ON THE MAXIMUM CARDINALITY OF BRAID CLASSES

By Zachary Parker

A Thesis

Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Science

in Mathematics

Northern Arizona University

August 2017

Approved:

Dana C. Ernst, Ph.D., Chair

Michael Falk, Ph.D.

Nándor Sieben, Ph.D.

ABSTRACT

ON THE MAXIMUM CARDINALITY OF BRAID CLASSES

ZACHARY PARKER

A Coxeter group W is often thought of as a generalized reflection group generated by a set of elements of order two coupled with rules about when generators commute and the so-called braid relations. Every element w of some Coxeter group W can be written as an expression in these generators. When the number of generators used is minimal (including multiplicity), the expression is reduced and the number of generators is its length. Given some $w \in W$, we can form an equivalence relation on its set of reduced expressions: reduced expressions w and w' are braid equivalent if w' is obtainable from w via a sequence of braid moves. The corresponding equivalence classes are called braid classes. For each $w \in W$, we can form its braid graph, which has the set of braid classes for its vertex set and two braid classes are connected by an edge if a representative from one braid class is related to a representative of the other class via a single commutation. Moreover, the weight of each vertex is the cardinality of the corresponding braid class. Unlike the analogous commutation classes and commutation graphs, braid classes and braid graphs have received little attention in any context. In this thesis, we investigate a concise way of encoding braid classes to obtain a sharp upper bound on the cardinality of braid classes among all elements of a fixed length. This upper bound corresponds to the largest possible weight of a vertex in braid graphs for elements of a given length. This result previously appeared in Zollinger's thesis, but our proof is new.

Acknowledgements

Thank you, first, Dr. Dana C. Ernst, for your unparalleled patience and invaluable guidance throughout this effort. I will be forever grateful to you, my mentor, teacher, and friend. You believed in me. And so did you, Jeff Rushall, during my time here at NAU. Nothing I've accomplished could have been a reality without you or Dana, and I am proud and happy to share my accomplishments with you. I am fondly indebted to both of you. Dr. Michael Falk and Dr. Nándor Sieben, thank you for your time in sitting on my thesis committee with additional thanks to Dr. Sieben for Figure 1.3. Finally, warm thanks to all the wonderful folks at Dark Sky Brewing for providing a place for me to write, belong, and be (somewhat) productive.

Contents

List of Figures v

Chapter 1 Preliminaries **1**
1.1 Introduction 1
1.2 Coxeter Systems 2
1.3 Elementary Commutation and Braid Equivalences 8
1.4 Heaps 13

Chapter 2 Encoding Braid Classes **16**
2.1 σ-strings and Tracks 16
2.2 Maximal σ-string Decompositions and Codes 20

Chapter 3 Cardinality of Braid Classes **24**
3.1 Preparatory Lemmas 24
3.2 Main Result 28
3.3 Open Problems 29

Bibliography **31**

List of Figures

1.1 Examples of common Coxeter graphs . . . 3
1.2 Matsumoto graph for $w_0 \in W(A_3)$ . . . 6
1.3 Matsumoto graph for $w_0 \in W(A_4)$ . . . 7
1.4 Commutation graph for $w_0 \in W(A_3)$ . . . 10
1.5 Braid graph for $w_0 \in W(A_3)$ . . . 10
1.6 Commutation graph for $w_0 \in W(A_4)$ . . . 11
1.7 Braid graph for $w_0 \in W(A_4)$ . . . 12
1.8 Labeled Hasse diagram for the heap of a reduced expression for an element in $W(A_4)$ . . . 14
1.9 Two heaps for the same element in $W(A_7)$ . . . 15

2.1 A collection of heaps for six reduced expressions corresponding to a single braid class of a group element in $W(A_9)$ . . . 17
2.2 Tracks for the first three defined σ-strings . . . 19
2.3 The track for $\sigma_{l,k,n,+}$ with $l \geq 3$ . . . 22
2.4 Product of tracks corresponding to the maximal σ-string decomposition given in Example **??** . . . 23

Chapter 1

Preliminaries

1.1 Introduction

Among the variety of interesting topics in group theory lies the study of symmetry. Reflection groups are used to understand the reflection and rotational symmetries of particular objects. Coxeter groups can be thought of as generalized reflection groups with a fixed set of generating involutions and a certain set of guiding interactions among these involutions. The study of Coxeter groups looks to explain the intricacies of those interactions.

Every element of a Coxeter group can be written as an expression in the generators and when the number of generators in an expression is minimal (counting multiplicity), the expression is said to be reduced. It is likely that an element in a Coxeter group has many different reduced expressions representing it. According to Matsumoto's Theorem [4], any two reduced expressions for a group element are related via commutations and the so-called braid relations.

Following Stembridge [9], we define a relation $\sim$ on the set of reduced expressions for w. Let w and w' be two reduced expressions for w and define $\mathsf{w} \sim \mathsf{w}'$ if we can obtain w' from w by applying a single commutation. Now, define the equivalence relation $\approx$ by taking the reflexive transitive closure of $\sim$. Each equivalence class under $\approx$ is called a *commutation class*.

There are several open questions involving commutation classes. In particular, enumerating the number of commutation classes for an arbitrary element is a long-standing open problem. Even the case involving the longest element (in terms of Coxeter length) of the symmetric group remains open.

Given a Coxeter system (W, S) and a fixed element $w \in W$, one can form the corresponding *commutation graph* having the set of commutation classes as its vertices. There is an edge from one commutation class to another if there exists a reduced expression of one class that can be transformed into a reduced expression of the other class via a braid move. Commutation graphs have been studied in a variety of contexts, but answers to many natural questions remain elusive.

Analogous to commutation classes and commutation graphs, one can define braid classes and braid graphs. Fix an element $w \in W$ and consider the set of all reduced expressions for w. Two reduced expressions w and w' for w are in the same *braid class* if w' can be obtained from w via a sequence of braid moves. The corresponding *braid graph* for w has the braid classes as its vertices and two braid classes are joined by an edge if there exists a reduced expression of one class that can be transformed into a reduced expression of the other class via a commutation move.

Comparable questions in the context of braid classes rather than commutation classes remain relatively untouched. The focus of this thesis is clarifying some of the questions concerning braid classes for elements in Coxeter systems of type A (i.e., the symmetric group) as well as providing additional and potentially interesting open questions and conjectures.

This thesis is organized as follows. After introducing Coxeter systems, Matsumoto's Theorem, and the Matsumoto graph of a group element in Section 1.2, we define commutation and braid equivalences and their respective graphs in Section 1.3. After that, in Section 1.4, we establish a convenient visual representation for elements of Coxeter groups called heaps, concluding Chapter 1. Chapter 2 introduces σ-strings and tracks, both particularly useful for encoding information about braid equivalences, along with various, previously known results corresponding to braid classes from [11]. In Section 3.1, we provide several lemmas key to the proof of our main result (Theorem 3.2.1) contained in Section 3.2, which describes a sharp upper bound on the cardinality of a braid class among all elements of a fixed length. The upper bound corresponds to the largest possible weight of a vertex in braid graphs for elements of a given length. This result previously appeared in [11], but the proof presented here is new. We conclude with open problems in Section 3.3.

1.2 Coxeter Systems

A *Coxeter system* is a pair (W, S) consisting of a finite set S of generating involutions and a group W, called a *Coxeter group*, with presentation

$$W = \langle S \mid (st)^{m(s,t)} = e \rangle,$$

where e is the identity element, $m(s,t) = 1$ if and only if $s = t$, and $m(s,t) = m(t,s) \geq 2$ for $s \neq t$. If there is no relation between $s, t \in S$, then $m(s,t) = \infty$. It is worth noting that the elements of S are distinct as group elements and $m(s,t)$ is the order of st [5]. We refer to $m(s,t)$ as the *bond strength* between s and t.

Since s and t are elements of order 2, the relation $(st)^{m(s,t)} = e$ can be rewritten as

$$\underbrace{sts\cdots}_{m(s,t)} = \underbrace{tst\cdots}_{m(s,t)} \tag{1.2.1}$$

with $m(s,t) \geq 2$ factors. When $m(s,t) = 2$, $st = ts$ is called a *commutation relation* and when $m(s,t) \geq 3$, the relation in (1.2.1) is called a *braid relation*. The replacement

$$\underbrace{sts\cdots}_{m(s,t)} \longmapsto \underbrace{tst\cdots}_{m(s,t)}$$

will be referred to as a *commutation move* if $m(s,t) = 2$ and a *braid move* if $m(s,t) \geq 3$.

We can represent a Coxeter system (W,S) with a *Coxeter graph* Γ having vertex set S and edges $\{s,t\}$ for each $m(s,t) \geq 3$. Each edge $\{s,t\}$ is labeled with its corresponding bond strength. Since $m(s,t) = 3$ occurs frequently, we omit this label. Note that s and t are not connected by a single edge in the graph if and only if $m(s,t) = 2$. Given a Coxeter graph Γ, we can easily reconstruct the corresponding Coxeter system. If (W,S) is a Coxeter system with corresponding Coxeter graph Γ, we may denote the Coxeter group as $W(\Gamma)$ and the generating set as $S(\Gamma)$. A Coxeter system (W,S) is said to be *irreducible* if and only if the corresponding Coxeter graph Γ is connected.

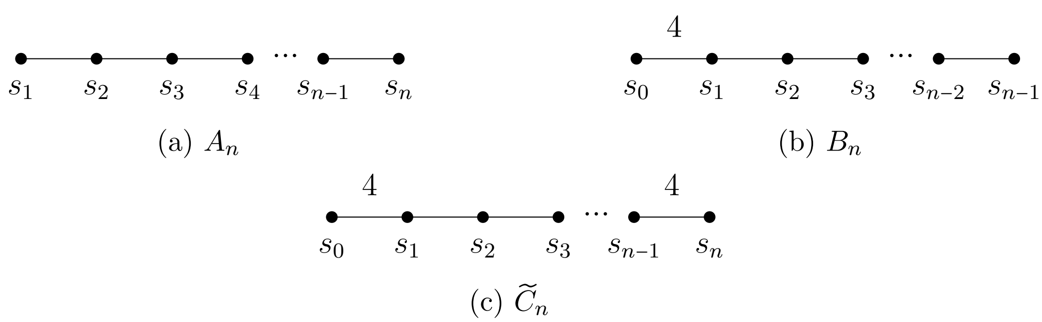


Figure 1.1: Examples of common Coxeter graphs

Example 1.2.1. The Coxeter graphs given in Figure 1.1 correspond to common Coxeter systems.

(a) The Coxeter system of type A_n is given by the graph in Figure 1.1(a). The corresponding Coxeter group $W(A_n)$ has generating set $S(A_n) = \{s_1, s_2, \ldots, s_n\}$ and defining relations

(1) $s_i^2 = e$ for all i;

(2) $s_i s_j = s_j s_i$ when $|i - j| > 1$;

(3) $s_i s_j s_i = s_j s_i s_j$ when $|i - j| = 1$.

The Coxeter group $W(A_n)$ is isomorphic to the symmetric group S_{n+1} under the correspondence $s_i \mapsto (i, i+1)$, where $(i, i+1)$ is the adjacent transposition that swaps i and $i+1$. This particular Coxeter group is the main focus of this thesis.

(b) The Coxeter system of type B_n is given by the graph in Figure 1.1(b). The Coxeter group $W(B_n)$ has generating set $S(B_n) = \{s_0, s_1, \ldots, s_{n-1}\}$ and defining relations

(1) $s_i^2 = e$ for all i;

(2) $s_i s_j = s_j s_i$ when $|i - j| > 1$;

(3) $s_i s_j s_i = s_j s_i s_j$ when $|i - j| = 1$ for $i, j \in \{1, 2, \ldots, n-1\}$;

(4) $s_0 s_1 s_0 s_1 = s_1 s_0 s_1 s_0$.

The Coxeter group $W(B_n)$ is isomorphic to S_n^B, the group of signed permutations on the set $\{1, 2, \ldots, n\}$.

(c) The Coxeter system of type $\widetilde{C}_n$ is given by the graph in Figure 1.1(c). The Coxeter group $W(\widetilde{C}_n)$ has generating set $S(\widetilde{C}_n) = \{s_0, s_1, \ldots, s_n\}$ and defining relations

(1) $s_i^2 = e$ for all i;

(2) $s_i s_j = s_j s_i$ when $|i - j| > 1$;

(3) $s_i s_j s_i = s_j s_i s_j$ when $|i - j| = 1$ for $i \in \{1, 2, \ldots, n-1\}$;

(4) $s_0 s_1 s_0 s_1 = s_1 s_0 s_1 s_0$;

(5) $s_n s_{n-1} s_n s_{n-1} = s_{n-1} s_n s_{n-1} s_n$.

Observe that $W(\widetilde{C}_n)$ has $n+1$ generators. It is worth noting that, unlike $W(A_n)$ and $W(B_n)$, $W(\widetilde{C}_n)$ is an infinite group [5].

Given a Coxeter system (W, S), a word $s_{x_1} s_{x_2} \cdots s_{x_m}$ in the free monoid S^* on the alphabet S is called an *expression* for $w \in W$ if it is equal to w when considered as a group element. If m is minimal among all expressions for w, the corresponding word is called a *reduced expression* for w and the *length* of w, denoted $\ell(w)$, is m. Each element $w \in W$ may have multiple reduced expressions that represent it. A specific, possibly reduced, expression for $w \in W$ is represented (using sans serif font) as $\mathsf{w} = s_{x_1} s_{x_2} \cdots s_{x_m}$. A product of group elements $w_1 w_2 \ldots w_r$ with $w_i \in W$ is called *reduced* if $\ell(w_1 w_2 \cdots w_r) = \sum_{i=1}^{r} \ell(w_i)$.

According to [5], every finite Coxeter group contains a unique element of maximal length, which we refer to as the *longest element* and denote by w_0. It is well known that the longest element in $W(A_n)$ is given in 1-line notation by

$$w_0 = [n+1, n, \dots, 2, 1]$$

and that $\ell(w_0) = \binom{n+1}{2}$. One possible reduced expression for w_0 is given by

$$s_1 s_2 s_1 s_3 s_2 s_1 \cdots s_n s_{n-1} \cdots s_3 s_2 s_1.$$

A formula for the number of reduced expressions for $w_0 \in W(A_n)$ is given in [8].

The following theorem, called Matsumoto's Theorem [4], illuminates how reduced expressions for a given group element are related.

Proposition 1.2.2. In a Coxeter system (W, S) any two reduced expressions for the same group element differ by a sequence of commutations and braid moves. □

In light of Matsumoto's Theorem, we can define a graph on the set of reduced expressions for a fixed group element $w \in W$. Formally, for $w \in W$, define the Matsumoto graph, $M(w)$, to be the graph having vertex set equal to the set of reduced expressions for w, where two reduced expressions w_1 and w_2 are connected by an edge if and only if w_1 and w_2 are related via a single commutation or braid move. In type A_n, colored edges are used to distinguish between commutation and braid moves. An edge is colored red if it corresponds a commutation move and colored green if it corresponds to a braid move. Matsumoto's Theorem implies that $M(w)$ is connected.

Example 1.2.3. The Matsumoto graph for $w_0 \in W(A_3)$ is given in Figure 1.2. The sixteen reduced expressions for w_0 are the vertices of $M(w_0)$. The edges of $M(w_0)$ show how pairs of reduced expressions are related via commutation or braid moves. For sake of brevity, i is written in place of s_i, a convention used throughout the remainder of this paper. The Matsumoto graph for $w_0 \in W(A_4)$ is given in Figure 1.3. Because of its size, the vertex labels have been omitted.

For $w \in W$ define the *support* of w, denoted $\text{supp}(w)$, to be the set of generators that appear in any reduced expression for w. Note that Matsumoto's Theorem guarantees that $\text{supp}(w)$ is well-defined. Given $w \in W$ and a fixed reduced expression w for w, any subsequence of w is called a *subexpression* of w. We will refer to a subexpression consisting of a consecutive subsequence of w as a *subword* of w.

Example 1.2.4. Let $\mathsf{w} = s_5 s_4 s_3 s_2 s_4 s_2$ be an expression for some $w \in W(A_5)$. Then we have

$$\begin{aligned}
s_5 s_4 s_3 s_2 s_4 s_2 &= s_5 s_4 s_3 s_4 s_2 s_2 && \text{(via the commutativity of } s_2 \text{ and } s_4) \\
&= s_5 s_4 s_3 s_4 && \text{(because the order of } s_2 \text{ is 2)} \\
&= s_5 s_3 s_4 s_3 && \text{(via the braid relation } s_4 s_3 s_4 = s_3 s_4 s_3).
\end{aligned}$$

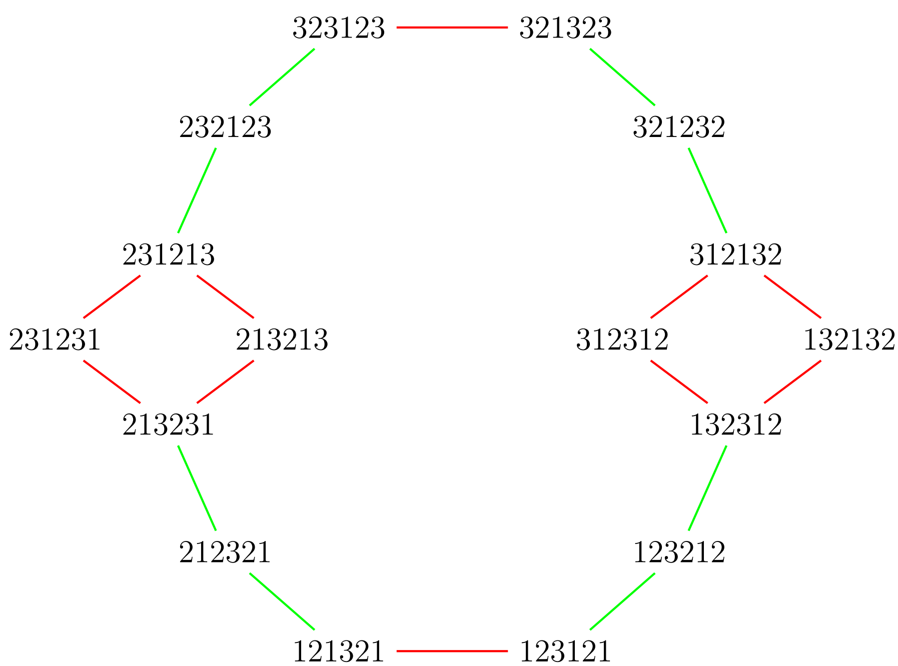


Figure 1.2: Matsumoto graph for $w_0 \in W(A_3)$

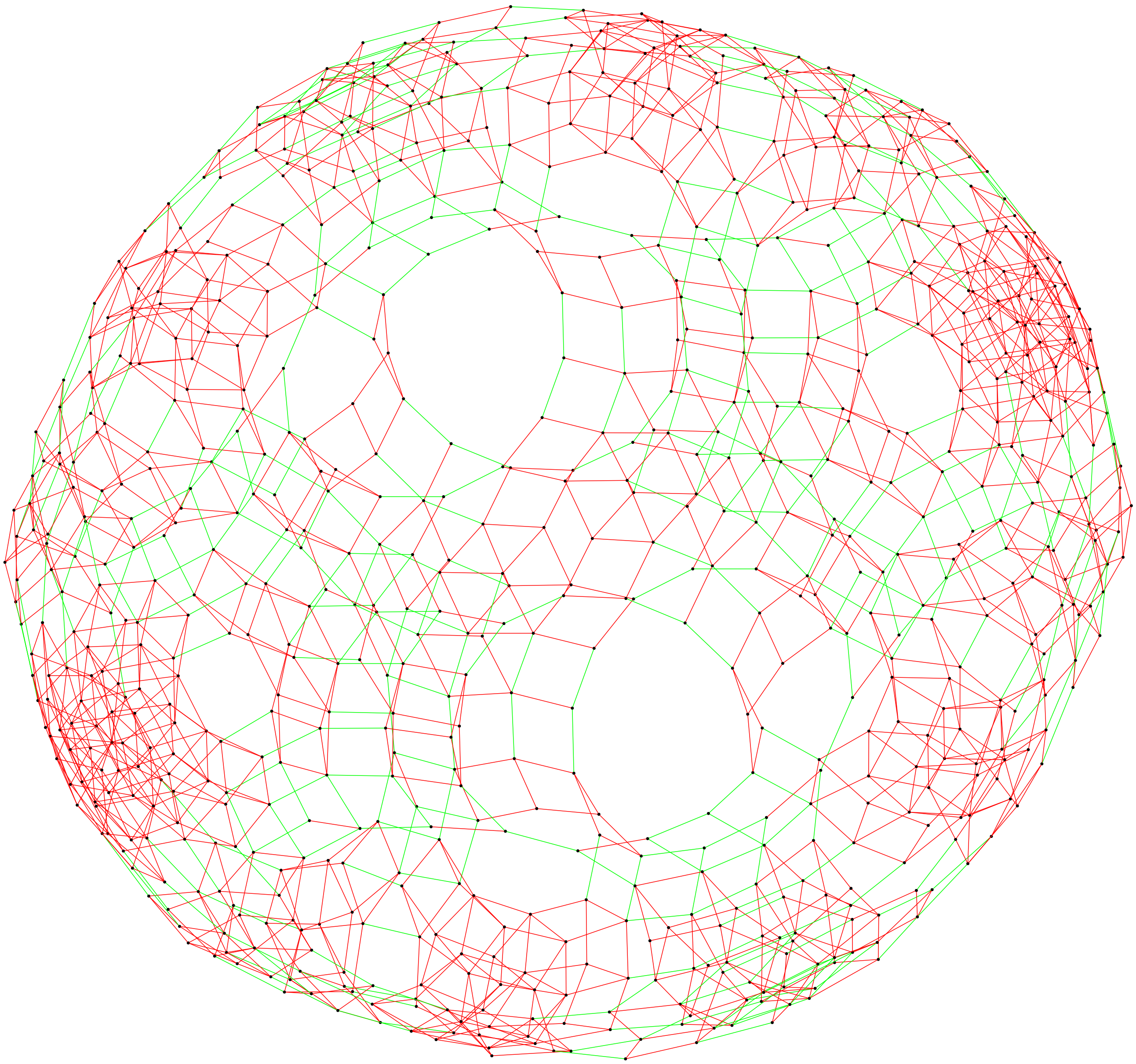


Figure 1.3: Matsumoto graph for $w_0 \in W(A_4)$

This shows that the original expression w is not reduced. However, one can check that $s_5s_3s_4s_3$ and $s_5s_4s_3s_4$ are reduced, so $\ell(w) = 4$ and $\operatorname{supp}(w) = \{s_3, s_4, s_5\}$.

1.3 Elementary Commutation and Braid Equivalences

Let (W, S) be a Coxeter system of type Γ and let $w \in W(\Gamma)$. As in [9], define the relation $\sim_c$ on the set of reduced expressions for w as follows: if w_1 and w_2 are two reduced expressions for w, then $\mathsf{w}_1 \sim_c \mathsf{w}_2$ if and only if we can obtain w_2 from w_1 using a single commutation move. The reflexive transitive closure of $\sim_c$ is defined to be the equivalence relation $\approx_c$. Equivalence classes under $\approx_c$ are called *commutation classes*, denoted $[\mathsf{w}]_c$, and two elements in the same commutation class are said to be *commutation equivalent*. The number of commutation classes for $w \in W$ is denoted $\mathfrak{c}(w)$. When the set of reduced expressions for w has only a single commutation class, then w is said to be *fully commutative* (FC).

The set of FC elements of $W(\Gamma)$ is denoted by $\mathrm{FC}(\Gamma)$. It follows from the definition that, given some $w \in \mathrm{FC}(\Gamma)$ and some starting reduced expression for w, all other reduced expressions can be obtained via commutations. The following result, due to Stembridge [9], states that when w is FC, performing commutations is the only possible method for obtaining other reduced expressions for w.

Proposition 1.3.1. Let (W, S) be a Coxeter system of type Γ. Then $w \in W(\Gamma)$ is FC if and only if no reduced expression for w contains $\underbrace{sts\cdots}_{m(s,t)}$ as a subword for all $m(s,t) \geq 3$. $\square$

Said differently, w is FC if and only if no reduced expression allows a braid move. For example, in a type A_n Coxeter system, an element is FC when no reduced expression contains $s_ks_{k+1}s_k$ or $s_{k+1}s_ks_{k+1}$ as a subword.

Similarly, define $\mathsf{w}_1 \sim_b \mathsf{w}_2$ if and only if w_2 can be obtained from w_1 via a single braid move. The equivalence relation $\approx_b$ is defined by taking the reflexive transitive closure of $\sim_b$. The equivalence classes under $\approx_b$ are called *braid classes*, denoted $[\mathsf{w}]_b$, and two reduced expressions are *braid equivalent* if they are in the same braid class. The number of braid classes for $w \in W$ is denoted $\mathfrak{b}(w)$. If w is a reduced expression for $w \in W$, define

$$\star(\mathsf{w}) := \operatorname{card}([\mathsf{w}]_b).$$

For a group element $w \in W$, we define

$$\circledast(w) := \max\left(\{\star(\mathsf{w}) \mid \mathsf{w} \text{ is a reduced expression for } w \in W\}\right).$$

Example 1.3.2. Consider the sixteen reduced expressions for $w_0 \in W(A_3)$ depicted in Figure 1.2. Applying all possible commutation moves, there are eight commutation classes:

$$[232123]_c = \{232123\}$$
$$[231213]_c = \{231213, 213213, 213231, 231231\}$$
$$[321323]_c = \{321323, 323123\}$$
$$[212321]_c = \{212321\}$$
$$[321232]_c = \{321232\}$$
$$[123123]_c = \{123121, 121321\}$$
$$[132312]_c = \{132312, 132132, 312132, 312312\}$$
$$[123212]_c = \{123212\}$$

So $\mathfrak{c}(w_0) = 8$. Similarly, applying all possible braid moves gives eight braid classes:

$$[123121]_b = \{123121, 123212, 132312\}$$
$$[312312]_b = \{312312\}$$
$$[312132]_b = \{312132, 321232, 321323\}$$
$$[132132]_b = \{132132\}$$
$$[121321]_b = \{121321, 212321, 213231\}$$
$$[213213]_b = \{213213\}$$
$$[231213]_b = \{231213, 232123, 323123\}$$
$$[231231]_b = \{231231\}$$

Thus, $\mathfrak{b}(w_0) = 8$ and $\circledast(w_0) = 3$. Also, we see that $\star(123121) = 3$ and $\star(312321) = 1$, for example. In general, it is not the case that the number of commutation classes for a fixed $w \in W$ is equal to the number of braid classes.

For $w \in W$, define the *commutation graph* (respectively, *braid graph*), denoted $C(w)$ (respectively, $B(w)$), to be the graph with vertex set equal to the set of commutation classes (respectively, braid classes) of w where two vertices are connected by an edge if some representative of one commutation class is related to a representative of another commutation class via a braid move (respectively, some representative of a braid class is related to a representative of another braid class via a commutation move). Note that we can obtain $C(w)$ (respectively, $B(w)$), from $M(w)$ by contracting edges corresponding to braid moves (respectively, commutation moves). The weight of a vertex in the braid graph is defined to be the cardinality of the corresponding braid class. Note that $\mathfrak{b}(w)$ is the number of vertices in $B(w)$ while $\circledast(w)$ is the maximum weight of a vertex.

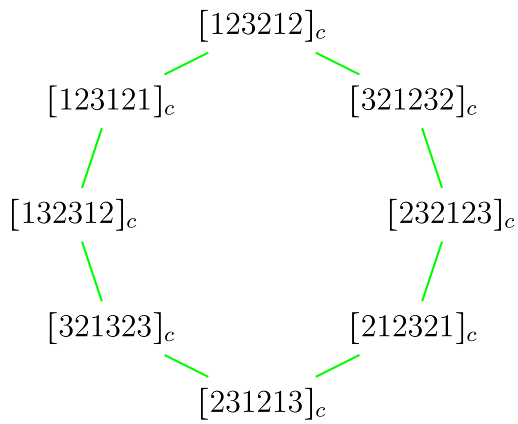


Figure 1.4: Commutation graph for $w_0 \in W(A_3)$

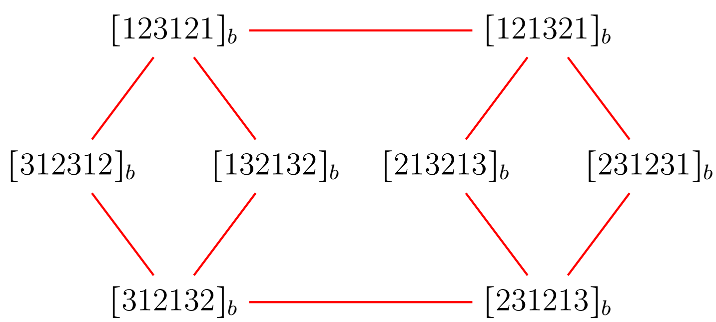


Figure 1.5: Braid graph for $w_0 \in W(A_3)$

Example 1.3.3. The commutation graphs for the longest element in $W(A_3)$ and $W(A_4)$ are given in Figure 1.4 and Figure 1.6, respectively, and the braid graphs are given in Figure 1.5 and Figure 1.7, respectively.

Determining the number of commutation classes for the longest element in $W(A_n)$ remains an open problem [**?**]. That is, the number of vertices in $C(w_0)$ is unknown for an arbitrary $W(A_n)$. To our knowledge, this problem was first introduced in 1992 by Knuth in Section 9 of [7] using different terminology. A more general version of the problem appears in Section 5.2 of [6]. In the paragraph following the proof of Proposition 4.4 of [10], Tenner explicitly states the open problem in terms of commutation classes. The best known bound appears in [3]: for sufficiently large n corresponding to $w_0 \in W(A_{n-1})$, $\mathfrak{c}(w_0) \leq 2^{0.6571n^2}$. Even less is known about he number of commutation classes of the longest elements in other

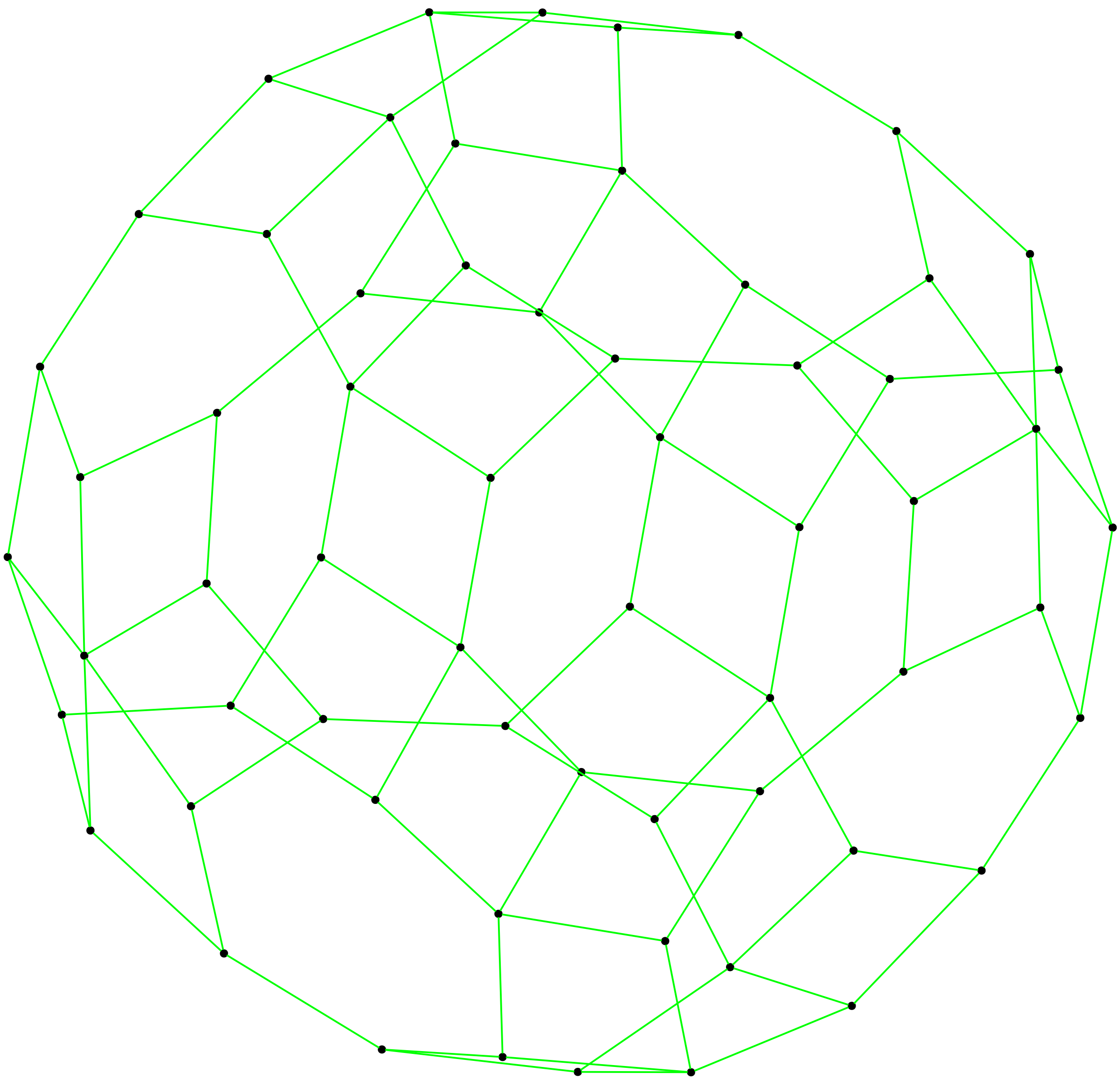


Figure 1.6: Commutation graph for $w_0 \in W(A_4)$

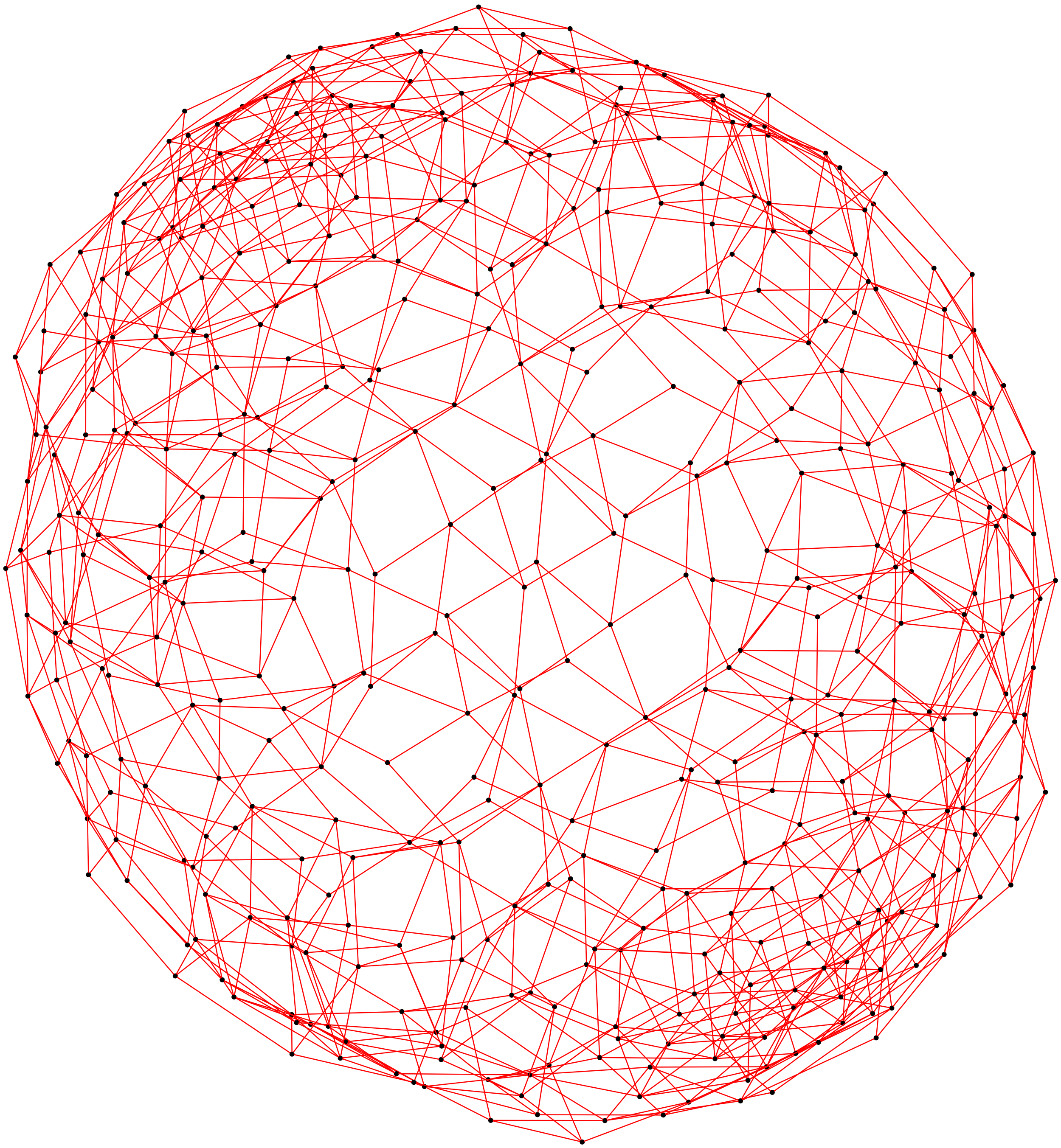


Figure 1.7: Braid graph for $w_0 \in W(A_4)$

Coxeter systems.

According to [?] the set of commutation classes for $w_0 \in W(A_n)$ is in bijection with various surprising (and interesting) sets of objects, including:

(1) heaps (defined in the following section) for w_0 in A_n;

(2) primitive sorting networks on n elements;

(3) rhombic tilings of a regular n-gon;

(4) uniform oriented matroids of rank 3 on n elements;

(5) arrangements of n pseudolines.

Unlike commutation classes and commutation graphs, braid classes and braid graphs have received very little attention in any context. It is unknown when $B(w)$ is a tree or path and what kinds of degrees can appear among the vertices. Moreover, very little is known about the upper and lower bounds on the value of $\mathfrak{b}(w)$ (i.e., the number of vertices in $B(W)$) for elements with fixed length. We address questions concerning
$ostar(w)$ (i.e., the maximum weight of a vertex in $B(w)$) in Chapter 3.

1.4 Heaps

Every reduced expression can be associated with a labeled poset called a heap. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators. We follow the development of heaps for straight-line Coxeter groups found in [1], [2], and [9].

Given a Coxeter system (W, S) of type Γ, let $\mathsf{w} = s_{x_1} s_{x_2} \cdots s_{x_r}$ be a fixed reduced expression for $w \in W(\Gamma)$. As in [9], define a partial ordering on the indices $\{1, 2, \ldots, r\}$ of w to be the transitive closure of the relation $\lessdot$ defined by $j \lessdot i$ if $i < j$ and $m(s_{x_i}, s_{x_j}) > 2$. In particular, since w is reduced, $j \lessdot i$ whenever $s_{x_i} = s_{x_j}$ and $i < j$. This partial order is referred to as the *heap* of w, where i is labeled by s_{x_i}. For simplicity, we omit the labels of the underlying poset yet retain the labels of the corresponding generators.

It follows from [9] that heaps are well-defined up to commutation class. That is, if w_1 and w_2 are two reduced expressions for $w \in W$ that are commutation equivalent, the heaps for w_1 and w_2 qre equal. In particular, when $w \in \mathrm{FC}(\Gamma)$, then w has a unique heap. Alternatively, if w_1 and w_2 are in different commutation classes, the heap of w_1 is distinct from the heap of w_2.

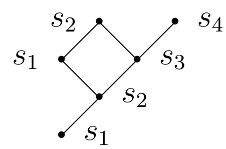


Figure 1.8: Labeled Hasse diagram for the heap of a reduced expression for an element in $W(A_4)$

Example 1.4.1. Consider the reduced expression $\mathsf{w} = s_2 s_1 s_4 s_3 s_2 s_1$ for $w \in W(A_4)$. Then w is indexed by $\{1,2,3,4,5,6\}$. As an example, $6 \prec 5$ since $5 < 6$ and s_2 and s_1 do not commute. The labeled Hasse diagram for the corresponding heap poset is seen in Figure 1.8.

Let w be a reduced expression for an element $w \in W(A_n)$. As in [1] and [2] we can represent a heap for w as a set of lattice points embedded in $\{1,2,\ldots,n\} \times \mathbb{N}$. To do so, we assign coordinates (not necessarily unique) $(x,y) \in \{1,2,\ldots,n\} \times \mathbb{N}$ to each entry of the labeled Hasse diagram for the heap for w in such a way that:

(1) An entry with coordinates (x,y) is labeled s_i (or i) in the heap if and only if $x = i$;

(2) If an entry with coordinates (x,y) is greater than an entry with coordinates (x',y') in the heap, then $y > y'$.

Although the above is specific to $W(A_n)$, the same construction works for any straight-line Coxeter graph with appropriate adjustments made to the label set and assignment of coordinates. For example, for type B_n, the label set is $\{0,1,\ldots,n-1\}$ and for type $\widetilde{C}_n$, the label set is $\{0,1,\ldots,n\}$.

For any straight-line Coxeter graph, it follows that (x,y) covers (x',y') in the heap if and only if $x = x' \pm 1$, $y' < y$, and there are no entries (x'',y'') such that $x'' \in \{x,x'\}$ and $y' < y'' < y$. This implies we can entirely reconstruct the edges of the Hasse diagram and corresponding heap poset from a lattice point configuration. This lattice point configuration allows us to visualize words and the relations therein. Further, the visualization can potentially give intuition into a seemingly complex argument.

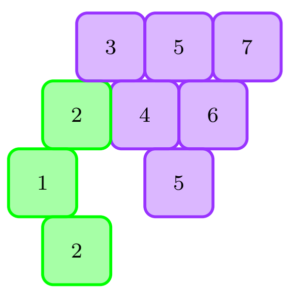
We denote the lattice point configuration of the heap poset in $\{1,2,\ldots,n\} \times \mathbb{N}$ described in the preceding paragraphs via $H(\mathsf{w})$ where w is a reduced expression for $w \in W(A_n)$. If w is FC, then the choice of the reduced expression for w is irrelevant and we refer to $H(w)$ as the heap of w.

Let $\mathsf{w} = s_{x_1} s_{x_2} \cdots s_{x_r}$ be a reduced expression for $w \in W(A_n)$. When s_{x_i} and s_{x_j} are adjacent generators in the Coxeter graph with $i < j$, the point labeled by s_{x_i} must be placed *above* the level of the point labeled by s_{x_j}. Because non-adjacent generators in a Coxeter graph commute, points whose x-coordinates differ by more than one can "slide" past each other in the configuration, possibly existing on the same level (i.e., the points can potentially have equal y-coordinates). To emphasize the covering relations of the lattice point configuration, each entry in the heap will be enclosed by a square with rounded corners, called a *block*, such that the blocks overlap halfway if one entry covers the other. Additionally, each block for s_i will be labelled by i.

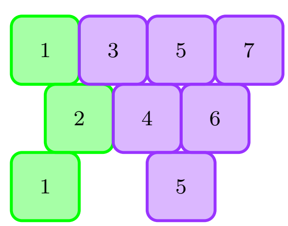
Via this convention, there are potentially many ways to illustrate a heap of an arbitrary reduced expression by differing the vertical placement of blocks. For example, we can place blocks in vertical positions as high as possible, as low as possible, or in some combination. We often choose what we view as the best representation of the heap of each example (nearly always, this defaults to placing blocks as low as possible). When illustrating the heaps of arbitrary reduced expressions, the relative position of the entries are discussed but never absolute coordinates.

Notice blocks appearing on the bottom (respectively, top) of a heap correspond to generators that can potentially appear on the right (respectively, left) of the reduced expressions corresponding to that heap.

Example 1.4.2. Let $\mathsf{w}_1 = s_7s_5s_3s_6s_4s_2s_5s_1s_2$ be a reduced expression for $w \in W(A_7)$. Applying the commutation move $s_2s_5 \mapsto s_5s_2$, we can obtain another reduced expression for w, namely $\mathsf{w}_2 = s_7s_5s_3s_6s_4s_5s_2s_1s_2$, which is in the same commutation class as w_1, and hence has the same heap. However, applying the braid move $s_2s_1s_2 \mapsto s_1s_2s_1$, we obtain another reduced expression $\mathsf{w}_3 = s_7s_5s_3s_6s_4s_5s_1s_2s_1$. Since w_3 differed from w_2 by a braid move, w_3 is in a different commutation class than w_1 and w_2, and hence the heaps are different. Representations of $H(\mathsf{w}_1), H(\mathsf{w}_2)$, and $H(\mathsf{w}_3)$ are seen in Figure 1.9, where the subword corresponding to the braid move is colored in green. The lone 3-block moves from the "fourth level" in Figure 1.9(a) to the "third" in Figure 1.9(b) via the braid move.



(a) $H(\mathsf{w}_1)$ and $H(\mathsf{w}_2)$.



(b) $H(\mathsf{w}_3)$

Figure 1.9: Two heaps for the same element in $W(A_7)$

Chapter 2

Encoding Braid Classes

Before the main topic of this thesis is discussed, we must first introduce notation (mimicking Zollinger in [11]) to write a particular family of reduced expressions for a given $w \in W(A_n)$ that accurately and concisely captures information about the braid classes of w.

2.1 σ-strings and Tracks

We begin with an example to motivate our notation.

Example 2.1.1. Consider the reduced expression 5465**6**768798 for an element in $W(A_9)$ (where i corresponds to s_i). There are two opportunities to apply a braid move, namely using the subwords 656 and 676, where the **boldfaced** generator contributes in both cases. The left braid move is depicted in blue while the right braid move is depicted in magenta:

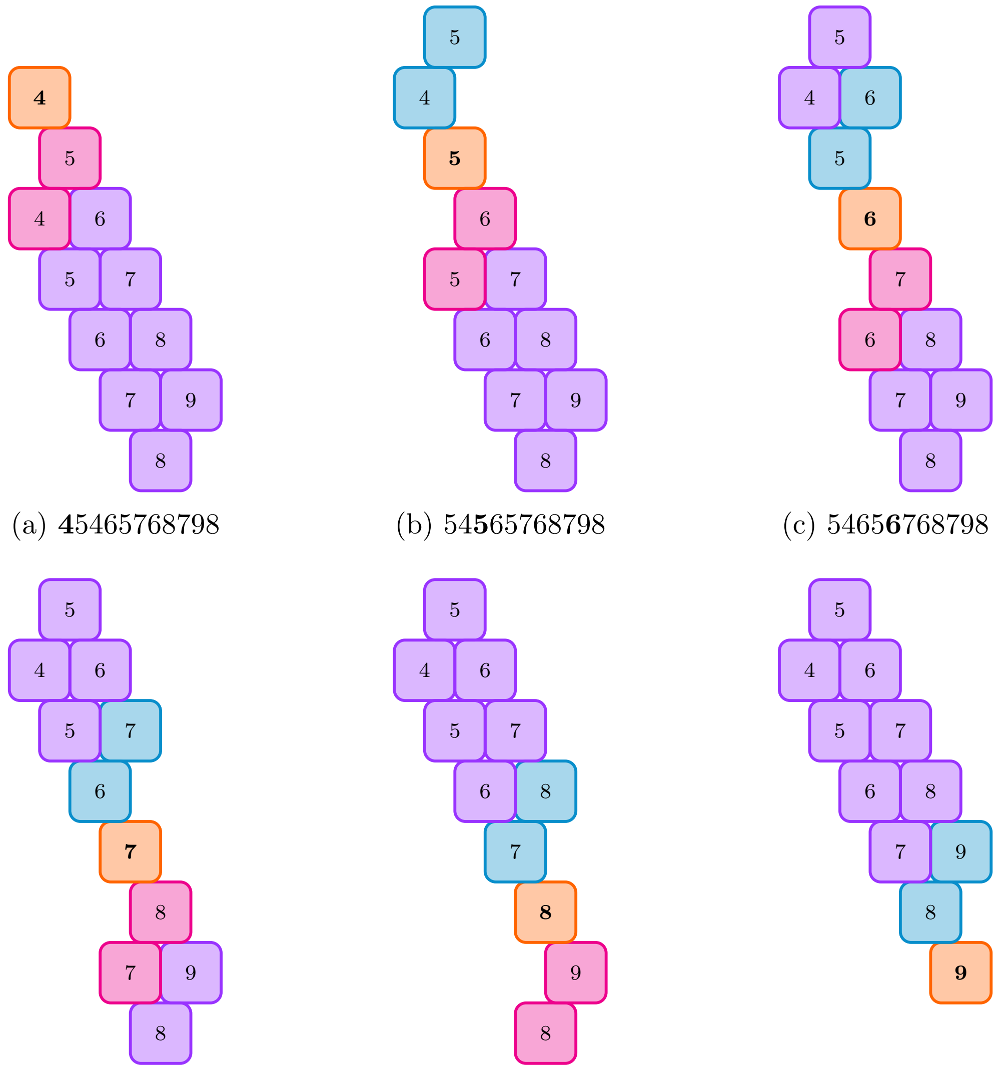
$$546\mathbf{5}6\mathbf{6}768798 \longleftrightarrow 54\mathbf{5}65768798$$

54656768798 ⟷ 54657678798

Now, each braid move creates opportunities for additional braid moves. The entire braid class corresponding to the original reduced expression is given below with a specific generator of interest **boldfaced** in each expression:

45465768798	54**5**65768798	5465**6**768798
546576**7**8798	54657687**8**98	5465768798**9**

Note that we have exactly two opportunities to apply a braid move if and only if the boldfaced letter is not the leftmost or rightmost letter in the word, and aside from these cases, the boldfaced letter is involved in both braid moves. The heap for each word in the braid class is given in Figure 2.1. The corresponding **boldfaced** generator is given in orange, the subword corresponding to the left braid move in blue, the subword corresponding to the right braid move in magenta, and the remainder of the heap blocks in purple.



(a) **4**5465768798 (b) 54**5**65768798 (c) 5465**6**768798

(d) 546576**7**8798 (e) 5465768**7**898 (f) 5465768798**9**

Figure 2.1: A collection of heaps for six reduced expressions corresponding to a single braid class of a group element in $W(A_9)$

We note some observations about Figure 2.1. In Figures 2.1(b) through 2.1(e), the subword corresponding to the left braid move, the orange generator, and the subword corresponding to the right braid move make an “S” shape, where Figures 2.1(a) and 2.1(f) have this shape truncated. Further, as the heaps progress from Figure 2.1(a) to Figure 2.1(f), the label of the orange block increases by one along with the number of pairs in front (i.e., top) of that element while the number of pairs behind (i.e., below) it decreases by one. Finally, all purple blocks come in pairs.

The previous example motivates the next definition, appearing in [11]. Let (l, k, n, ϵ) be a quadruple satisfying:

(1) l is a positive integer;

(2) k is a nonnegative integer less than or equal to $l - 1$;

(3) n is a positive integer (not necessarily distinct from l or k) and;

(4) ϵ is one of $\{+, -, 0\}$;

where $\epsilon = 0$ only when $l \leq 2$. From this quadruple, define $\sigma_{l,k,n,\epsilon}$ with $\epsilon \in \{+, 0\}$ to be the word, called a *σ-string*, satisfying:

(1) when $l = 1$, then $\sigma_{1,0,n,0} = \boldsymbol{n}$ (see Figure 2.2(a));

(2) when $l = 2$, then $\boldsymbol{n}, n+1, n$ and $\sigma_{2,1,n,0} = n+1, n, \boldsymbol{n+1}$ (see Figure 2.2(b) and Figure 2.2(c), respectively);

(3) when $l \geq 3$, then $\sigma_{l,k,n,+}$ is the first $2l - 1$ letters from the following list:

$$\begin{gathered} n+1, n, n+2, n+1, n+3, \ldots, n+k, n+k-1, \\ \boldsymbol{n+k}, n+k+1, n+k, \\ n+k+2, n+k+1, n+k+3, n+k+2, \ldots \end{gathered}$$

where the element in **bold** is called the *core*. We define $\sigma_{l,k,n,-}$ to be the reverse (i.e., inverse) of $\sigma_{l,l-1-k,n,+}$.

It turns out that every σ-string is a reduced expression. When convenient, we will consider a σ-string to be both a reduced expression and a group element. This will allow us to avoid cumbersome notation. Observe that $\sigma_{l,k,n,+}$ has length $2l - 1$.

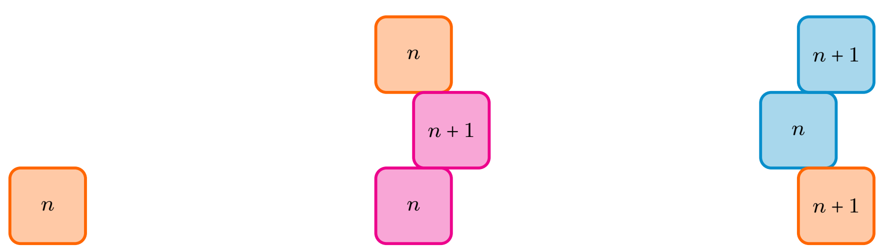
Example 2.1.2. The six reduced expressions from Example 2.1.1 are identified as σ-strings below:

$$\begin{array}{lll} \sigma_{6,0,4,+} = \mathbf{4}5465768798 & \sigma_{6,1,4,+} = 54\mathbf{5}65768798 & \sigma_{6,2,4,+} = 54656\mathbf{6}768798 \\ \sigma_{6,3,4,+} = 546576\mathbf{7}8798 & \sigma_{6,4,4,+} = 54657687\mathbf{8}98 & \sigma_{6,5,4,+} = 5465768798\mathbf{9} \end{array}$$

There are a few points to note following the definition of σ-strings. First, k is called the *pairing number* because k signifies the number of pairs preceding the core. Second, ϵ is referred to as the *pairing order*. If $\epsilon = +$, then pairs come in the form $i+1, i$ and if $\epsilon = -$, pairs come as $i, i+1$. Certainly there are no pairs with $l = 1, \epsilon = 0$, but the cases $l = 2, k = 0, \epsilon = 0$ and $l = 2, k = 1, \epsilon = 0$ pair like $\epsilon = +$ which is clear by Figures 2.2(b) and 2.2(c).

Because every σ-string can be thought of as another σ-string's reverse, and because an element and its inverse have the same number of braid classes with the same cardinalities, we need to only consider σ-strings where $\epsilon = 0$ or $\epsilon = +$ and do so for the remainder of this paper.

Recall that each heap corresponds to a single commutation class. A heap is largely used as a tool for studying commutation classes within a Coxeter system but a certain ambiguity occurs when observing braid classes since two braid equivalent reduced expressions have different heaps. This lead us to define a track. A *track* is a lattice point representation of the heap of a σ-string where we canonically place each block in the "heap" as high as possible and we unambiguously read the rows in the "heap" left to right and top to bottom. We warn here that a track technically is not a heap since we forbid commutations. Tracks are simply visual aids for understanding σ-strings. We typically color the core orange and the left and right braids are the *left* and *right switches*, respectively. The left (respectively, right) switch corresponds to the blocks involved in the left (respectively, right) braid move. We have colored the non-core blocks blue and magenta, respectively. The tracks for $\sigma_{1,0,n,0}$, $\sigma_{2,0,n,0}$, and $\sigma_{2,1,n,0}$ are given in Figure 2.2 and the track for $\sigma_{l,k,n,+}$ for $l \geq 3$ is depicted in Figure 2.3 with corresponding colors.



(a) $\sigma_{1,0,n,0}$ (b) $\sigma_{2,0,n,0}$ (c) $\sigma_{2,1,n,0}$

Figure 2.2: Tracks for the first three defined σ-strings

We examine the role k plays between σ-strings that are related via a braid move (or a sequence thereof) through an example.

Example 2.1.3. Consider the first two σ-strings from Example 2.1.2, namely $\sigma_{6,0,4,+} =$ **4**5465768798 and $\sigma_{6,1,4,+} =$ 54**5**65768798, whose tracks are depicted in Figures 2.1(a) and 2.1(b), respectively. Notice that $\sigma_{6,0,4,+}$ and $\sigma_{6,1,4,+}$ differ by a single braid move, namely

$$45465768798 \longleftrightarrow 54565768798.$$

After the braid move is applied, the core moves from the first position down to the third position in the reduced expression and increases in value by one. We observe two effects: first, the change in value of k from 0 to 1 and second in the tracks. We see the same sort of relationship is preserved: the core moves two more spots down and its index increases in value by one. In fact, from track to track in Figure 2.1, the same observations hold (mirrored by the changes to k). It is readily seen that all possible braid moves have been accounted for. The upshot is that $\star(\sigma_{6,0,4,+}) = 6$.

The phenomenon illustrated in the previous example holds in general. The following lemma is proved in [11].

Lemma 2.1.4. If $\sigma_{l,n,k,\epsilon}$ is a σ-string, then $[\sigma_{l,n,k,\epsilon}]_b = \{\sigma_{l,n,k,\epsilon}\}_{k\in[0,l-1]}$ and $\star(\sigma_{l,n,k,\epsilon}) = l$.

Said differently, Lemma 2.1.4 gives the braid class of $\sigma_{l,n,k,\epsilon}$ by ranging k over $[0, l-1]$. We note that when $k = 0$, the string $\sigma_{l,n,0,\epsilon}$ begins with n and for all other k, the σ-string begins with $n+1$. Due to this construction and the braid equivalences among σ-strings with the same l, n, and ϵ values, our canonical choice for the representative of the braid class will be $\sigma_{l,0,n,+}$ (and $\sigma_{1,0,n,0}$ and $\sigma_{2,0,n,0}$ when appropriate). Viewing the track with $k = 0$ provides a visualization for the entire braid class of a σ-string.

2.2 Maximal σ-string Decompositions and Codes

In a fixed reduced word, a subword is called a *maximal σ-string* if it is a σ-string that is contained, with respect to position, in no other σ-string in that word. We note that a σ-string may be maximal in one reduced expression but not in another. Given a reduced expression, a factorization into maximal σ-strings is called a *maximal string decomposition*. This leads us to the following lemma from [11].

Lemma 2.2.1. Every reduced word has a unique maximal string decomposition.

Therefore, we can accurately capture all the information for a particular reduced expression by giving it as a product of maximal σ-strings. This leads us to define the *code* for a reduced expression w of a group element in $W(A_n)$ to be the r-tuple

$$\big((l_1, n_1, \epsilon_1), \ldots, (l_r, n_r, \epsilon_r)\big)$$

where $\mathsf{w} = \sigma_{l_1,k_1,n_1,\epsilon_1} \cdots \sigma_{l_r,k_r,n_r,\epsilon_r}$ and the product on the right is a maximal σ-string decomposition. If $\mathsf{w} = \sigma_1 \cdots \sigma_k$ is a maximal σ-string decomposition, then the *string structure* of w is the multiset consisting of $\{\ell(\sigma_1), \ldots, \ell(\sigma_k)\}$. Note that, if

$$\mathrm{code}(\mathsf{w}) = \big((l_1, n_1, \epsilon_1), \ldots, (l_r, n_r, \epsilon_r)\big),$$

then $\mathrm{Struct}(\mathsf{w}) = \{2l_1 - 1, \ldots, 2l_r - 1\}$.

Example 2.2.2. Consider the reduced expression $\mathsf{w} = 3121434546576$ for an element in $W(A_7)$. Then its maximal σ-string decomposition is

$$\mathbf{3}|\mathbf{1}21|43\mathbf{4}546576 = \sigma_{1,0,3,0} \cdot \sigma_{2,0,1,0} \cdot \sigma_{5,1,3,+}$$

where we use vertical bars to designate the factorization into maximal σ-strings. Its code is

$$\big((1,3,0),(2,1,0),(5,3,+)\big)$$

and Figure 2.4 depicts this decomposition as a product of tracks, where the horizontal lines correspond to the vertical bars from the reduced expression. Also, $\mathrm{Struct}(\mathsf{w}) = \{1,3,9\}$.

It is easy to verify that every braid equivalent reduced expression has the same code. This is true in general, as the next theorem from [11] states.

Theorem 2.2.3. If w and w' are two reduced words for $w \in W(A_n)$, then $[\mathsf{w}]_b = [\mathsf{w}']_b$ if and only if $\mathrm{code}(\mathsf{w}) = \mathrm{code}(\mathsf{w}')$.

Sketch of Proof. We start with the converse implication. Consider the maximal σ-string decompositions $\mathsf{w} = \sigma_1 \cdot \sigma_2 \cdots \sigma_r$ and $\mathsf{w}' = \sigma_1' \cdot \sigma_2' \cdots \sigma_s'$ with $\sigma_i = \sigma_{l_{i_1},k_{i_1},n_{i_1},\epsilon_{i_1}}$ and $\sigma_i' = \sigma_{l_{i_2},k_{i_2},n_{i_2},\epsilon_{i_2}}$, respectively. Since $\mathrm{code}(\mathsf{w}) = \mathrm{code}(\mathsf{w}')$, $r = s$, $l_{i_1} = l_{i_2}$, $n_{i_1} = n_{i_2}$, and $\epsilon_{i_1} = \epsilon_{i_2}$ for all i. Then $[\mathsf{w}]_b = [\mathsf{w}']_b$ by Lemma 2.1.4.

For the forward direction, assume $[\mathsf{w}]_b = [\mathsf{w}']_b$ and proceed by contradiction. Without loss of generality, we can assume that w and w' differ by a single braid equivalence, say in the p^{th} component, σ_p, in w and the q^{th} component, σ_q, of w'. The work comes in showing that either the p^{th} string of w or the q^{th} string of w' is not maximal. Once that is shown, we have that the code of each component is equal and so $\mathrm{code}(\mathsf{w}) = \mathrm{code}(\mathsf{w}')$. □

The next corollary follows immediately from Lemma 2.1.4 and Theorem 2.2.3 (and appears as Corollary 6 in [11]).

Corollary 2.2.4. If w is a reduced expression for $w \in W(A_n)$ such that

$$\mathrm{code}(\mathsf{w}) = \big((l_1,n_1,\epsilon_1),\ldots,(l_r,n_r,\epsilon_r)\big),$$

then

$$[\mathsf{w}]_b = \{\sigma_{l_1,k_2,n_1,\epsilon_1}\cdots\sigma_{l_r,k_r,n_r,\epsilon_r} \mid k_i \text{ over } 0 \text{ to } l_i - 1 \text{ for each } i\}$$

and

$$\star(\mathsf{w}) = \prod_{i=1}^{r} l_i.$$

The final result of this chapter is an immediate consequence of Theorem 2.2.3.

Corollary 2.2.5. If two reduced expressions w and w' have the same string structure, then $\star(\mathsf{w}) = \star(\mathsf{w}')$.

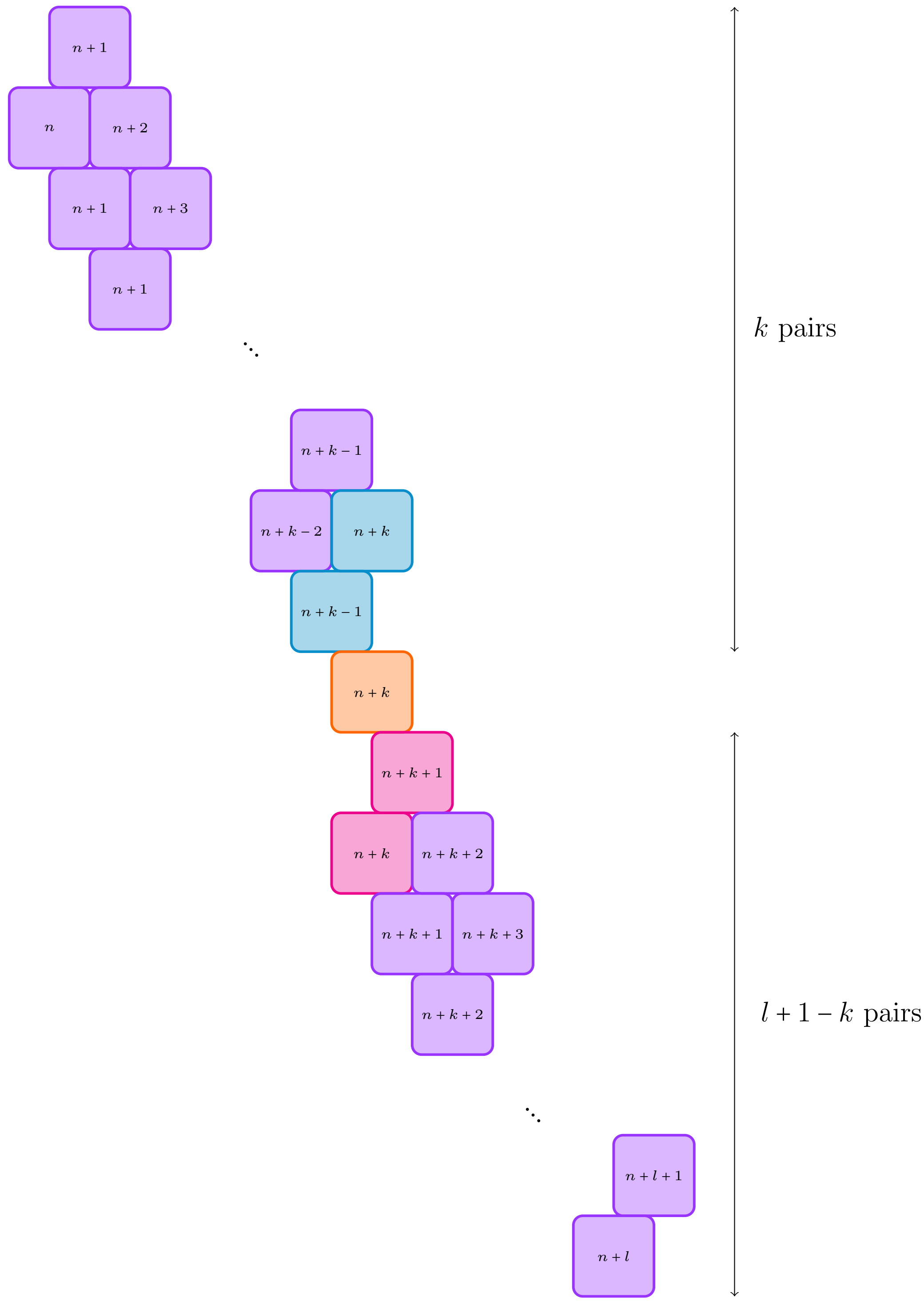


Figure 2.3: The track for $\sigma_{l,k,n,+}$ with $l \geq 3$

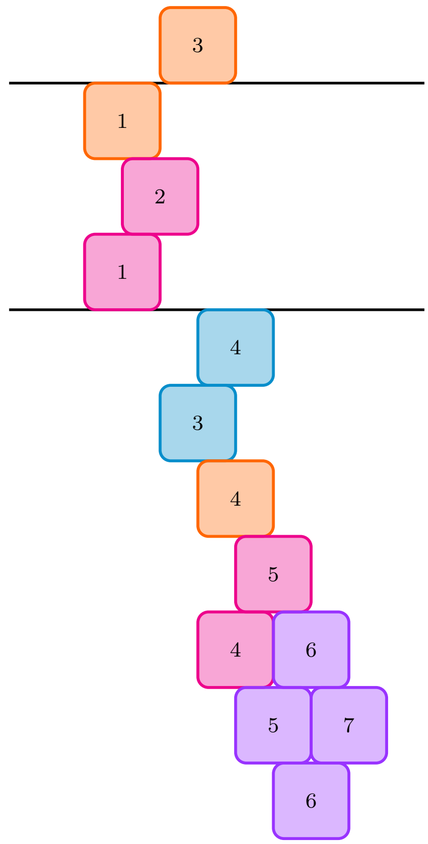


Figure 2.4: Product of tracks corresponding to the maximal σ-string decomposition given in Example 2.2.2

Chapter 3

Cardinality of Braid Classes

This chapter begins with a few lemmas in Section 3.1 that are used to prove our main result, namely Theorem 3.2.1. We conclude with several open problems in Section 3.3.

3.1 Preparatory Lemmas

First, we establish some notation. We use τ or τ_i to denote a σ-string of length 3 and α or α_i to denote a σ-string of length 5. Throughout this chapter, we assume that the number of generators of $W(A_n)$ is sufficiently large. That is, we assume that each group has enough generators for any of our constructions.

Our first lemma concerns a single σ-string, which necessarily has odd length.

Lemma 3.1.1. Let w_1 and w_2 be reduced expressions for $w_1, w_2 \in W(A_n)$ such that w_1 is equal to a single σ-string with length greater than or equal to 9 and w_2 is a reduced expression such that

$$\text{Struct}(\mathsf{w}_2) = \begin{cases} \{\underbrace{3, \ldots, 3}_{j}\} & \text{if } \ell(w_1) = 3j \geq 9 \text{ with } j \text{ odd} \\ \{5, 5, \underbrace{3, \ldots, 3}_{j-3}\} & \text{if } \ell(w_1) = 3j + 1 \geq 13 \text{ with } j \text{ even} \\ \{5, \underbrace{3, \ldots, 3}_{j-1}\} & \text{if } \ell(w_1) = 3j + 2 \geq 11 \text{ with } j \text{ odd.} \end{cases}$$

Then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$.

Proof. First, we see that $\ell(w_1) = \ell(w_2)$. Note, since w_1 is a single string, the length of w_1, and hence w_2, must be odd. By Lemma 2.1.4, we see that

$$\star(\mathsf{w}_1) = \begin{cases} \dfrac{3j+1}{2} & \text{if } \ell(w_1) = 3j \geq 9 \text{ with } j \text{ odd} \\ \dfrac{3j+2}{2} & \text{if } \ell(w_1) = 3j+1 \geq 13 \text{ with } j \text{ even} \\ \dfrac{3j+3}{2} & \text{if } \ell(w_1) = 3j+2 \geq 11 \text{ with } j \text{ odd.} \end{cases}$$

But, by Corollary 2.2.4,

$$\star(\mathsf{w}_2) = \begin{cases} 2^j & \text{if } \ell(w_1) = 3j \geq 9 \text{ with } j \text{ odd} \\ 3^2 \cdot 2^{j-3} & \text{if } \ell(w_1) = 3j+1 \geq 13 \text{ with } j \text{ even} \\ 3 \cdot 2^{j-1} & \text{if } \ell(w_1) = 3j+2 \geq 11 \text{ with } j \text{ odd,} \end{cases}$$

and since $\frac{3j+1}{2} < 2^j$ for all odd $j > 1$, $\frac{3j+2}{2} < 3^2 \cdot 2^{j-3}$ for all even $j > 2$, and $\frac{3j+3}{2} < 3 \cdot 2^{j-1}$ for all odd $j > 1$, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$. □

We see that Lemma 3.1.1 fails to handle strings of certain small odd lengths, specifically the lengths $1, 3, 5$, and 7. However, due to the nature of their construction and small size, these lengths can be dealt with via brute force, which is done in what follows. Although the statement of Lemma 3.1.1 could potentially be modified to accommodate these particular hitches, it is our belief that Lemma 3.1.1 is more intuitive as stated. Because we will need to handle both even and odd lengths, the next lemma handles a few small, special cases involving both even and odd lengths.

Lemma 3.1.2. If $w \in W(A_n)$, then

$$\circledast(w) \leq \begin{cases} 1 & \text{if } \ell(w) \in \{1, 2\} \\ 2 & \text{if } \ell(w) \in \{3, 4\} \\ 3 & \text{if } \ell(w) = 5 \\ 4 & \text{if } \ell(w) \in \{6, 7\} \\ 6 & \text{if } \ell(w) = 8 \\ 9 & \text{if } \ell(w) = 10, \end{cases}$$

and in each case, there exists a w attaining the upper bound.

Proof. Let w be a reduced expression for $w \in W(A_n)$. We consider several cases, and only those constructions for which $\star(\mathsf{w}) > 1$ when $\ell(w) \geq 3$. In the proof of this lemma, we use string instead of maximal σ-string.

Case 1: If $\ell(w) \in \{1,2\}$, then it is clear that no reduced expression can contain a braid.

Case 2: Assume $\ell(w) = 3$. There are two possible decompositions. First, w is a product of three strings with length 1. Since w is reduced, then w would contain no braids (this construction occurs in the remainder of cases and is not considered for the reason above). Second, w may be one string with length 3, in which case $\circledast(w) = 2$ by Corollary 2.2.4.

Case 3: Assume $\ell(w) = 4$. There is only one nontrivial construction (that is, a construction with $\star(\mathsf{w}) > 1$), which is the product of one string of length 3 and one string of length 1. Then $\circledast(w) = 2$.

Case 4: Assume $\ell(w) = 5$. There are two nontrivial constructions. First, w may be the product of two strings of length 1 and one of length 3. Then $\circledast(w) = 2$. Alternatively, w may be be constructed with one string of length 5, which means $\circledast(w) = 3$.

Case 5: Assume $\ell(w) = 6$. There are three possible constructions: one string of length 3 and three strings of length 1, one string of length 5 and one string of length 1, or two strings of length 3. Then $\circledast(w)$ is $2, 3$, and 4, respectively.

Case 6: Assume $\ell(w) = 7$. There are four possible constructions: one string of length 3 and four strings of length 1, one string of length 5 and two strings of length 1, one string of length 7, and two strings of length 3 and one string of length 1. Then $\star(\mathsf{w})$ is $2, 3, 4$, and 4, respectively. This case is particularly interesting and will be discussed later.

Case 7: Assume $\ell(w) = 8$. There are five possible constructions: one string of length 3 and five strings of length 1, one string of length 5 and three strings of length 1, one string of length 7 and one string of length 1, two strings of length 3 and two strings of length 1, and one string of length 5 and one string of length 3. Then $\circledast(w)$ is $2, 3, 4, 4$, and 6, respectively.

Case 8: Assume $\ell(w) = 10$. There are four possible constructions involving three or more strings of length 1 and will not be considered as they are addressed above. There are five remaining constructions: three strings of length 3 and one string of length 1, one string of length 3, one string of length 5, and two strings of length 1, one string of length 7 and one string of length 3, one string of length 9 and one string of length 1, and two strings of length 5. Then $\circledast(w)$ is $8, 6, 8, 5$, and 9, respectively. $\square$

As mentioned above, Case 6 is special. This is because a single σ-string of length seven belongs to a braid class of size 4, as does the product of two σ-strings of length three with an additional generator preceding the strings, sandwiched in the middle of the σ-strings, or ending the expression. This issue arises later.

The next lemma of this section provides an algorithm for obtaining a reduced expression that belongs to a braid class that is at least as large as the braid class for a given reduced expression of any length by reconstituting maximal σ-strings of the lengths from Lemma 3.1.1 or the exceptions from Lemma 3.1.2. That is, given a maximal σ-string decomposition, we will potentially replace it by systematically swapping out larger σ-strings with products of smaller σ-strings. This likely involves utilizing additional Coxeter generators.

Lemma 3.1.3. Let w_1 and w_2 be reduced expressions for $w_1, w_2 \in W(A_n)$ such that $\sigma_1\sigma_2\cdots\sigma_m$ is a maximal σ-string decomposition for w_1 where each $\ell(\sigma_i) = k_i$ and $\mathsf{w}_2 = \widehat{\sigma_1}\widehat{\sigma_2}\cdots\widehat{\sigma_m}$ where each $\widehat{\sigma_i}$ is a reduced expression with maximal σ-string decomposition satisfying

$$\mathrm{Struct}(\widehat{\sigma_i}) = \begin{cases} \{\underbrace{3,\ldots,3}_{j}\} & \text{if } k_i = 3j \text{ and } k_i \geq 3 \text{ with } j \text{ odd} \\ \{5,5,\underbrace{3,\ldots,3}_{j-3}\} & \text{if } k_i = 3j+1 \text{ and } k_i \geq 9 \text{ with } j \text{ even} \\ \{5,\underbrace{3,\ldots,3}_{j-1}\} & \text{if } k_i = 3j+2 \text{ and } k_i \geq 5 \text{ with } j \text{ odd} \\ \{1,3,3\} & \text{if } k_i = 7 \end{cases}$$

If $\widehat{\sigma_1}\widehat{\sigma_2}\cdots\widehat{\sigma_m}$ results in a maximal σ-string decomposition for w_2, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$ when at least one $k_i \geq 9$ and $\star(\mathsf{w}_1) = \star(\mathsf{w}_2)$, otherwise.

Proof. First, observe that we can guarantee the existence of such a w_2 if n is sufficiently large. Second, $\widehat{\sigma_i}$ is not necessarily a σ-string; rather, it is a product of σ-strings. The result now follows from Corollary 2.2.4, Lemma 3.1.1, and Lemma 3.1.2. □

If w is a reduced expression for $w \in W(A_n)$, let $\mu_k(\mathsf{w})$ denote the number of times a k appears in $\mathrm{Struct}(\mathsf{w})$. This definition is used in the next three lemmas, which all follow immediately from Corollary 2.2.4. The next lemma demonstrates that having too many σ-strings of length one does not maximize $\star$.

Lemma 3.1.4. Let w_1 and w_2 be reduced expressions for $w_1, w_2 \in W(A_n)$ with $\ell(w_1) = \ell(w_2)$. If $\mu_1(\mathsf{w}_1) = 3k+r$ with $k > 0$ and $r \in \{0,1,2\}$ and $\mathrm{Struct}(\mathsf{w}_2)$ is identical to $\mathrm{Struct}(\mathsf{w}_1)$ except $\mu_1(\mathsf{w}_2) = r$ and $\mu_3(\mathsf{w}_2) = \mu_3(\mathsf{w}_1) + k$, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$. In particular, $\circledast(w_1)$ is increased by a factor of 2^k.

The following lemma shows that $\star$ is not maximized in the presence of either one or two σ-strings of length one and sufficiently many σ-strings of length 3.

Lemma 3.1.5. Let w_1 and w_2 be reduced expressions for $w_1, w_2 \in W(A_n)$. If $\mu_3(\mathsf{w}_1) \geq 3$, $\mu_1(\mathsf{w}_1) = 1$, and $\mathrm{Struct}(\mathsf{w}_1)$ is identical to $\mathrm{Struct}(\mathsf{w}_2)$ except $\mu_5(\mathsf{w}_2) = \mu_5(\mathsf{w}_1)+2$, $\mu_1(\mathsf{w}_2) = 0$, and $\mu_3(\mathsf{w}_2) = \mu_3(\mathsf{w}_1) - 3$, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$. Further, if $\mu_3(\mathsf{w}_1) \geq 1$, $\mu_1(\mathsf{w}_1) = 2$, and $\mathrm{Struct}(\mathsf{w}_1)$ is identical to $\mathrm{Struct}(\mathsf{w}_2)$ except $\mu_5(\mathsf{w}_2) = \mu_5(\mathsf{w}_1) + 1$, $\mu_1(\mathsf{w}_2) = 0$, and $\mu_3(\mathsf{w}_2) = \mu_3(\mathsf{w}_1) - 1$, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$.

The final lemma plays an important role in increasing $\star$ in Theorem 3.2.1. In particular, $\star$ of a maximal string decomposition containing five σ-strings of length 3 is greater than $\star$ of a maximal string decomposition containing three σ-strings of length 5 where the string structures are otherwise the same.

Lemma 3.1.6. Let w_1 and w_2 be reduced expressions for $w_1, w_2 \in W(A_n)$. If $\mu_5(\mathsf{w}_1) \geq 3$ and $\text{Struct}(\mathsf{w}_1)$ is identical to $\text{Struct}(\mathsf{w}_2)$ except $\mu_5(\mathsf{w}_2) = \mu_5(\mathsf{w}_1) - 3$ and $\mu_3(\mathsf{w}_2) = \mu_3(\mathsf{w}_1) + 5$, then $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$.

3.2 Main Result

In this section, we let $\mathfrak{n}$ be a natural number and define $m(\mathfrak{n}) = \max\big(\circledast(w)\big)$ and w ranges over all permutations of length $\mathfrak{n}$. The following theorem is Corollary 7 from [11], but with a new proof that pieces together the new lemmas given in Section 3.1. In the proof of the following theorem, we refer to the replacement of generators, where this replacement may involve generators yet to appear.

Theorem 3.2.1. Let $\mathfrak{n}$ be a natural number. Then

$$m(\mathfrak{n}) = \begin{cases} 1 & \text{if } \mathfrak{n} \in \{1,2\} \\ 2^j & \text{if } \mathfrak{n} = 3j \text{ with } j \geq 1 \text{ or } \mathfrak{n} = 3j+1 \text{ with } j \leq 2 \\ 2^{j-3} \cdot 3^2 & \text{if } \mathfrak{n} = 3j+1 \text{ with } j \geq 3 \\ 2^{j-1} \cdot 3 & \text{if } \mathfrak{n} = 3j+2 \text{ with } j \geq 1. \end{cases}$$

Proof. The proof of Lemma 3.1.2 shows how to maximize $\circledast$ for group elements with lengths less than 9.

For the remainder of the proof, suppose $\mathsf{w} = \sigma_1 \cdots \sigma_r$ is a maximal σ-string decomposition for a reduced expression for $w \in W(A_n)$ and let $\ell(w) = \mathfrak{n} \geq 9$. The proof is handled in steps that will systemically replace w with new reduced expressions that increase $\star$ and eventually guarantees that $\star$ achieves the maximum. If a step requires no action, we simply relabel accordingly. Also, note that, when doing replacements, we may have to replace generators in surrounding σ-strings to maintain the original length.

Step 1: First, we utilize Lemma 3.1.2 for each σ_i with length in $\{1,3,5,7\}$ and Lemma 3.1.3 for each σ_i with length greater than or equal to 9 to replace each σ_i with some new σ-string that potentially increases $\star$. That is, for each σ_i with relatively small length, σ_i is replaced by the maximizing constructions given in Lemma 3.1.2, and for each σ_1 with relatively large length, σ_i is replaced by a product of σ-strings (the $\widehat{\sigma}$'s appearing in the lemma) that individually increase $\star$. This replacement likely involves generators not appearing in $\text{supp}(w)$.

We call this reduced expression w_1 and it is made up entirely of σ-strings of lengths 1, 3, and 5 and $\star(\mathsf{w}) \leq \star(\mathsf{w}_1)$ with equality if and only if each $\ell(\sigma_i) \leq 7$.

Step 2: We now examine w_1 in the context of Lemma 3.1.4. Step 1 potentially leaves many σ-strings of length one. We collect these particular σ-strings and form a new reduced expression w_2 via replacement by the maximizing construction from the lemma. Again, we likely do this swap with generators yet to appear. This process will leave 0, 1, or 2 σ-strings of length 1. That is, $\mu_1(\mathsf{w}_2) \in \{0, 1, 2\}$. Also, $\star(\mathsf{w}_1) < \star(\mathsf{w}_2)$

Step 3: Now that w_2 is made up entirely of σ-strings of lengths 3 and 5 with up to two σ-strings of length 1, we create some w_3 according to Lemma 3.1.5. If $\mu_1(\mathsf{w}_2) = 1$, then $\mu_5(\mathsf{w}_3)$ is increased by two while $\mu_3(\mathsf{w}_3)$ is decreased by 3 and $\mu_1(\mathsf{w}_3)$ is decreased by 1, both relative to those measures for w_2. If $\mu_1(\mathsf{w}_2) = 2$, then $\mu_5(\mathsf{w}_3)$ is increased by 1 while $\mu_3(\mathsf{w}_3)$ is decreased by 1 and $\mu_1(\mathsf{w}_3)$ is decreased by 2, both relative to those measures of w_2. The new reduced expression w_3 contains only σ-strings of length 3 and 5 and $\star(\mathsf{w}_2) < \star(\mathsf{w}_3)$. Note, if Step 2 created w_2 such that $\mu_1(\mathsf{w}_2) = 0$, no change is made to w_3.

Step 4: Finally, we create w_4 from w_3 by iteratively replacing three σ-strings of length 5 with five σ-strings of length 3. This process will terminate with $\mu_5(\mathsf{w}_4) \in \{0, 1, 2\}$.

Let w_4 be the group element that corresponds to w_4. If $\ell(w_4) = 3j$, $\mu_5(\mathsf{w}_4) = 0$ and $\star(\mathsf{w}_4) = 2^j$. Alternatively, if $\ell(w_4) = 3j+1$, $\mu_5(\mathsf{w}_4) = 2$ and $\star(\mathsf{w}_4) = 2^{j-3} \cdot 3^2$ or if $\ell(w_4) = 3j+2$, $\mu_5(\mathsf{w}_4) = 1$ and $\star(\mathsf{w}_4) = 2^{j-1} \cdot 3$.

We claim that w_4 maximizes $\circledast$ for length $\mathfrak{n}$. Lemmas 3.1.3, 3.1.4, and 3.1.5 guarantee that for elements with length at length nine, the string structure for an element that maximizes $\circledast$ will not contain a 1 nor a number larger than 5. This implies that the string structure for any element with length at least nine that maximizes $\circledast$ will consist solely of 3's and 5's. Lemma 3.1.6 confirms that the structure of w_4 optimizes our use of 3's and 5's. $\square$

We note here that this result corresponds to the maximum weight of a vertex in all possible braid graphs for elements with length $\mathfrak{n}$.

3.3 Open Problems

Theorem 3.2.1 leads naturally to the following open problem: what is the smallest possible group for which $m(\mathfrak{n})$ is achieved? For example, to maximize $\circledast(w)$ for all w with length 11, a maximal σ-string decomposition for a reduced expression for such a w is made of 2 σ-strings of length 3 and 1 σ-string of length 5 via Theorem 3.2.1. One possible reduced expression is 121|343|56576, which corresponds to a group element in at least $W(A_7)$. However, it may be the case that there exist elements in smaller groups with braid classes that attain the same cardinality.

For a brief period at the beginning of our research, we tried to investigate the so-called *evolution* of commutation/braid/Matsumoto graphs in relation to a left (or right) Bruhat ordering. That is, we created graphs, called the expanded Bruhat graph, via the left (or right) Bruhat ordering with the vertex set as the collection of all commutation (or braid or Matsumoto) graphs where a vertex is connected to another via the corresponding Bruhat ordering. While these graphs are easily obtainable by brute force for small examples, due to the immense number of elements in groups like $W(A_{10})$, we hoped to be able to predict the commutation/braid/Matsumoto graph for a certain element by inspecting the commutation/braid/Matsumoto's graphs below it in the Bruhat ordering. However, we were unsuccessful and hope this idea can be looked into further.

In Section 1.3, fully commutative (FC) elements were defined to be those elements with a unique commutation class. Recall that an element $w \in W$ is FC if and only if no reduced expression contains a braid. We analogously define *commutation free* (CF) elements: when the set of reduced expressions for $w \in W$ has only a single braid class, then w is said to be commutation free. Similarly, an element $w \in W$ is CF if and only if no reduced expression gives way to a commutation move. Unlike the FC property, the CF property has received little to no attention. For instance, it is known that an element is FC if and only if it is 321-avoiding. Does any such pattern avoidance classify CF elements, or, in general, what is the classification of CF elements? This opens many other potential problems. For example, we believe classifying the left (or right) Bruhat graphs with nontrivial CF elements would be interesting. Additionally, if one were given a Coxeter graph with some quality (such as a 3-cycle), what do CF elements look like and what are their possible lengths? Work has been done investigating FC finite Coxeter groups (that is, Coxeter groups with a finite number of FC elements) and FC infinite Coxeter groups (Coxeter groups with an infinite number of FC elements). The analogous question concerning CF finite and CF infinite is potentially interesting. That is, what Coxeter groups are CF infinite versus CF finite?

Bibliography

[1] S.C. Billey and B.C. Jones. Embedded factor patterns for Deodhar elements in Kazhdan–Lusztig theory. *Ann. Comb.*, 11(3–4):285–333, 2007.

[2] D.C. Ernst. Non-cancellable elements in type affine C Coxeter groups. *Int. Electron. J. Algebr.*, 8:191–218, 2010.

[3] S. Felsner. On the Number of Arrangements of Pseudolines. *Discret. Comput. Geom.*, 18(3), 1997.

[4] M. Geck and G. Pfeiffer. *Characters of finite Coxeter groups and Iwahori–Hecke algebras.* 2000.

[5] J.E. Humphreys. *Reflection Groups and Coxeter Groups.* Cambridge University Press, Cambridge, 1990.

[6] M.M. Kapranov and V.A. Voevodsky. Combinatorial-geometric aspects of polycategory theory: pasting schemes and higher Bruhat orders (list of results). *Cah. Topol. géométrie différentielle catégoriques*, 32(1), 1991.

[7] D. Knuth. *Axioms and Hulls.* Springer-Verlag, Berlin, 1992.

[8] R.P. Stanley. On the number of reduced decompositions of elements of Coxeter groups. *Eur. J. Comb.*, 5(4), 1984.

[9] J.R. Stembridge. On the fully commutative elements of Coxeter groups. *J. Algebr. Comb.*, 5, 1996.

[10] B.E. Tenner. Reduced decompositions and permutation patterns. *J. Algebr. Comb.*, 24(3):263–284, 2006.

[11] D.M. Zollinger. *Equivalence classes of reduced words.* Ms thesis, University of Minnesota, 1994.