y_1\}}-1})$$

which leads to

$$\begin{aligned} |x|_{\{x_0,x_1,y_0,y_1\}} &\geq \log_8\left(\frac{7}{39}L_x + \frac{4}{39}\right) + 1 \\ &\geq \log_8\left(\frac{7}{39}L_x\right) + 1 \\ &= \log_8 L_x + \left(\log_8 \frac{7}{39} + 1\right) \end{aligned}$$

So we have as a lower bound for the length of arbitrary $x$ in $F(2,3)$:

$$|x|_{\{x_0,x_1,y_0,y_1\}} \geq \log_8 L_x + \left(\log_8 \frac{7}{39} + 1\right)$$

□

**Corollary 10.5.** *The order of the lower bound given in Theorem 10.4 is sharp.*

*Proof.* This follows immediately from the proof of Theorem 10.2:

$$L_{y_0^n} = 3^{|y_0^n|_{\{x_0,x_1,y_0,y_1\}}}$$

and therefore

$$|y_0^n|_{\{x_0,x_1,y_0,y_1\}} = \log_3(L_{y_0^n})$$

□

Now we proceed to find an upper bound for the metric on the standard finite generating set in $F(2,3)$. We will find that the upper bound is linear with respect to the number leaves in the minimal tree-pair diagram representative. However, before we can establish such a result, we will need several lemmas. First we will need to establish a way to calculate the number of leaves in a minimal tree-pair diagram representative for a positive word, and then we will relate this to the length of that word with respect to the standard infinite generating set. Finally we will extend this result to a bound on the length of the positive element in a finite generating set so that we can bound the length in the standard finite generating set.

For the duration of this section, when we refer to the *right side* of a tree, we mean the subtree consisting of the root caret and all right carets.

We will begin by considering only positive words in $F(2,3)$; this will then allow us to generalize our results to all words in $F(2,3)$. We look at positive words first because they have a simpler form; when considering positive words, since the negative tree is trivial with respect to its contribution to the normal form, we can limit our focus to a single tree: the positive tree of the minimal tree-pair diagram representative of the word.

Our proof of the upper bound of the metric will use the normal from which was developed in the previous chapter. We recall that the normal form of a positive word $w$ will be of the following form:

$$NF(w) = \mathcal{B}_0 \cdots \mathcal{B}_k$$

where

$$\mathcal{B}_0 = y_{\beta_1} \cdots y_{\beta_N}$$

such that $\beta_{i+1} > \beta_i + 1$ for all $i \in \{1, ..., N\}$ and where

$$\mathcal{B}_i = (\gamma^{i,1}_{\alpha i,1})^{e_{i,1}} \cdots (\gamma^{i,s_i}_{\alpha i,s_i})^{e_{i,s_i}}$$

such that

$$\alpha_{1,j} \neq 0 \text{ for all } j \in \{1, ..., s_1\}$$

and

$$\alpha_{i,1} = 0 \text{ and } e_{i,1} = 1 \text{ for all } i \in \{2, ..., k\}$$

and

$$\alpha_{i,j} \neq 0 \text{ for all } j \in \{2, ..., s_i\} \text{ and } i \in \{2, ..., k\}$$

and where for all $i \in \{2, ..., k\}$

$$\alpha_{i,j} \leq \sum_{l=1}^{j-1} e_{i,l} \cdot \left(\omega\left(\gamma_{\alpha_{i,l}}^{i,l}\right) - 1\right)$$

for all $j \in \{1, ..., s_i\}$.

We have simplified the way in which we wrote our indices in the previous chapter for the sake of simplicity, but otherwise this is the same normal form introduced in Theorem 9.10.

**Remark 10.6.** *The total number of $x_i$ and $z_i$ generators (for all $i \in \mathbb{Z}^*$) present in the normal form expression for a positive word in $F(2,3)$ is equal to the number of carets in the positive tree of the minimal tree-pair diagram representative which are not on the right side of the tree. Or, more formally, if we let $N_x$ and $N_z$ denote the number of $x_i$ and $z_i$ generators respectively in the normal form for any $i \in \mathbb{Z}^*$, then:*

$$N_x = \sum_{j=1}^{k} \sum_{\{i|\gamma_{\alpha_{j,i}}^{j,i} = x_{\alpha_{j,i}}\}} e_{j,i}$$

*for some $j \in \mathbb{Z}^*$ and*

$$N_z = \sum_{j=1}^{k} \sum_{\{i|\gamma_{\alpha_{j,i}}^{j,i} = z_{\alpha_{j,i}}\}} e_{j,i}$$

*for some $j \in \mathbb{Z}^*$.*

*Proof.* This is obvious if we consider how each generator in the expression of the normal form is derived from the minimal tree-pair diagram representative. □

So we have calculated the total number of carets in the positive tree of the minimal tree-pair diagram representative for a positive word $w$. Now we calculate the number of leaves in the right side of the positive tree in the minimal tree-pair diagram representative for a positive word $w$. This will allow us to calculate the total number of carets or leaves in the positive tree.

**Lemma 10.7.** *We consider a positive word $w$ in $F(2,3)$. We recall that the indices of the generators in $\mathcal{B}_1$ of the normal form expression for $w$ are denoted by $\alpha_{1,j}$ where $j = 1, ..., s_1$, and let $n_{x_{\alpha_{1,j}}}$ and $n_{z_{\alpha_{1,j}}}$ denote the number of $x_{\alpha_{1,j}}$ and $z_{\alpha_{1,j}}$ generators respectively which are present in the block. More formally, we define:*

$$n_{x_{\alpha_{1,j}}} = \sum_{\{i|\alpha_{1,i}=\alpha_{1,j}\wedge\gamma^{1,i}_{\alpha_{1,i}}=x_{\alpha_{1,i}}\}} e_{1,i}$$

*and*

$$n_{z_{\alpha_{1,j}}} = \sum_{\{i|\alpha_{1,i}=\alpha_{1,j}\wedge\gamma^{1,i}_{\alpha_{1,i}}=z_{\alpha_{1,i}}\}} e_{1,i}$$

*Let $\alpha_{1,M}$ be the largest index of a generator in $\mathcal{B}_1$ of the normal form expression which satisfies the following inequality:*

$$\alpha_{1,M} - \alpha_{1,M-1} > n_{x_{\alpha_{1,M-1}}} + 2n_{z_{\alpha_{1,M-1}}}$$

*Let $\beta_N$ be the largest index of a generator present in block zero of the normal form expression for $w$.*

*If*

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) > \beta_N$$

*then*

$$L(R) = \alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) + 2$$

*where $L(R)$ represents the number of leaves in the subtree of the positive tree in the minimal tree-pair diagram representative of $w$ which consists solely of the right side of the tree. If*

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) \leq \beta_N$$

*then*

$$L(R) = \beta_N + 3$$

*Proof.* We begin by observing that since the positive tree in the minimal tree-pair diagram representative for $w$ is paired with a negative tree where the last two leaves are in an exposed 2–ary caret, and since the tree-pair diagram must by definition be reduced, the positive tree in the minimal tree-pair diagram representative for $w$ must have its last leaf in a 2–ary caret which is not exposed (and therefore has a non-empty left subtree) or in a 3–ary caret. We also observe that the last leaf in the tree must always be contained in a caret which is on the right side of the tree. For the duration of the proof, whenever we refer simply to "the tree," we mean the positive tree in the reduced tree-pair diagram representative of $w$.

Also, for the sake of simplicity, throughout this proof, $R$ denotes the subtree of the tree which consists entirely of carets which belong to the right side of the tree, and $S_R$ denotes the subtree which consists of the root caret and all of its descendent carets which are connected to the root caret through its right and middle (if it exists) child nodes. We note that $R$ is also a subtree of $S_R$.

We break this into two cases: when the last caret on the right side of the tree has a nonempty left subtree and is a 2–ary caret, and when the last caret on the right side of the tree is a 3–ary caret.

First we consider the case in which the last caret on the right side of the tree is not a 3–ary caret, and we try to calculate the number of leaves contributed to the total number of leaves in the tree by the carets on the right side of the tree in this case. The highest generator index $\alpha_{1,M}$ in $\mathcal{B}_1$ which satisfies

$$\alpha_{1,M} - \alpha_{1,M-1} > n_{x_{\alpha_{1,M-1}}} + 2n_{z_{\alpha_{1,M-1}}}$$

will correspond to the leaf index number of the leftmost leaf in $S_R$.

To see that this is true, we see that for a given $i \in \mathbb{N}$,

$$\alpha_{1,i} - \alpha_{1,i-1} \leq n_{x_{\alpha_{1,i-1}}} + 2n_{z_{\alpha_{1,i-1}}}$$

if and only if all the carets representing $x_{\alpha_{1,i}}$ and $z_{\alpha_{1,i}}$ generators present in $\mathcal{B}_1$ will be contained in the same left subtree of a caret on the right side of the tree as all the carets representing $x_{\alpha_{1,i-1}}$ and $z_{\alpha_{1,i-1}}$ generators present in $\mathcal{B}_1$. Since all the carets representing $x_{\alpha_{1,i-1}}$ and $z_{\alpha_{1,i-1}}$ present in $\mathcal{B}_1$ would have a lower index than $\alpha_{1,i}$ and therefore would have leaves with lower index numbers in that left or middle subtree of a caret on the right side of the tree than the leaves of the carets representing $x_{\alpha_{1,i}}$ and $z_{\alpha_{1,i}}$ generators present in $\mathcal{B}_1$, $\alpha_{1,i}$ cannot be the index number of the first leaf in the left subtree of the last right caret; however, if $\alpha_{1,M} - \alpha_{1,M-1} > n_{x_{\alpha_{1,M-1}}} + 2n_{z_{\alpha_{1,M-1}}}$, then $\alpha_{1,M}$ will correspond to the leaf index number of the leftmost leaf in the left subtree of the last right caret in $S_R$.

Now we consider the subtree $R$ and compare it to the subtree $S_R$. Since every $x_{\alpha_{1,j}}$ or $z_{\alpha_{1,j}}$ in the $\mathcal{B}_1$ of the normal form contributes one 2–ary or 3–ary caret respectively to the subtree $S_R$, each of these $x_{\alpha_{1,j}}$ or $z_{\alpha_{1,j}}$ generators will contribute one or two leaves more respectively in $S_R$ than in $R$ alone. Since $\alpha_{1,M}$ is the index of the first leaf descended from the last right caret, subtracting all of the leaves which were contributed by $x_{\alpha_{1,j}}$ or $z_{\alpha_{1,j}}$ generators when $j < M$ will give the index with respect to the leaf numbering in $R$ of the first leaf in the last right caret of the right side of the tree. Since the leaf index numbers begin with 0, adding 2 to this number yields number of leaves in the right side of the tree if the last right caret is 2–ary. So in this case this yields

$$L(R) = \alpha_{1,M} - \sum_{j=1}^{M-1} (n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) + 2$$

However, if the last right caret in the tree is 3–ary, the highest index of a generator in block zero of the normal form, $\beta_N$, will correspond to the leaf index number with respect to leaf numbering in $R$ of the leftmost leaf of the last right caret in the right side of the tree; this is clear by the way in which $\mathcal{B}_0$ in the normal form corresponds to the 3–ary leaves in the right side of the tree. Since leaf numbering begins with 0, and the last caret on the right side of the tree is 3–ary, this means that

$$L(R) = \beta_N + 3$$

□

Now we include a quick remark which will be needed in order to prove the next lemma.

**Remark 10.8.** *The number of $y_i$ generators present in $\mathcal{B}_0$ of the normal form of a positive word $w$ will always be less than or equal to the maximum index of a generator in $\mathcal{B}_0$, unless the only generator in the block is $y_0$, in which case number of $y_i$ generators present in $\mathcal{B}_0$ will be less than or equal to the maximum index of a generator in $\mathcal{B}_0$ plus one. More formally,*

$$N \leq \beta_N + 1$$

*Proof.* This is obvious when we consider that by definition of the normal form, no two generators present in block zero may have the same index, and in fact, two consecutive generators must have indices which are at least 2 apart. □

Now we are ready to establish a linear expression in the number of leaves in the minimal tree-pair diagram representative as an upper bound for length, although at this point we will only be able to do this with respect to the standard infinite generating set.

**Lemma 10.9.** *For a positive word $w$, its length with respect to the standard infinite generating set is*

$$|w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} \leq L(w)$$

*where $L(w)$ denotes the number of leaves in the minimal reduced tree-pair diagram representative for $w$.*

*Proof.* By Lemma 10.6, we can see that the number of leaves contributed to the positive tree of the minimal tree-pair diagram representative of $w$ by those carets which correspond to $x_i$ or $z_i$ generators for any $i \in \mathbb{Z}^*$ in the normal form of $w$ is

$$N_x + 2N_z$$

and from this lemma, we also know that these are the only leaves which come from the addition of carets to the tree which are not on the right side. So adding this to the number of leaves contributed to the tree by carets on the right side given in Lemma 10.7 gives us the total number of leaves in the minimal tree-pair diagram representative for $w$:

$$L(w) = N_x + N_z + \alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) + 2$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) > \beta_N$$

and

$$L(w) = N_x + N_z + \beta_N + 3$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) \leq \beta_N$$

Now we note that

$$n(NF_w) = N_x + N_z + N$$

where $n(NF_w)$ denotes the number of generators present in the normal form for $w$. Clearly

$$|w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} \leq n(NF_w)$$

So

$$|w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} \leq n(NF_w) = N_x + N_z + N \leq N_x + N_z + \beta_N + 3 = L(w)$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1} (n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) \leq \beta_N$$

and

$$\begin{aligned} |w|_{\{x_0,...,y_0,...,z_0,...\}} \leq n(NF_w) &= N_x + N_z + N \\ &\leq N_x + N_z + \beta_N + 1 \\ &\leq N_x + N_z + \alpha_{1,M} - \sum_{j=1}^{M-1} (n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) + 2 = L(w) \end{aligned}$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1} (n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) > \beta_N$$

So:

$$|w|_{\{x_0,...,y_0,...,z_0,...\}} \leq n(NF_w) \leq L(w)$$

□

Now we want to improve the result of Lemma 10.9 so that it (or something similar) holds for lengths in a finite generating set. Before we can establish a result of this kind, we will first need the following lemma.

**Lemma 10.10.** *If $\alpha_{i,M_i}$ is the maximum index of any generator in $\mathcal{B}_i$ (for any integer $i \in \{2, ..., k\}$) in the normal form for a positive word $w$ in $F(2,3)$, then the total number of leaves contributed to the positive tree of the minimal tree-pair diagram representative of $w$ by carets which correspond to the generators present in $\mathcal{B}_i$ in the normal form must be greater than or equal to $\alpha_{i,M_i} + 1$.*

*Proof.* We can see this by looking at the portion of the tree which is contributed by each block. Block $\mathcal{B}_k$ of the normal form will contribute the first caret on the left side of the tree and all of its middle and right children. So

$$L(B_k) \geq \alpha_{k,M_k} + 1$$

where $L(B_i)$ indicates the number of leaves contributed to the minimal tree-pair diagram representative of $w$ by generators present in block number $i$ for any $i \in \mathbb{Z}^*$.

More generally, $\mathcal{B}_i$ of the normal form will contribute the $(k-i+1)$th caret on the left side of the tree and all of its middle and right children. So

$$L(B_i) \geq \alpha_{i,M_i} + 1$$

□

Now we relate the length of a positive word with respect to the standard infinite generating set to the length of that word with respect to a finite generating set.

**Lemma 10.11.** *For any positive word $w$ in $F(2,3)$:*

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} \leq |w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} + 2(\beta_N + \alpha_{1,M} + \alpha_{2,M_2} + \alpha_{3,M_3} + \cdots \alpha_{k,M_k})$$

*Proof.* This follows immediately from the fact that using the relators to substitute $x_0^{-(i-1)}x_1x_0^{i-1}$ and $x_0^{-(i-1)}z_{i-n}x_0^{i-1}$ for all $x_i$ and $z_i$ respectively in a single block $\mathcal{B}_j$ of $NF(w)$ whenever $i > 1$ and then canceling all adjacent $x_0$s and $x_0^{-1}$s will increase the number of generators in the expression for $\mathcal{B}_j$ by at most $2\alpha_{j,M_j}$, if $\alpha_{j,M_j}$ is the maximum index of a generator in $\mathcal{B}_j$. Similarly, if we also replace all occurrences of $y_i$ for $i > 1$ in $\mathcal{B}_0$ with $x_0^{-(i-1)}y_{i-n}x_0^{i-1}$ using the group relators, this increases the number of generators in the expression by at most $2\beta_N$. Taking the sum over all blocks in the normal form, we obtain

$$2(\beta_N + \alpha_{1,M} + \alpha_{2,M_2} + \alpha_{3,M_3} + \cdots \alpha_{k,M_k})$$

as the upper bound on the total increase in the number of generators present in the expression when we replace all the infinite generators with expressions in the finite generators $\{x_0, x_1, y_0, y_1, z_0, z_1\}$. □

**Theorem 10.12.** *For any positive word $w$ in $F(2,3)$:*

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} < 3L(w)$$

*Proof.* From Lemma 10.11, we have

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} \leq |w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} + 2(\beta_N + \alpha_{1,M} + \alpha_{2,M_2} + \alpha_{3,M_3} + \cdots \alpha_{k,M_k})$$

From Lemma 10.10,

$$2(\beta_N+\alpha_{1,M}+\alpha_{2,M_2}+\alpha_{3,M_3}+\cdots \alpha_{k,M_k}) \leq 2(\beta_N+\alpha_{1,M}+L(B_2)+L(B_3)+\cdots L(B_k))$$

From Remark 10.7 and Lemma 10.10, we can see immediately that:

$$L(B_0) + L(B_1) = \left(\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) + 2\right) + \sum_{j=1}^{k}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}})$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) > \beta_N$$

and

$$L(B_0) + L(B_1) = \beta_N + 3 + \sum_{j=1}^{k}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}})$$

when

$$\alpha_{1,M} - \sum_{j=1}^{M-1}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) \leq \beta_N$$

In both of these cases, by Lemma 10.10,

$$L(B_0) + L(B_1) > \beta_N + \sum_{j=1}^{k}(n_{x_{\alpha_{1,j}}} + 2n_{z_{\alpha_{1,j}}}) \geq \beta_N + \alpha_{1,M}$$

So

$$2(\beta_N+\alpha_{1,M}+L(B_2)+L(B_3)+\cdots L(B_k)) < 2(L(B_0)+L(B_1)+L(B_2)+L(B_3)+\cdots L(B_k)) = 2L(w)$$

This yields

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} \leq |w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} + 2L(w)$$

and since by Lemma 10.9 we we have

$$|w|_{\{x_0,\ldots,y_0,\ldots,z_0,\ldots\}} \leq L(w)$$

this yields

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} \leq 3L(w)$$

□

**Lemma 10.13.** *For any positive word $w$ in $F(2,3)$,*

$$|w|_{\{x_0,x_1,y_0,y_1\}} \leq 24L(w)$$

*Proof.* From Lemma 10.12 we have

$$|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}} < 3L(w)$$

Once our normal form has been rewritten using only finite generators from the set $\{x_0, x_1, y_0, y_1, z_0, z_1\}$, we then replace all occurrences of $z_0$ and $z_1$ with the equivalent $y_1^{-1}y_0x_1x_0$ and $x_0^{-1}y_1^{-1}x_0y_1x_0^{-1}x_1x_0x_1$ respectively, which we obtain from the relators visible in the group presentation. This can make the number of generators in the expression at most 8 times larger so we obtain:

$$|w|_{\{x_0,x_1,y_0,y_1\}} \leq 8|w|_{\{x_0,x_1,y_0,y_1,z_0,z_1\}}$$

So this yields

$$|w|_{\{x_0,x_1,y_0,y_1\}} \leq 24L(w)$$

□

**Theorem 10.14.** *There exists fixed constant $c \in \mathbb{N}$ such that for any word $w$ in $F(2,3)$ (not necessarily positive)*

$$|w|_{\{x_0,x_1,y_0,y_1\}} \leq cL(w)$$

*Proof.* For any word $w$, we can factor it uniquely into the product $w = w_+w_-^{-1}$ where $w_+$ and $w_-$ are both positive words ($w_+$ and $w_-$ are as defined in Lemma 9.1). Clearly

$$|w| \leq |w_+| + |w_-| \leq 24L(w_+) + 24L(w_-) \leq 24(2\max\{L(w_+), L(w_-)\}) = 48L(w)$$

. ☐

**Remark 10.15.** *The order of the upper bound given in Theorem 10.14 is sharp.*

*Proof.* To see this we need only consider any element of $F(2)$ as an element of $F(2,3)$. Since the length of any element of $F(2)$ under the standard finite generating set is the same as its length under the standard finite generating set in $F(2,3)$, and since the metric in $F(2)$ is quasi-isometric to the number of carets in the minimal tree-pair diagram representative, and the number of carets in the minimal tree-pair diagram representative is quasi-isometric to the number of leaves in the minimal tree-pair diagram representative, any word in $F(2)$ considered as a word in $F(2,3)$ will have length which is linear with respect to the number of leaves in the minimal tree-pair diagram representative. To see a specific example, we can consider the word $x_0^n$ for any $n \in \mathbb{N}$: the minimal tree-pair diagram representative for this word will consist of a negative tree consisting of the root with $n$ 2–ary right caret descendants and a positive tree consisting of the root with $n$ 2–ary left caret descendants. The length of this word is clearly $n$, and the number of leaves in the reduced tree-pair diagram representative is $n+2$. So in this particular case, we have $|x_0^n| = L(x_0^n) - 2$. ☐

# Bibliography


[1] James Belk and Kai-Uwe Bux. Thompson's group $F$ is maximally nonconvex. *Contemp. Math.*, 372:131–146, 2005.

[2] K.S. Brown. Finiteness properties of groups. *J. Pure App. Algebra*, 44:45–75, 1987.

[3] K.S. Brown and Geoghegan. R. An infinite-dimensional torsion-free $FP1$ group. *Invent. Math.*, 77:367–381, 1984.

[4] Jose Burillo, Sean Cleary, and Melanie Stein. Metrics and embeddings of generalizations of Thompson's group F. *Trans. Amer. Math. Soc.*, 353:1677–1689, 2001.

[5] J. W. Cannon. Almost convex groups. *Geom. Dedicata*, 22:197–210, 1987.

[6] James W. Cannon. The combinatorial structure of cocompact discrete hyperbolic groups. *Geom. Dedicata*, 16(2):123–148, 1984.

[7] J.W. Cannon, W.J. Floyd, and W.R. Parry. Introductory notes on Richard Thompson's groups. *Enseign. Math.*, 42:215–256, 1996.

[8] Sean Cleary, Murray Elder, and Jennifer Taback. Cone types and geodesic languages for lamplighter groups and Thompson's group $F$. *J. Algebra*, 303(2):476–500, 2006.

[9] Sean Cleary and Jennifer Taback. Thompson's group $F$ is not almost convex. *J. Algebra*, 270:133–149, 2003.

[10] Sean Cleary and Jennifer Taback. Combinatorial properties of Thompson's group $F$. *Trans. Amer. Math. Soc.*, 356:2825–2849, 2004.

[11] Sean Cleary and Jennifer Taback. Seesaw words in Thompson's group $F$. *Contemp. Math.*, 372:147–159, 2005.

[12] S. Blake Fordham. Minimal length elements of $F(p)$. Preprint.

[13] S. Blake Fordham. *Minimal Length Elements of Thompson's Group $F$*. PhD thesis, Brigham Young University, 1995.

[14] S. Blake Fordham. Minimal length elements of Thompson's group $F$. *Geom. Dedicata*, 99:179–220, 2003.

[15] G. Higman. Finitely presented infinite simple groups. *Notes on Pure Math.*, 8, 1974.

[16] Walter D. Neumann and Michael Shapiro. A short course in geometric group theory. In *Notes for the ANU Workshop*, January/February 1996. Topology Atlas Document no. iaai-13.

[17] Melanie Stein. Groups of piecewise linear homeomorphisms. *Trans. Amer. Math. Soc.*, 332(2):477–514, 1992.

[18] R. J. Thompson and R. McKenzie. An elementary construction of unsolvable word problems in group theory. In *Word problems, Conference at University of California, Irvine.* North Holland, 1969 (1973).