

A STUDY OF T-AVOIDING ELEMENTS OF COXETER GROUPS

By Taryn Laird

A Thesis

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Mathematics

Northern Arizona University

May 2016

Approved:

Dana Ernst, Ph.D., Chair

Michael Falk, Ph.D.

Nándor Sieben, Ph.D.

ABSTRACT

A STUDY OF T-AVOIDING ELEMENTS OF COXETER GROUPS

Taryn Laird

Coming soon. We are finishing writing everything else first so we know the appropriate roadmap to be in this part.

Contents

List of Figures	v
Chapter 1 Preliminaries	1
1.1 Coxeter Systems	1
1.2 Fully Commutative Elements	7
1.3 Heaps	9
Chapter 2 Star Reductions and Property T	13
2.1 Star Reductions	13
2.2 Property T	16
2.3 Non-Cancellable Elements	18
Chapter 3 T-Avoiding Elements in Types $\tilde{A}_n, A_n, D_n, F_n$, and $I_2(m)$	22
3.1 Types \tilde{A}_n and A_n	22
3.2 Type D_n	23
3.3 Type F_n	24
3.4 Type $I_2(m)$	27
Chapter 4 T-Avoiding Elements in Type B_n	28
4.1 Tools for the Classification	28
4.2 Classification of T-Avoiding Elements in Type B_n	30
Chapter 5 T-Avoiding Elements in Type \tilde{C}_n	38
5.1 Classification of T-Avoiding Elements in $W(\tilde{C}_n)$	38
5.2 Future Work	47
Bibliography	48

List of Figures

1.1	A few labeled Coxeter graphs.	2
1.2	Irreducible finite Coxeter systems.	4
1.3	Irreducible affine Coxeter systems.	5
1.4	Irreducible FC-finite Coxeter systems.	8
1.5	Labeled hasse diagram for the heap of an element in $\text{FC}(\tilde{C}_6)$	9
1.6	A lattice point representation for the heap of an FC element in $W(\tilde{C}_6)$	11
1.7	Two heaps of a non-FC element in $W(\tilde{C}_4)$	11
1.8	Impermissible configurations for heaps of $\text{FC}(\tilde{C}_n)$	12
2.1	A visual representation of an element that is left star reducible.	14
2.2	Visualization of Example 2.1.1.	15
2.3	Visualization of Example 2.1.3	15
2.4	Heap of an element with Property T.	17
2.5	Heap of a non-trivially T-avoiding element in $W(\tilde{C}_4)$	18
2.6	Heap of a weak star reducible element of $\text{FC}(B_4)$	19
2.7	Heap of a non-cancellable element of $\text{FC}(B_4)$	20
2.8	Heap of a single sandwich stack in $W(\tilde{C}_n)$	20
2.9	Heap of a stack of sandwich stacks in $W(\tilde{C}_n)$	20
3.1	Visual representation of non-trivial T-avoiding elements in $W(D_5)$	24
3.2	Heap of a single bowtie in $W(F_5)$	25
3.3	Heap of a stack of bowties in $W(F_5)$	26
3.4	Heap of a non-trivial T-avoiding element in $W(F_6)$	27
5.1	Visual representation of the heap configuration discussed in Case (1)a.	39
5.2	Visual representation of the heap configuration discussed in Case (1)b.	40
5.3	Visual representation of the heap configuration discussed in Case (2)a.	41
5.4	Visual representation of the heap configuration discussed in Case (2b).	41
5.5	42
5.6	42
5.7	43

5.8	44
5.9	Visual Representation of the heap configuration discussed in Case (2b).	45
5.10	Visual representation of the heap configuration discussed in Case (3)a.	45
5.11	Visual representation of the heap configuration discussed in Case (3)b.	46
5.12	Visual representation of the heap configuration discussed in Case (4).	46

Chapter 1

Preliminaries

1.1 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

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

where e is the identity, $m(s,t) = 1$ if and only if $s = t$, and $m(s,t) = m(t,s)$. It turns out that the elements of S are distinct as group elements and that $m(s,t)$ is the order of st [10]. We call $m(s,t)$ the *bond strength* of s and t .

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

$$\underbrace{sts \cdots}_{m(s,t)} = \underbrace{tst \cdots}_{m(s,t)} \quad (1.1)$$

with $m(s,t) \geq 2$ factors. If $m(s,t) = 2$, then $st = ts$ is called a *commutation relation*. Otherwise, if $m(s,t) \geq 3$, then the relation in (1.1) is called a *braid relation*. Replacing $\underbrace{sts \cdots}_{m(s,t)}$ with $\underbrace{tst \cdots}_{m(s,t)}$ will be referred to as a *commutation* 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 unique *Coxeter graph* Γ having

- (1) vertex set S and
- (2) edges $\{s, t\}$ for each $m(s,t) \geq 3$ labeled by its corresponding bond strength.

Since $m(s,t) = 3$ occurs frequently, it is customary to omit this label. Note that s and t are not connected by a single edge in the graph if and only if $m(s,t) = 2$. There

is a one-to-one correspondence between Coxeter systems and Coxeter graphs. That is, given a Coxeter graph Γ , we can uniquely 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 $S(\Gamma)$ for clarity. Also, the Coxeter system (W, S) is said to be *irreducible* if and only if Γ is connected. If the graph Γ is disconnected, the connected components correspond to factors in a direct product of the corresponding Coxeter groups [10]. The Coxeter graphs given in Figure 1.1 correspond to the Coxeter systems that will be primarily addressed in this thesis. Notice that the vertices are labeled with the corresponding generators to provide context when talking about the different generating sets $S(\Gamma)$.

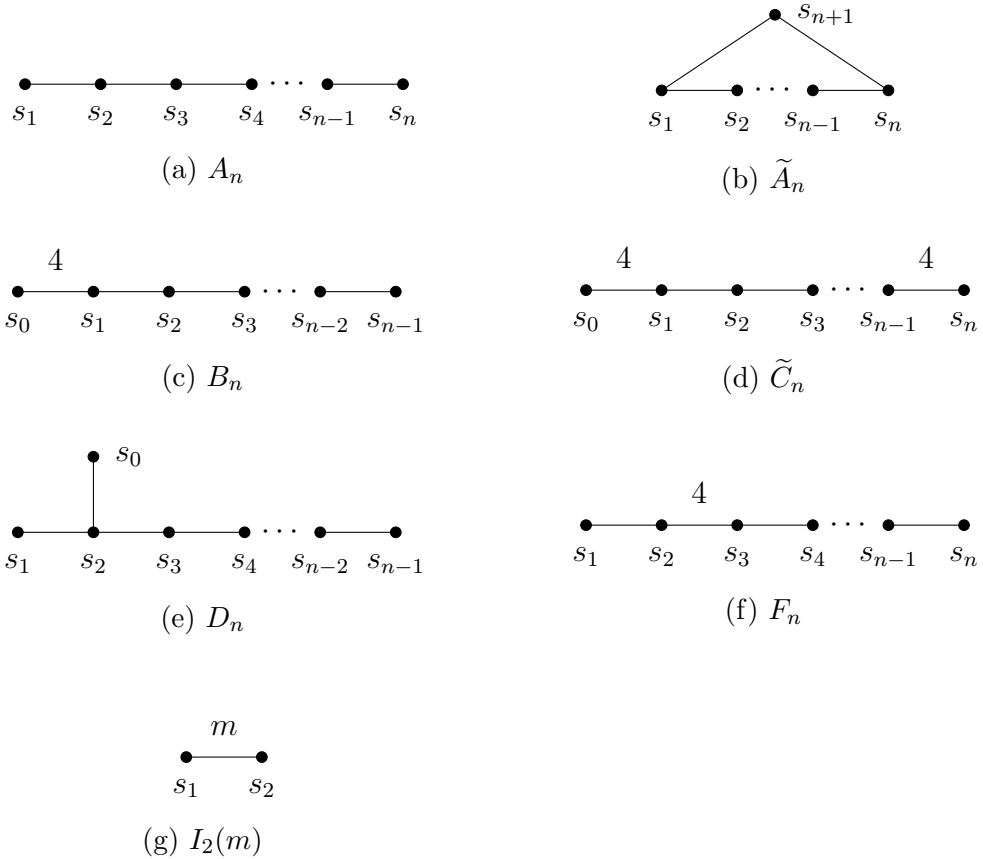


Figure 1.1: A few labeled Coxeter graphs.

Example 1.1.1.

- (a) The Coxeter system of type A_n is given by the graph in Figure 1.1(a). We can construct the corresponding Coxeter group $W(A_n)$ with generating set $S(A_n) = \{s_1, s_2, \dots, s_n\}$ 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 Sym_{n+1} under the correspondence $s_i \mapsto (i, i + 1)$, where $(i, i + 1)$ is the adjacent transposition that swaps i and $i + 1$.

- (b) The Coxeter system of type B_n is given by the graph in Figure 1.1(c). We can construct the corresponding Coxeter group $W(B_n)$ with generating set $S(B_n) = \{s_0, s_1, \dots, s_{n-1}\}$ 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 $i, j \in \{1, 2, \dots, n - 1\}$;
- (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 Sym_n^B , where Sym_n^B is the group of signed permutations on the set $\{1, 2, \dots, n\}$.

- (c) The Coxeter system of type \tilde{C}_n is seen in Figure 1.1(d). We can construct the corresponding Coxeter group $W(\tilde{C}_n)$ with generating set $S(\tilde{C}_n) = \{s_0, s_1, \dots, s_n\}$ and defining relations

- (1) $s_i^2 = e$ for all i ;
- (2) $s_i s_j = s_j s_i$ when $|i - j| > 1$ for $i \in \{0, 2, \dots, n\}$;
- (3) $s_i s_j s_i = s_j s_i s_j$ when $|i - j| = 1$ for $i \in \{1, 2, \dots, n - 1\}$;
- (4) $s_0 s_1 s_0 s_1 = s_1 s_0 s_1 s_0$;
- (5) $s_n s_{n-1} s_n s_{n-1} = s_{n-1} s_n s_{n-1} s_n$.

Note that $W(\tilde{C}_n)$ has $n + 1$ generators.

The Coxeter graphs given in Figure 1.2 correspond to the collection of irreducible finite Coxeter systems, whose corresponding Coxeter groups are finite, while the Coxeter graphs given in Figure 1.3 are the so-called irreducible *affine Coxeter systems*, which are infinite [10]. Note that $W(B_n)$ is one of the irreducible finite Coxeter groups so it is finite while $W(\tilde{C}_n)$ is one of the affine groups making it infinite. The irreducible affine Coxeter systems are unique in that if a vertex is removed along with the corresponding edges from the Coxeter graph, the newly created graph will result in a Coxeter system with a finite Coxeter group.

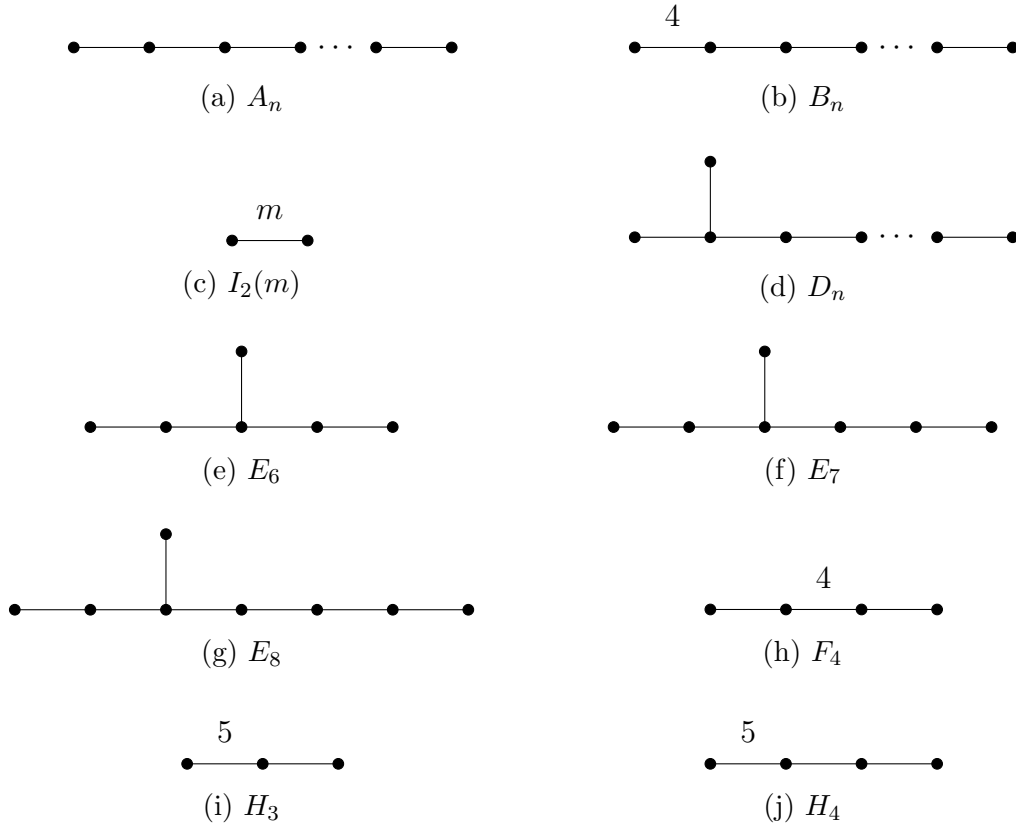


Figure 1.2: Irreducible finite Coxeter systems.

Given a Coxeter system (W, S) , a word $s_{x_1}s_{x_2}\cdots s_{x_m}$ in the free monoid S^* on 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 . In this case, we define the *length* of w to be $l(w) := m$. Each element $w \in W$ may have multiple reduced expressions that represent it. If we wish to emphasize a specific, possibly reduced, expression for $w \in W$ we will represent it as $\mathbf{w} = s_{x_1}s_{x_2}\cdots s_{x_m}$ (using **sans serif font**). If $u, v \in W(\Gamma)$, we say that the product

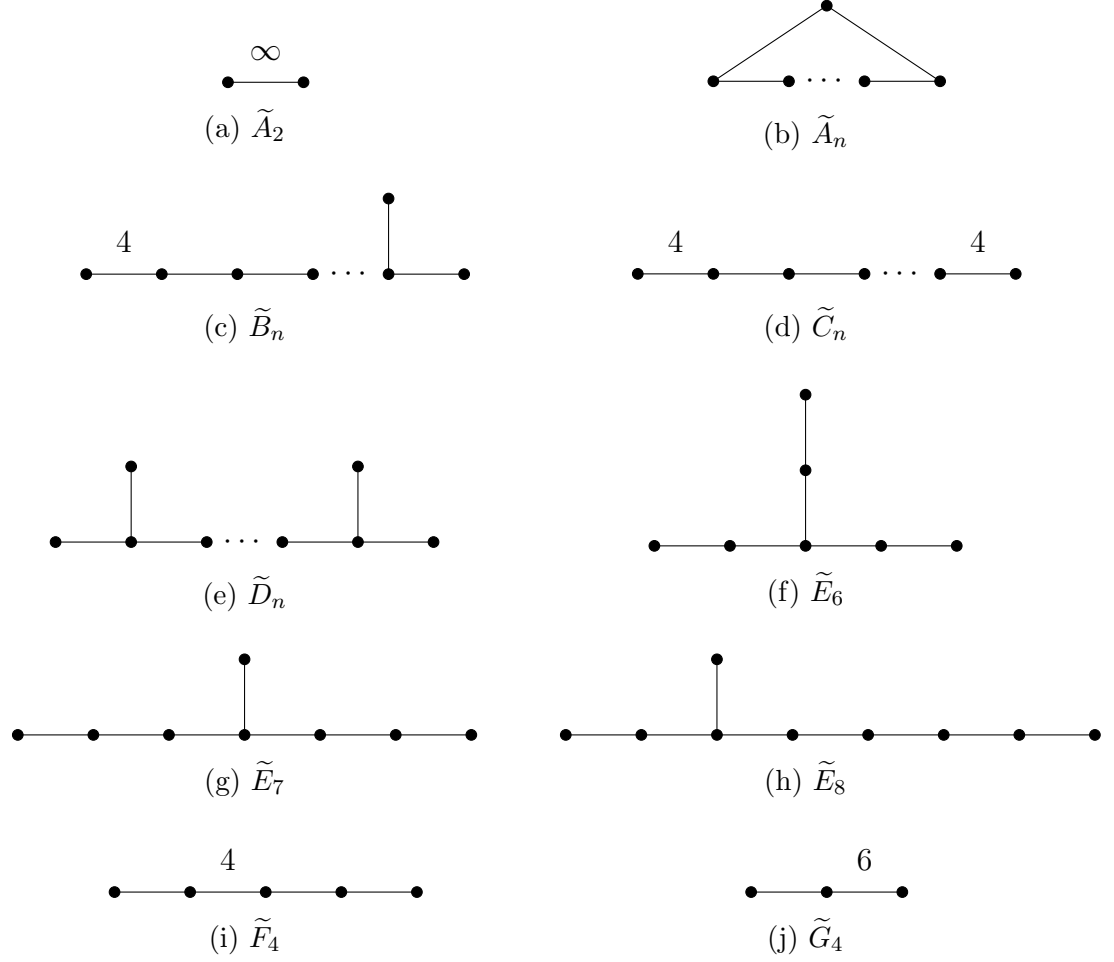


Figure 1.3: Irreducible affine Coxeter systems.

of group elements uv is *reduced* if $l(uv) = l(u) + l(v)$. Matsumoto's Theorem, which follows, tells us more about how reduced expressions for a given group element are related.

Proposition 1.1.2 (Matsumoto, [7]). Let (W, S) be a Coxeter system. If $w \in W$, then given a reduced expression for w we can obtain every other reduced expression for w by a sequence of braid moves and commutations of the form

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

where $s, t \in S$ and $m(s, t) \geq 2$. □

It follows from Matsumoto's Theorem that if a generator s appears in a reduced expression for $w \in W$, then s appears in all reduced expressions for w . Let $w \in W$ and define the *support* of w , denoted $\text{supp}(w)$, to be the set of all generators that appear in any reduced expression for w . If $\text{supp}(w) = S$, we say that w has *full support*.

Given $w \in W$ and a fixed reduced expression \mathbf{w} for w , any subsequence of \mathbf{w} is called a *subexpression* of \mathbf{w} . We will refer to a subexpression consisting of a consecutive subsequence of \mathbf{w} as a *subword* of \mathbf{w} .

Example 1.1.3. Let $w \in W(A_7)$ and let $\mathbf{w} = s_7 s_2 s_4 s_5 s_3 s_2 s_3 s_6$ be a fixed expression for w . Then we have

$$\begin{aligned} s_7 s_2 s_4 s_5 s_3 s_2 s_3 s_6 &= s_7 s_4 s_2 s_5 s_3 s_2 s_3 s_6 \\ &= s_7 s_4 s_5 s_2 s_3 s_2 s_3 s_6 \\ &= s_7 s_4 s_5 s_3 s_2 s_3 s_3 s_6 \\ &= s_7 s_4 s_5 s_3 s_2 s_6, \end{aligned}$$

where the **purple** highlighted text corresponds to a commutation, the **teal** highlighted text corresponds to a braid move, and the **red** highlighted text corresponds to cancellation. This shows that the original expression \mathbf{w} is not reduced. However, it turns out that $s_7 s_4 s_5 s_3 s_2 s_6$ is reduced. Thus $l(w) = 6$ and $\text{supp}(w) = \{s_2, s_3, s_4, s_5, s_6, s_7\}$.

Let (W, S) be a Coxeter system of type Γ and let $w \in W(\Gamma)$. We define the *left descent set* and *right descent set* of w as follows:

$$\mathcal{L}(w) := \{s \in S \mid l(sw) < l(w)\}$$

and

$$\mathcal{R}(w) := \{s \in S \mid l(ws) < l(w)\}.$$

In [2] it is shown that $s \in \mathcal{L}(w)$ (respectively, $\mathcal{R}(w)$) if and only if there is a reduced expression for w that begins (respectively, ends) with s .

Example 1.1.4. The following list consists of all reduced expressions some $w \in W(B_4)$:

$$\begin{array}{cc} s_0 s_1 s_2 s_1 s_3 & s_0 s_2 s_1 s_2 s_3 \\ s_0 s_1 s_2 s_3 s_1 & s_2 s_0 s_1 s_2 s_3 \end{array}$$

We see that $l(w) = 5$ and w has full support. Also, we see that $\mathcal{L}(w) = \{s_0, s_2\}$ while $\mathcal{R}(w) = \{s_1, s_3\}$.

1.2 Fully Commutative Elements

Let (W, S) be a Coxeter system of type Γ and let $w \in W(\Gamma)$. Following [13], we define a relation \sim on the set of reduced expressions for w . Let \mathbf{w}_1 and \mathbf{w}_2 be two reduced expressions for w . We define $\mathbf{w}_1 \sim \mathbf{w}_2$ if we can obtain \mathbf{w}_2 from \mathbf{w}_1 by applying a single commutation move of the form $st \mapsto ts$ where $m(s, t) = 2$. Now, define the equivalence relation \approx by taking the reflexive transitive closure of \sim . Each equivalence class under \approx is called a *commutation class*. If w has a single commutation class, then we say that w is *fully commutative* (FC).

The set of FC elements of $W(\Gamma)$ is denoted by $\text{FC}(\Gamma)$. Given some $w \in \text{FC}(\Gamma)$, and a starting reduced expression for w , observe that the definition of FC states that one only needs to perform commutations to obtain all reduced expressions for w , but the following result due to Stembridge [13] states that when w is FC, performing commutations is the only possible way to obtain another reduced expression for w .

Proposition 1.2.1 (Stembridge, [13]). Let (W, S) be a Coxeter system. An element $w \in W$ is FC if and only if no reduced expression for w contains $\underbrace{sts \cdots}_{m(s,t)}$ as a subword

for all $m(s, t) \geq 3$. □

In other words, w is FC if and only if no reduced expression provides the opportunity to apply a braid move. For example, for a Coxeter system of type B_n an element is FC if no reduced expression contains the subwords $s_0 s_1 s_0 s_1$, $s_1 s_0 s_1 s_0$, $s_k s_{k+1} s_k$, and $s_{k+1} s_k s_{k+1}$ where $0 < k < n - 2$. In a Coxeter system of type \tilde{C}_n , an element is FC if no reduced expression for the element contains the subwords seen above and does not contain the subwords $s_{n-1} s_n s_{n-1} s_n$ and $s_n s_{n-1} s_n s_{n-1}$.

Example 1.2.2. Let $w \in W(\tilde{C}_4)$ and let $\mathbf{w} = s_0 s_1 s_2 s_0 s_3 s_1$ be a reduced expression for w . Although it is not immediately obvious, there is no possible way to perform a braid move in any reduced expression for w . Hence w is FC.

Example 1.2.3. Let $\mathbf{w}_1 = s_1 s_0 s_4 s_1 s_3 s_5 s_2 s_4 s_6$ be a reduced expression for $w \in W(\tilde{C}_6)$. Applying the commutation $s_4 s_2 \mapsto s_2 s_4$, we can obtain another reduced expression for w , namely $\mathbf{w}_2 = s_1 s_0 s_4 s_1 s_3 s_5 s_4 s_2 s_6$, which is in the same commutation class as \mathbf{w} . However, applying the braid move $s_2 s_3 s_2 \mapsto s_3 s_2 s_3$, we obtain another reduced expression $\mathbf{w}_3 = s_1 s_3 s_2 s_3 s_4 s_0$. Note that since \mathbf{w}_3 was obtained by applying a braid move, \mathbf{w}_3 is in a different commutation class than \mathbf{w}_1 and \mathbf{w}_2 . Since w has at least two commutation classes, one containing \mathbf{w}_1 and \mathbf{w}_2 and another containing \mathbf{w}_3 , w is not FC by Proposition 1.2.1.

Stembridge classified the Coxeter systems whose groups contain a finite number of FC elements, the so-called *FC-finite Coxeter groups*. Both $W(A_n)$ and $W(B_n)$

are finite Coxeter groups, and thus are FC-finite. On the other hand, $W(\tilde{C}_n)$ is infinite and happens to also contain infinitely many FC elements. However, there exist infinite Coxeter groups that contain finitely many FC elements. For example, $W(E_n)$ for $n \geq 9$ (see Figure 1.4) is infinite, but contains only finitely many FC elements.

Proposition 1.2.4 (Stembridge, [13]). The irreducible FC-finite Coxeter systems are of type A_n with $n \geq 1$, B_n with $n \geq 2$, D_n with $n \geq 4$, E_n with $n \geq 6$, F_n with $n \geq 4$, H_n with $n \geq 3$, and $I_2(m)$ with $5 \leq m < \infty$. \square

The irreducible FC-finite Coxeter graphs are given in Figure 1.4. Note that the irreducible finite Coxeter systems given in Figure 1.2 certainly have only a finite number of FC elements. We have not yet encountered the Coxeter groups determined by graphs in Figures 1.4(d) for $n \geq 9$, 1.4(e) for $n \geq 5$, 1.4(f) for $n \geq 5$. All of these Coxeter systems are infinite for sufficiently large n , yet contain only finitely many FC elements.

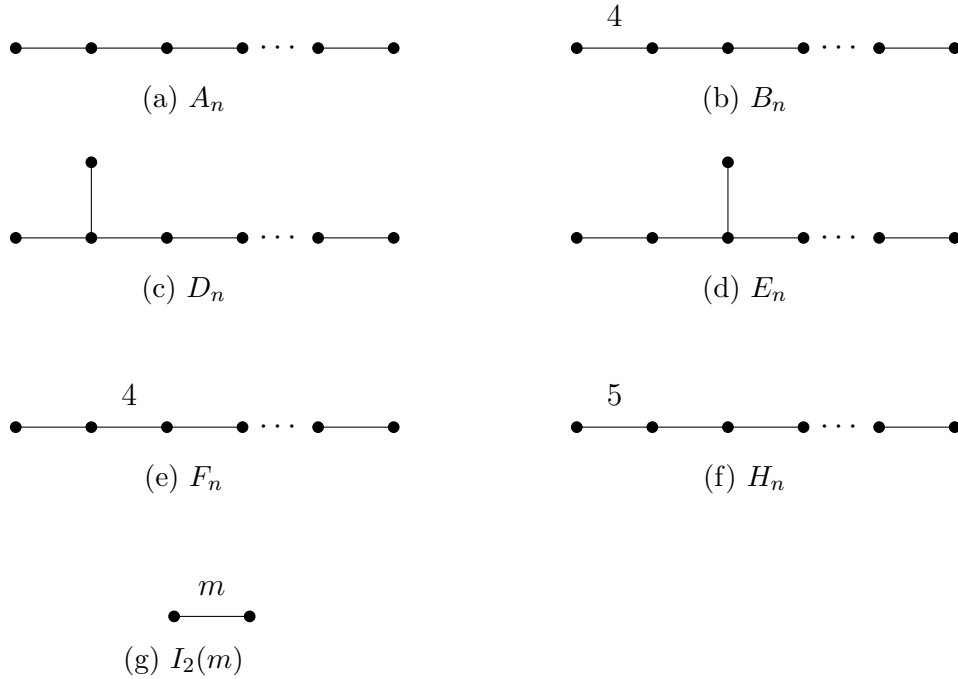


Figure 1.4: Irreducible FC-finite Coxeter systems.

1.3 Heaps

We now discuss a visual representation of Coxeter group elements. Each reduced expression can be associated with a labeled partially ordered set (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], [3], and [13].

Let (W, S) be a Coxeter system of type Γ . Suppose $\mathbf{w} = s_{x_1}s_{x_2}\cdots s_{x_r}$ is a fixed reduced expression for $w \in W(\Gamma)$. As in [13], we define a partial ordering on the indices $\{1, 2, \dots, r\}$ by the transitive closure of the relation \triangleleft defined via $j \triangleleft i$ if $i < j$ and s_{x_i} and s_{x_j} do not commute. In particular, since \mathbf{w} is reduced, $j \triangleleft i$ if $s_{x_i} = s_{x_j}$ by transitivity. This partial order is referred to as the *heap* of \mathbf{w} , where i is labeled by s_{x_i} . Note that for simplicity we are omitting the labels of the underlying poset yet retaining the labels of the corresponding generators.

It follows from [13] that heaps are well-defined up to commutation class. That is, given two reduced expressions \mathbf{w}_1 and \mathbf{w}_2 for $w \in W$ that are in the same commutation class, the heaps for \mathbf{w}_1 and \mathbf{w}_2 will be equal. In particular, if $w \in \text{FC}(\Gamma)$, then w has one commutation class, and thus w has a unique heap. Conversely, if \mathbf{w}_1 and \mathbf{w}_2 are in different commutation classes, then the heap for \mathbf{w}_1 will be distinct from the heap for \mathbf{w}_2 .

Example 1.3.1. Let $\mathbf{w} = s_6s_4s_2s_5s_3s_1s_4s_0s_1$ be a reduced expression for $w \in \text{FC}(\tilde{C}_6)$. We see that \mathbf{w} is indexed by $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. As an example, $9 \triangleleft 8$ since $8 < 9$ and s_0 and s_1 do not commute. The labeled Hasse diagram for the heap poset is seen in Figure 1.5.

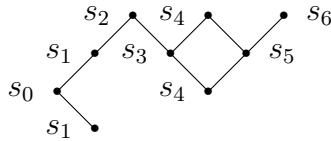


Figure 1.5: Labeled hasse diagram for the heap of an element in $\text{FC}(\tilde{C}_6)$.

Let \mathbf{w} be a reduced expression for an element $w \in W(\tilde{C}_n)$. As in [1] and [3] we can represent a heap for \mathbf{w} as a set of lattice points embedded in $\{0, 1, 2, \dots, n\} \times \mathbb{N}$. To do so, we assign coordinates (not unique) $(x, y) \in \{0, 1, 2, \dots, n\} \times \mathbb{N}$ to each entry of the labeled Hasse diagram for the heap of \mathbf{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 $W(\tilde{C}_n)$, the same construction works for any straight line Coxeter graph with the appropriate adjustments made to the label set and assignment of coordinates. Specifically, for type A_n our label set is $\{1, 2, \dots, n\}$ and for type B_n our label set is $\{0, 1, \dots, n-1\}$.

In the case of any straight line Coxeter graph it follows from the definition 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 that we can completely reconstruct the edges of the Hasse diagram and the corresponding heap poset from a lattice point representation. The lattice point representation can help us visualize arguments that are potentially complex. Note that in our heaps the entries fully exposed to the top (respectively, bottom) correspond to the generators occurring in the left (respectively, right) descent set of the corresponding reduced expression. Since there are multiple heap representations when $w \in W(\Gamma)$ is not FC, it is possible that a block that is fully exposed in one heap may not be fully exposed in a different heap representing w .

Let \mathbf{w} be a reduced expression for $w \in W(\tilde{C}_n)$. We denote the lattice representation of the heap poset in $\{0, 1, 2, \dots, n\} \times \mathbb{N}$ described in the preceding paragraphs via $H(\mathbf{w})$. If w is FC, then the choice of reduced expression for w is irrelevant and we will often write $H(w)$ and we refer to $H(w)$ as the heap of w . Note that we will use the same notation for heaps in Coxeter groups of all types with straightline Coxeter graphs.

Let $\mathbf{w} = s_{x_1} s_{x_2} \cdots s_{x_r}$ be a reduced expression for $w \in W(\tilde{C}_n)$. If s_{x_i} and s_{x_j} are adjacent generators in the Coxeter graph with $i < j$, then we must place the point labeled by s_{x_i} at a level that is *above* the level of the point labeled by s_{x_j} . Because generators in a Coxeter graph that are not adjacent do commute, points whose x -coordinates differ by more than one can slide past each other or land in the same level. To emphasize the covering relations of the lattice point representation we will enclose each entry in the heap in a square with rounded corners (called a block) in such a way that if one entry covers another the blocks overlap halfway. In addition, we will also label each square for s_i with i .

There are potentially many ways to illustrate a heap of an arbitrary reduced expression, each differing by the vertical placement of the blocks. For example, we can place blocks in vertical positions as high as possible, as low as possible, or some combination of low/high. In this thesis, we choose what we view to be the best representation of the heap for each example and when illustrating the heaps of arbitrary reduced expressions we will discuss the relative position of the entries but never the absolute coordinates.

Example 1.3.2. Let $w = s_6 s_4 s_2 s_5 s_3 s_1 s_4 s_0 s_1$ be a reduced expression for $w \in \text{FC}(\tilde{C}_6)$ as seen in Example 1.3.1. Figure 1.6 shows a possible lattice point representation for $H(w)$. Since w is FC this is the unique heap representation for w . Because $H(w)$ is the unique heap we can obtain $\mathcal{L}(w)$ (respectively, $\mathcal{R}(w)$) from the blocks that are fully exposed to the top (respectively, bottom) of the heap. We see that $\mathcal{L}(w) = \{s_2, s_4, s_6\}$ and $\mathcal{R}(w) = \{s_1, s_4\}$.

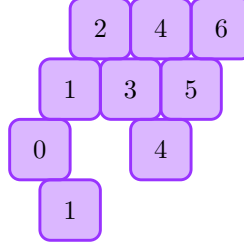


Figure 1.6: A lattice point representation for the heap of an FC element in $W(\tilde{C}_6)$.

Example 1.3.3. Let $w_1 = s_0 s_2 s_4 s_3 s_2 s_1$ be a reduced expression for $w \in W(\tilde{C}_4)$. Applying the commutation move $s_2 s_4 \mapsto s_4 s_2$, we can obtain another reduced expression for w , namely $w_2 = s_0 s_4 s_2 s_3 s_2 s_1$, which is in the same commutation class as w_1 , and hence has the same heap. However, applying the braid move $s_2 s_3 s_2 \mapsto s_3 s_2 s_3$, we obtain another reduced expression $w_3 = s_0 s_4 s_3 s_2 s_3 s_1$. Note that since w_3 was obtained by applying a braid move, w_3 is in a different commutation class than w_1 and w_2 . Representations of $H(w_1)$, $H(w_2)$, and $H(w_3)$ are seen in Figure 1.7, where the braid relation is colored in teal. Notice that from the heaps we see that $\mathcal{L}(w) = \{0, 2, 4\}$ and $\mathcal{R} = \{s_1, s_3\}$. However, if we only had one heap or the other, we would miss some elements in the left and right descent sets as s_3 is not fully exposed to the bottom of the heap in Figure ?? and s_2 is not fully exposed to the top of the heap in 1.7(b).

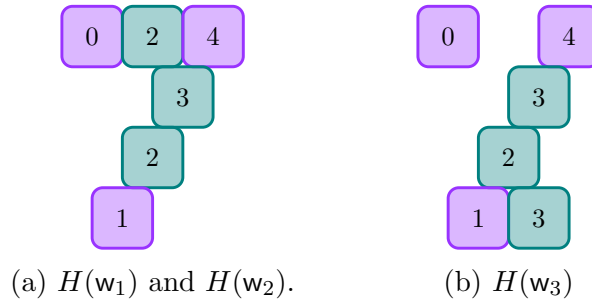


Figure 1.7: Two heaps of a non-FC element in $W(\tilde{C}_4)$.

It will be extremely useful for us to be able to quickly determine whether a heap corresponds to an element in $\text{FC}(B_n)$ or $\text{FC}(\tilde{C}_n)$. The next proposition is a special case of [13, Proposition 3.3] and follows quickly when one considers the consecutive subwords that are impermissible in reduced expressions for elements in $\text{FC}(B_n)$ and $\text{FC}(\tilde{C}_n)$ as discussed in Section 1.2.

Proposition 1.3.4. If $w \in \text{FC}(\tilde{C}_n)$, then $H(w)$ cannot contain any of the configurations seen in Figure 1.8, where $0 < k < n - 1$ and we use a square with a dotted boundary to emphasize that no element of the heap may occupy the corresponding position. \square

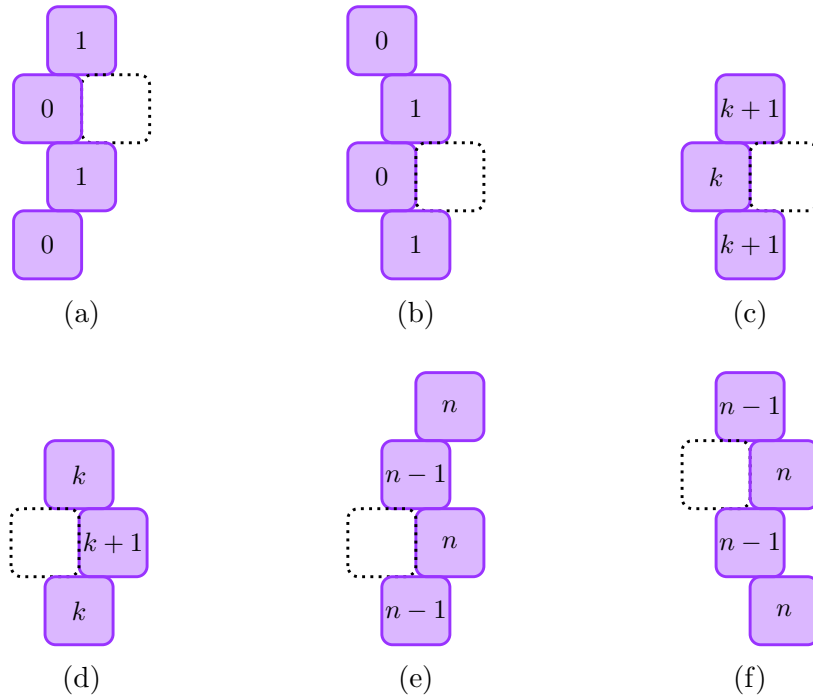


Figure 1.8: Impermissible configurations for heaps of $\text{FC}(\tilde{C}_n)$.

Since $W(B_n)$ is a parabolic subgroup of $W(\tilde{C}_n)$, we can use Figure 1.8 to classify the impermissible configurations for elements of $\text{FC}(B_n)$. In particular, the impermissible configurations for elements of $\text{FC}(B_n)$ are those seen in Figures 1.8(a), 1.8(b), 1.8(c), and 1.8(d).

Chapter 2

Star Reductions and Property T

2.1 Star Reductions

The notion of a star operation was originally introduced by Kazhdan and Lusztig in [11] for simply-laced Coxeter systems (i.e., $m(s, t) \leq 3$ for all $s, t \in S$), and was later generalized to all Coxeter systems in [12]. If $I = \{s, t\}$ is a pair of non-commuting generators of a Coxeter group W , then I induces four partially defined maps from W to itself, known as *star operations*. A star operation, when it is defined, increases or decreases the length of an element to which it is applied by 1. For our purposes it is enough to only define the star operations that decrease the length of an element by 1, and as a result we will not develop the notion in full generality.

Let (W, S) be a Coxeter system of type Γ and let $I = \{s, t\} \subseteq S$ be a pair of generators with $m(s, t) \geq 3$. Let $w \in W(\Gamma)$ such that $s \in \mathcal{L}(w)$. We define w to be *left star reducible by s with respect to t* if there exists $t \in \mathcal{L}(sw)$. We analogously define w to be *right star reducible by s with respect to t* . Observe that w is left (respectively, right) star reducible if and only if $w = stu$ (respectively, $w = uts$), where the product on the right hand side of the equation is reduced and $m(s, t) \geq 3$. We say that w is *star reducible* if it is either left or right star reducible.

Example 2.1.1. Let $w = s_0s_1s_0s_2$ be a reduced expression for $w \in W(B_4)$. We see that w is left star reducible by s_0 with respect to s_1 to $s_1s_0s_2$ since $m(s_0, s_1) = 4$ and $s_0 \in \mathcal{L}(w)$ while $s_1 \in \mathcal{L}(s_0w)$. Notice that w is FC and $\mathcal{R}(w) = \{s_2, s_0\}$. We see that $ws_2 = s_0s_1s_0$ and $ws_0 = s_0s_1s_2$. Note that in both instances $s_1 \notin \mathcal{R}(ws_2) = \{s_0\}$ and $s_1 \notin \mathcal{L}(ws_0) = \{s_2\}$. Because of this w is not right star reducible.

It may be helpful to visualize star reductions in terms of heaps. Let (W, S) be a Coxeter system with straightline Coxeter graph Γ and let $I = \{s, t\} \subseteq S$ be a pair of generators with $m(s, t) \geq 3$. Suppose w is left star reducible by s with respect to t . Then there exists a heap for w where the block for s is fully exposed to the top

such that removing the block for s off of the top allows for t to now be fully exposed to the top of the heap. Similarly if w is right star reducible by s with respect to t , then there exists a heap for w where the block for s is fully exposed to the bottom of the heap such that removing the block for s off of the bottom allows for t to now be fully exposed to the bottom. Conversely, if a heap for $w \in W(\Gamma)$ has this property, then w is star reducible. In Figure 2.1 we see the top portion of two possible heap representations of an element that is left star reducible by s with respect to t , where the dotted square represents no block may occupy this space. Notice that flipping the heap upside down in Figure 2.1 will result in a heap that is right star reducible. It is important to note that if the group element we are evaluating for star reducibility is not FC, then we must consider all heap representations for the element before concluding that an element is not star reducible.



Figure 2.1: A visual representation of an element that is left star reducible.

The following example utilizes heaps to show that an element is star reducible.

Example 2.1.2. Let $w = s_0 s_1 s_0 s_2$ be a reduced expression for $w \in W(B_4)$. Note that w is FC. By Example 2.1.1 we know that w is left star reducible by s_0 with respect to s_1 . In Figure 2.2(a), we see the heap for w . Notice that the block for s_0 is fully exposed to the top of the heap. Removing the block for s_0 gives the heap in Figure 2.2(b). Notice that the block for s_1 is now fully exposed to the top of the heap. However, notice that the blocks for s_0 and s_2 are fully exposed to the bottom. In removing either of these we are unable to fully expose s_1 to the bottom. Thus we can see that w is not right star reducible.

Notice that if w is not FC, then we are not able to say that w is not star reducible when viewing a single heap as there could be a different heap for w in which we are able to fully expose an element that was previously blocked.

Example 2.1.3. Let $w_1 = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(\tilde{C}_3)$. The heap for w is given in Figure 2.3(a), where we have highlighted a braid in teal. Notice that this heap appears to not be star reducible since if we were to remove the block for s_1 or s_3 we would still not fully expose s_2 to the top of the heap. The same goes for fully exposing blocks in the bottom of the heap. However, when we



Figure 2.2: Visualization of Example 2.1.1.

perform the braid move resulting in the heap seen in Figure 2.3(b) it is now obvious that the element is star reducible. Thus when considering a non-FC element for star reducibility via the heap, it is very important to consider all heaps for that element.

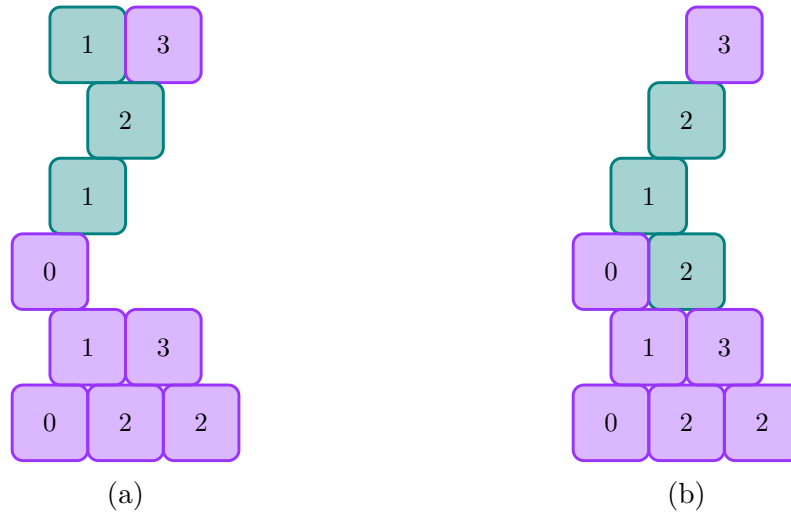


Figure 2.3: Visualization of Example 2.1.3

Using the notion of star reduction we are now able to introduce the concept of a star reducible Coxeter group. Let (W, S) be a Coxeter group of type Γ . We say that $W(\Gamma)$ is *star reducible* if every element of $FC(\Gamma)$ is star reducible to a product of commuting generators. That is, $W(\Gamma)$ is star reducible if when we apply star reductions on the left or right repeatedly to $w \in FC(\Gamma)$, eventually we obtain a product of commuting generators. Notice that we are restricting to just the FC elements in $W(\Gamma)$. Visually a star reducible Coxeter group can be thought of in the following way. Given a heap in $FC(\Gamma)$, we are able to systematically remove fully exposed blocks from the top or bottom of the heap and have a block that was

previously not fully exposed become fully exposed until we are left with a heap that can be drawn as a single row.

In [9], Green classified all star reducible Coxeter groups.

Proposition 2.1.4 (Green, [9]). Let (W, S) be a Coxeter system of type Γ . Then (W, S) is star reducible if and only if each component of Γ is either a complete graph with labels $m(s, t) \geq 3$, or is one of the following types: type A_n ($n \geq 1$), type B_n ($n \geq 2$), type D_n ($n \geq 4$), type F_n ($n \geq 4$), type H_n ($n \geq 2$), type $I_2(m)$ ($m \geq 3$), type \tilde{A}_n ($n \geq 3$ and n even), type \tilde{C}_n ($n \geq 3$ and n odd), type \tilde{E}_6 , or type \tilde{F}_5 . \square

2.2 Property T

In [9], Green utilizes the following theorem to help classify the star reducible Coxeter groups.

Proposition 2.2.1 (Green, [9]). Let (W, S) be a star reducible Coxeter system of type Γ , and let $w \in W$. Then one of the following possibilities occurs for some Coxeter generators s, t, u with $m(s, t) \neq 2$, $m(t, u) \neq 2$, and $m(s, u) = 2$:

- (1) w is a product of commuting generators;
- (2) w has a reduced product $w = stu$;
- (3) w has a reduced product $w = uts$;
- (4) w has a reduced product $w = svtu$. \square

Notice that Items (2) and (3) indicate an element that is left or right star reducible, respectively. Also notice that an element w that has the form of Item (1) does not meet the conditions of Items (2) and (3). In particular, w is not star reducible if Condition (1). Lastly, notice that if an element w is of the form of Item (4) and not of the form of Items (2)–(3), then w is not star reducible. Notice Items (2), (3), and (4) are not mutually exclusive if w is .

Motivated by Items (1) and (4) above, we define the notions of Property T and T-avoiding. Let (W, S) be a Coxeter system of type Γ and let $w \in W$. We say that w has *Property T* if and only if there exists a reduced product for w such that $w = stu$ or $w = uts$ where $m(s, t) \geq 3$. That is, w has Property T if there exists a reduced expression for w that begins or ends with a product of non-commuting generators. An element $w \in W(\Gamma)$ is called *T-avoiding* if w does not have Property T. This implies that a T-avoiding element is not star reducible.

Since elements that are star reducible also have Property T we already know how to visualize Property T in terms of heaps. It is clear that a product of commuting generators is T-avoiding, which we state as a proposition.

Proposition 2.2.2. Let (W, S) be a Coxeter system of type Γ . If $w \in W(\Gamma)$ such that w is a product of commuting generators, then w is T-avoiding. \square

Visually a product of commuting generators is a one row heap, it is clear a one row heap will not portray the characteristic of Property T as seen in Figure 2.1.

We will call an element that is a product of commuting generators *trivially T-avoiding*. If w is T-avoiding and not a product of commuting generators, we will say that w is *non-trivially T-avoiding*. It is not clear that such elements exist. Referring back to Green's classification (Proposition 2.2.1) of what elements in star reducible Coxeter groups look like, we see that Item (1) corresponds to an element w being trivially T-avoiding, Items (2) and (3) refer to the element w having Property T on the left and right, respectively and Item (4) refers to an element being non-trivially T-avoiding provided no reduced expression for the element exhibits Items (2) and (3). In star reducible Coxeter systems, every FC element is star reducible to a product of commuting generators, which implies that no FC element can be non-trivially T-avoiding in such groups. For example, as will be seen in the following sections, the Coxeter systems of type A_n and B_n have no non-trivial T-avoiding elements, while the Coxeter systems of type D_n do have non-trivially T-avoiding elements.

Example 2.2.3. Let $w \in W(A_5)$ with reduced expression $w = s_1 s_3 s_5$. It turns out that since w is a product of commuting generators, by Proposition 2.2.2 we know that w is trivially T-avoiding.

Example 2.2.4. Let $w \in W(A_5)$ with reduced expression $w_1 = s_5 s_3 s_2 s_4 s_1$. At first glance it may appear that w does not have Property T since both s_1 and s_4 commute as well as s_3 and s_5 . However, note that applying the commutation move $s_4 s_2 \mapsto s_2 s_4$ results in $w_2 = s_1 s_2 s_4 s_3 s_5$. Hence w has Property T since $m(s_1, s_2) = 3$ and there is a reduced expression for w that begins with $s_1 s_2$. In Figure 2.4 we see the heap for w . Note that we can see Property T in the bottom of the heap highlighted in orange.

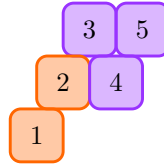


Figure 2.4: Heap of an element with Property T.

Example 2.2.5. Let $w \in W(\tilde{C}_4)$ with reduced expression $w = s_0 s_2 s_4 s_1 s_3 s_0 s_2 s_4$. It turns out that w is FC and non-trivially T-avoiding. The heap for w is seen in Figure 2.5. Notice that no matter which block we remove that is fully exposed to

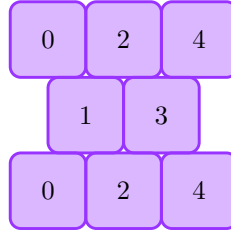


Figure 2.5: Heap of a non-trivially T-avoiding element in $W(\tilde{C}_4)$.

the top of the heap no new element becomes fully exposed. The same applies to the bottom of the heap. Thus, w is non-trivially T-avoiding.

One thing to notice here is that all Coxeter groups have trivially T-avoiding elements as they all contain products of commuting generators. The more interesting non-trivially T-avoiding elements do not appear in all Coxeter groups. In Chapter 3 we will summarize what is known about the T-avoiding elements in Coxeter systems of types \tilde{A}_n , A_n , D_n , F_n , and $I_2(m)$, and in Chapters 4 and 5 we classify the T-avoiding elements in Coxeter systems of types B_n and \tilde{C}_n .

2.3 Non-Cancellable Elements

We now introduce the concept of weak star reducible, which is related to the notion of cancellable in [5]. Let (W, S) be a Coxeter system of type Γ and let $I = \{s, t\} \subseteq S$ be a pair of noncommuting generators. If $w \in \text{FC}(\Gamma)$, then w is *left weak star reducible by s with respect to t to sw* if

- (1) w is left star reducible by s with respect to t , and
- (2) $tw \notin \text{FC}(\Gamma)$.

Notice that Condition (2) implies that $l(tw) > l(w)$. Also note that we are restricting out definition of weak star reducible to the set of FC elements of $W(\Gamma)$. We analogously define *right weak star reducible by s with respect to t to ws* . We say that w is *weak star reducible* if w is either left or right weak star reducible. Otherwise, we say that w is *non-cancellable* or *weak star irreducible*. Notice that from this we know that weak star reducible implies star reducible. However, star reducible does not imply weak star reducible.

Example 2.3.1. Let $w = s_0s_1s_0s_2$ be a reduced expression for $w \in W(B_4)$ as in Example 2.1.1. From Example 2.1.1 we know that w is left star reducible. However, $tw = s_1s_0s_1s_0s_2$, which is not in $\text{FC}(B_4)$. Thus, we see that w is left weak star

reducible by s_0 with respect to s_1 to $s_1s_0s_2$. In addition, Example 2.1.1 showed that w is not right star reducible and hence w is not right weak star reducible. However, since w is left weak star reducible we know that w is not non-cancellable.

Again it might be useful to visualize the concept of weak star reducible in terms of heaps. Recall that in Section 2.1 we described what a star reduction looks like in terms of heap. Since the definition of weak star reducible includes that a heap is star reducible we again need to have those properties. In addition, for a heap to be weak star reducible, adding the block that becomes fully exposed when a block is removed from the heap must create a braid in the heap forcing the new larger heap to not be FC. That is, one of the impermissible configurations seen in Section 1.3 will appear at the top or bottom of the heap.

Example 2.3.2. Let $\mathbf{w} = s_0s_1s_0s_2$ be a reduced expression for $w \in W(B_4)$ as in Example 2.3.1. Recall in Figure 2.2 the heap for w was seen along with what it star reduced to. In Figure 2.6 we see that adding s_1 to the top of the heap creates a braid which is highlighted in orange. Therefore, w is left star reducible by s_0 with respect to s_1 , to $\mathbf{w} = s_1s_0s_2$.

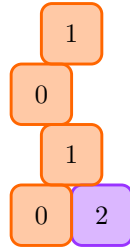
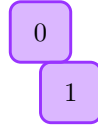
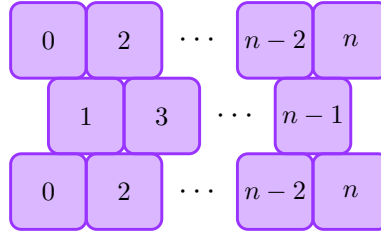


Figure 2.6: Heap of a weak star reducible element of $\text{FC}(B_4)$.

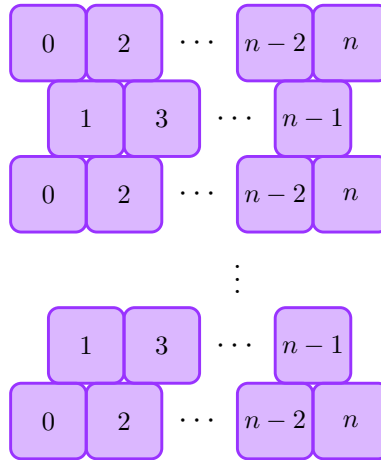
Example 2.3.3. Let $w \in \text{FC}(B_4)$ and let $\mathbf{w} = s_0s_1$ be a reduced expression for w . Note that w is left (respectively, right) star reducible by s_0 with respect to s_1 (respectively, by s_1 with respect to s_0). However, $s_1s_0s_1 \in \text{FC}(B_4)$ (respectively, $s_0s_1s_0 \in \text{FC}(B_4)$). Thus w is non-cancellable. Visually the heap appears in Figure 2.7. Clearly when s_0 is added to the bottom of the heap, the new heap is still in $\text{FC}(B_4)$ and the same can be said when s_1 is added to the top of the heap.

In [3], Ernst classified the non-cancellable elements in Coxeter groups of type $W(B_n)$ and $W(\tilde{C}_n)$. We will state part of the classification here as it is important to the development of the non-trivial T-avoiding elements in $W(\tilde{C}_n)$. To see the full classification see [3, Sections 4.2 and 5].

Figure 2.7: Heap of a non-cancellable element of $\text{FC}(B_4)$.Figure 2.8: Heap of a single sandwich stack in $W(\tilde{C}_n)$.

Before we state the classification we first define a specific group element in $W(\tilde{C}_n)$ which we will refer to as a single *sandwich stack*, which is seen in Figure 2.8.

We can stack single sandwich stacks together and get a sandwich stack. This is done by removing the top most layer of the heap and adding a new single sandwich stack to the stack. A sandwich stack is seen in Figure 2.9. These heaps are referenced in the classification of non-trivial T-avoiding elements in $W(\tilde{C}_n)$.

Figure 2.9: Heap of a stack of sandwich stacks in $W(\tilde{C}_n)$.

In [3], Ernst completely classified the non-cancellable elements in Coxeter systems of types B_n and \tilde{C}_n . In the type \tilde{C}_n case, the sandwich stacks are the only T-avoiding elements with full support. There are two other types of non-cancellable elements

that were classified in [3]. The first does not have full support which is important to our later classification. The second clearly has Property-T on the top.

Chapter 3

T-Avoiding Elements in Types $\tilde{A}_n, A_n, D_n, F_n$, and $I_2(m)$

In this Chapter we classify the non-trivial T-avoiding elements of Coxeter systems in types $\tilde{A}_n, A_n, D_n, F_n$ and $I_2(m)$.

3.1 Types \tilde{A}_n and A_n

We start with T-avoiding elements in Coxeter systems of type \tilde{A}_n and A_n . We first focus on non-trivial T-avoiding elements in $W(\tilde{A}_n)$.

Proposition 3.1.1. If $n \geq 2$ and n is odd, then there are no non-trivial T-avoiding elements in $W(\tilde{A}_n)$. Otherwise, if $n \geq 2$ and n is even then $W(\tilde{A}_n)$ contains non-trivial T-avoiding elements.

Proof. This is [6, Proposition 3.1.2] after a translation of terminology. \square

The classification seen in [6] did not specifically classify the non-trivial T-avoiding elements for type \tilde{A}_n for n odd. Since $W(\tilde{A}_n)$ for n odd is not star reducible, the non-trivial T-avoiding elements could be FC. The following is our conjecture regarding what the non-trivial T-avoiding elements are in $W(\tilde{A}_n)$ for n odd.

Conjecture 3.1.2. The only non-trivial T-avoiding elements in $W(\tilde{A}_n)$ for n odd are of the form $w = (s_0 s_2 \cdots s_{n-2} s_n s_1 s_3 \cdots s_{n-3} s_{n-1})^k$ for $k \in \mathbb{Z}^+$.

Recall that $W(\tilde{A}_n)$, for n even, is not a star reducible Coxeter group. Hence it makes sense that the T-avoiding elements in $W(\tilde{A}_n)$, for n even, can be FC. Further, as $W(A_n)$ is a parabolic subgroup of $W(\tilde{A}_n)$ and $W(A_n)$ is a star reducible Coxeter group, any FC non-trivial T-avoiding elements must have full support. First notice, that $w = (s_0 s_2 \cdots s_{n-2} s_n s_1 s_3 \cdots s_{n-3} s_{n-1})^k$ is reduced, FC, and has full support. In

addition, w is in fact T-avoiding. As stated in the conjecture we believe that these are the only non-trivial T-avoiding elements. However, it is not immediately obvious that there are not any non-FC non-trivial T-avoiding elements. Classifying these non-trivial T-avoiding elements remains an open problem. We now proceed with the classification of T-avoiding elements in Coxeter groups of type A_n .

Corollary 3.1.3. There are no non-trivially T-avoiding elements in $W(A_n)$.

Proof. Notice that the Coxeter graph of type A_n can be obtained from the Coxeter graph of type \tilde{A}_k , for $k > n$. This is done by removing the appropriate number of vertices and edges from the Coxeter graph of type \tilde{A}_k . Since $W(\tilde{A}_k)$ for k even has no non-trivial T-avoiding elements this forces $W(A_n)$ to not have non-trivial T-avoiding elements. Thus $W(A_n)$ does not have any non-trivial T-avoiding elements. \square

3.2 Type D_n

In this section summarize the classification of the T-avoiding elements in Coxeter systems of type D_n . Recall that $W(D_n)$ is a star reducible Coxeter group and as a result any potential non-trivial T-avoiding elements are not FC.

Proposition 3.2.1. There are non-trivial T-avoiding elements in $W(D_n)$ for $n \geq 4$.

Proof. This is a consequence of [8, Section 2.2]. \square

We now will classify these elements as seen in [8]. Before we do so we state interval notation useful to the classification from [8, Definition 2.3.1]. For $2 \leq i \leq j$ denote the element $s_i s_{i+1} \cdots s_{j-1} s_j$ by $[i, j]$. For $i \geq 3$, denote $s_1 s_3 s_4 \cdots s_i$ by $[1, i]$ and for $j \geq 2$ denote $s_1 s_2 s_3 \cdots s_j$ by $[0, j]$. If $0 \leq j < i$ and $i \geq 2$ define $[j, i] = [i, j]^{-1}$. Finally, for $i \leq -3$ and $j \geq 3$ denote $s_1 s_{i-1} s_{i-2} \cdots s_4 s_3 s_2 s_3 s_4 \cdots s_j$ by $[-i, j]$. The following determines the classification for T-avoiding elements in $W(D_n)$.

Proposition 3.2.2. Let $w \in W(B_n)$ be non-trivially T-avoiding. Then $w = w_n u$ reduced for some $m \leq n$, where u is a product of commuting generators such that $\text{supp}(u) \subseteq s_{m+2}, s_{m+3}, s_{m+4}, \dots, s_n$ and

$$w_n = \begin{cases} [2, 0][4, 0] \cdots [n-2, 0][n, 0][n-k, n-2k] \cdots [n-1, n-2][n, n] \\ [2, 0][4, 0] \cdots [m-2, 0][m, 0][m-k, m-2k] \cdots [m-1, m-2][m, m] \end{cases}$$

where the first case above corresponds to n even, the second case corresponds to n odd, $m = n - 1$ and

$$k = \begin{cases} \frac{n}{2} - 2 & \text{if, } n \text{ is even} \\ \frac{n-1}{2} - 2 & \text{if, } n \text{ is odd.} \end{cases}$$

Proof. This is [8, Lemmas 2.2.18 and 2.3.4]. Although it is not immediately obvious, w_n is reduced and not FC. \square

In Figure 3.1, we see two different elements that are T-avoiding in $W(D_5)$. Notice that the blocks that are highlighted in red alternate, this prevents the teal highlighted braid from forcing its way to the top or the bottom of the heap. Due to the fork in the graph we must make slight alterations to heaps for $W(D_n)$. Specifically we allow s_0 and s_1 to occupy the same horizontal placement.

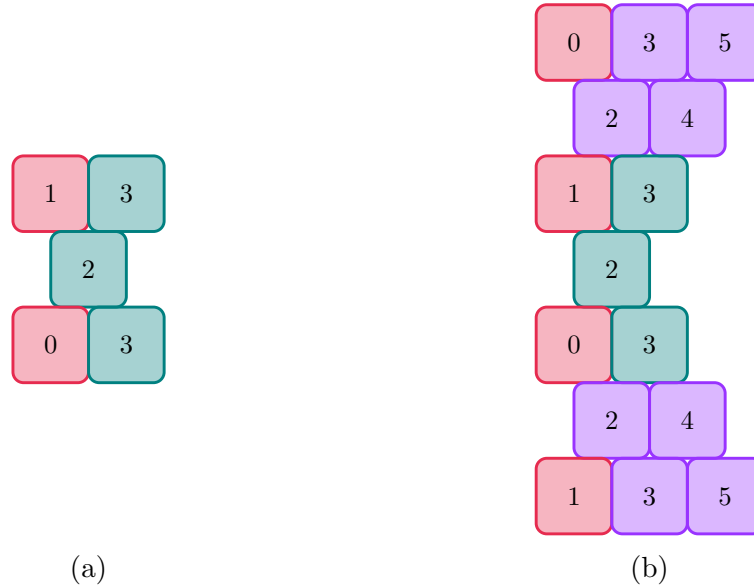


Figure 3.1: Visual representation of non-trivial T-avoiding elements in $W(D_5)$.

3.3 Type F_n

In this section we classify what is known regarding the T-avoiding elements in the Coxeter systems of type F_n for $n \geq 4$. Note that all of the following results are unpublished.

We start with the Coxeter system of type F_5 . Recall that $W(F_5)$ is a star reducible Coxeter group so any non-trivial T-avoiding elements will not be FC. Before we begin the classification we introduce the notion of a specific element in $W(F_5)$ called a *bowtie*, which is given by the heap in Figure 3.2. Note that in Figure 3.2(a), the orange blocks correspond to the elements that have bond strength 4. It turns out that the expression determined by this heap is in fact reduced. Looking at the heap in Figure 3.2(b), we have highlighted a braid in teal. We can obtain a “stack of bowties”

by removing the top most layer of the given heap for the bowtie and adding a new single bowtie to the stack. Doing this repeatedly results in the heap seen in Figure 3.3. Similar to a single bowtie, the expression that corresponds to a stack of bowties is reduced and not FC. These heaps are referenced in the following unpublished theorem by Cross, Ernst, Hills-Kimball, and Quaranta in 2012, which classifies the T-avoiding elements in the Coxeter systems of type F_5 .

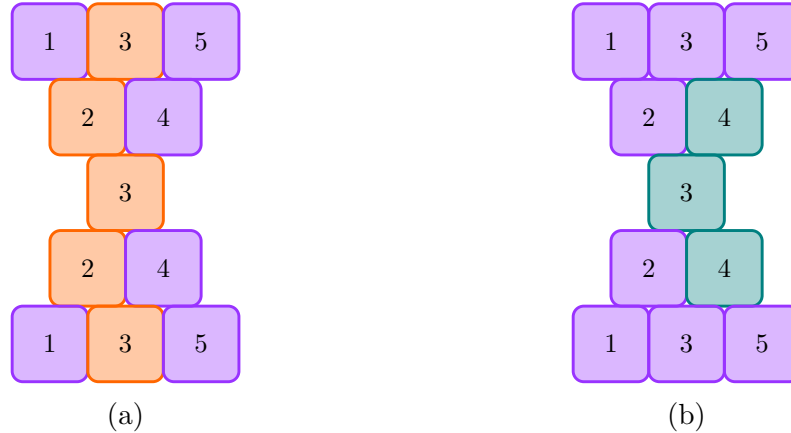


Figure 3.2: Heap of a single bowtie in $W(F_5)$.

Proposition 3.3.1. The only non-trivial T-avoiding elements in $W(F_5)$ are stacks of bowties. \square

As a result of the classification in type F_5 , Cross et al. were also able to classify the T-avoiding elements in $W(F_4)$.

Corollary 3.3.2. There are no non-trivial T-avoiding elements in the Coxeter system of type F_4 . \square

Proof. Since there are no non-trivial T-avoiding elements in $W(F_5)$ that do not have full support, we know that there are not any non-trivial T-avoiding elements in $W(F_4)$. Because if there were non-trivial T-avoiding elements they would also be non-trivially T-avoiding in $W(F_5)$. \square

Cross et al. conjectured that in Coxeter systems of type F_n for $n \geq 5$, an element is non-trivially T-avoiding if and only if it is a stack of bowties multiplied by a product of commuting generators. In 2013, Gilbertson and Ernst worked with this conjecture and quickly found it to be false. The heap seen in Figure 3.4 corresponds to a non-trivial T-avoiding element in Type F_6 that is not a bowtie. It turns out that like the bowties discussed above these elements can also be stacked to create an infinite

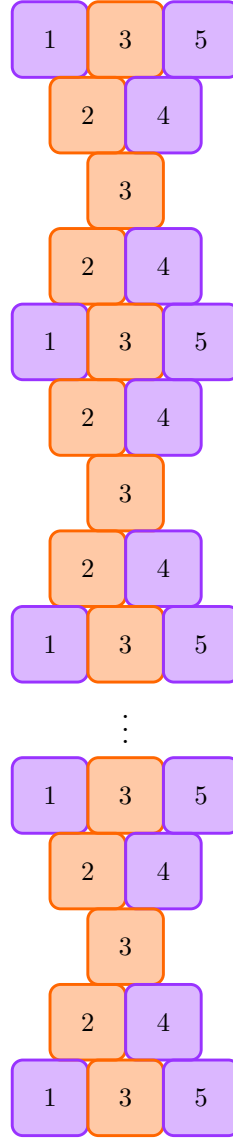


Figure 3.3: Heap of a stack of bowties in $W(F_5)$.

number of non-trivial T-avoiding elements. In addition, as n gets large there are a number of modifications that can be made that result in additional non-trivially T-avoiding elements. From this we conjecture that the classification of T-avoiding elements in Coxeter systems of type F_n for $n \geq 6$ gets complicated very quickly. Classifying T-avoiding elements in $W(F_n)$ for $n \geq 6$ remains an open problem.

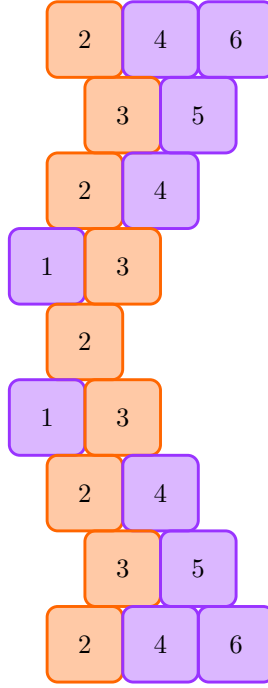


Figure 3.4: Heap of a non-trivial T-avoiding element in $W(F_6)$

3.4 Type $I_2(m)$

We next classify the T-avoiding elements in Coxeter systems of type $I_2(m)$. Note that in Coxeter systems of type $I_2(m)$, the only products of commuting generators have length 1. Although the following is a quick result, we believe that it does not already appear in the literature.

Theorem 3.4.1. There are no non-trivial T-avoiding elements in Coxeter systems of type $I_2(m)$.

Proof. The graph for the Coxeter system of $I_2(m)$ appears in Figure 1.2(c). Note that the graph consists of two vertices, namely, s_1 and s_2 , and a single edge with weight m . Also, recall that $W(I_2(m))$ is a star reducible Coxeter group. This implies that any non-trivial T-avoiding elements in $W(I_2(m))$ must not be FC, as all of the FC elements have Property T or are trivially T-avoiding. The only non-FC element in $W(I_2(m))$ is the element of length m that has exactly two reduced expressions consisting of alternating products of s_1 and s_2 . Clearly, this element begins and ends with a product of noncommuting generators. Thus, this element has Property T. Hence $W(I_2(m))$ has no non-trivial T-avoiding elements. \square

Chapter 4

T-Avoiding Elements in Type B_n

In this section we classify the T-Avoiding elements in Coxeter systems of type B_n . We start by introducing some combinatorial tools for type B_n and then finish with a proof of the classification in type B_n . Note that the proof for Coxeter systems of type B_n closely follows the classification of T-avoiding elements of type D_n seen in [8].

4.1 Tools for the Classification

Recall from Example 1.1.1 that $W(B_n) \cong \text{Sym}_n^B$. This implies that we can represent each element $w \in W(B_n)$ as a signed permutation. That is we can write $w \in W(B_n)$ using one-line notation

$$w = [w(1), w(2), \dots, w(n-1), w(n)],$$

where we write a bar underneath a number in place of a negative sign in order to simplify notation. In particular, for each $s_i \in S(B_n)$, $i = 1, 2, \dots, n-1$, we have

$$s_i = [1, 2, \dots, i-1, i+1, \bar{i}, i+2, \dots, n-1, n]$$

and we identify $s_0 \in S(B_n)$ via

$$s_0 = [\bar{1}, 2, \dots, n].$$

Further $w(-i) = -w(i)$ for $|i| \in \{1, 2, \dots, n\}$. The following propositions provide insight into what happens to a given on-line notation when we multiply by s_i on the right or the left.

Proposition 4.1.1. Let $w \in W(B_n)$ with corresponding signed permutation

$$w = [w(1), w(2), \dots, w(n)].$$

Suppose $s_i \in S(B_n)$. If $i \geq 1$, then multiplying w on the right by s_i has the effect of interchanging $w(i)$ and $w(i+1)$ in the one-line notation. If $i = 0$, then multiplying w on the right by s_i has the effect of switching the sign of $w(1)$.

Proof. This follows from [2, Section 8.1 and A3.1]. \square

Proposition 4.1.2. Let $w \in W(B_n)$ with corresponding signed permutation

$$w = [w(1), w(2), \dots, w(n)].$$

Suppose $s_i \in S(B_n)$. If $i \geq 1$, then multiplying on the left by s_i has the effect of interchanging the entries whose absolute values are i and $i+1$ in the one-line notation. If $i = 0$, then multiplying w on the left by s_i has the effect of switching the sign of the entry whose absolute value is 1.

Proof. This follows from [2, Section 8.1 and A3.1]. \square

Suppose $w \in W(B_n)$ has reduced expression $\mathbf{w} = s_{x_1}s_{x_2} \cdots s_{x_n}$. We may construct the signed permutation of w from left to right as it is the easier way to multiply based upon the above propositions. We provide an example of this construction below.

Example 4.1.3. Let $w \in W(B_6)$ with a given reduced expression $\mathbf{w} = s_0s_1s_3s_4s_5s_2$. Then we iteratively build the signed permutation as follows. First, $s_0 = [-1, 2, 3, 4, 5, 6]$ by definition. Next $s_0s_1 = [2, -1, 3, 4, 5, 6]$ since multiplying by s_1 on the right hand side switches the values in position 1 and position 2. Repeating this we get $s_0s_1s_3 = [2, -1, 4, 3, 5, 6]$ and ultimately we end with $w = [2, 4, -1, 5, 6, 3]$.

Notice that if we were to construct the signed permutation for w from right to left, we would start with $s_2 = [1, 3, 2, 4, 5, 6]$. Next we would have $s_5s_2 = [1, 3, 2, 4, 6, 5]$. However, $s_4s_5s_2 = [1, 3, 2, 5, 6, 4]$. Notice this time we were not able to just switch $w(i)$ and $w(i+1)$ instead we found 4 and 5 and switched their relative positions, which is more difficult than constructing the one-line notation left to right, which is why we choose to construct left to right.

Given the one-line notation for an element $w \in W(B_n)$ we can easily calculate the left and right descent sets of w . The following proposition explains how.

Proposition 4.1.4. Let $w \in W(B_n)$. Then

$$\mathcal{R}(w) = \{s_i \in S \mid w(i) > w(i+1)\}$$

where $w(0)=0$ by definition.

Proof. This is [2, Proposition 8.1.2]. \square

We now will introduce the concept of signed pattern avoidance, which will help with the classification of the T-avoiding elements in Coxeter systems of type B_n . Our approach mimics the one found in [8]. Let $w \in W(B_n)$. We say that w *avoids the consecutive pattern abc* if there is no $i \in \{1, 2, \dots, n-2\}$ such that $(w(i), w(i+1), w(i+2))$ is in the same relative order as (a, b, c) , where typically one takes $\{a, b, c\} = \{1, 2, 3\}$. We say that w *avoids the signed consecutive pattern abc* if there is no $i \in \{1, 2, \dots, n-2\}$ such that $(|w(i)|, |w(i+1)|, |w(i+2)|)$ is in the same consecutive order as $(|a|, |b|, |c|)$ and such that $\text{sgn}(w(i)) = \text{sgn}(a)$, $\text{sgn}(w(i+1)) = \text{sgn}(b)$, and $\text{sgn}(w(i+2)) = \text{sgn}(c)$.

Example 4.1.5. Let $w \in W(B_4)$ with signed permutation

$$w = [\underline{2}, 4, \underline{1}, 3].$$

We see that w has the signed consecutive pattern $\underline{2}3\underline{1}$, since $(|w(1)|, |w(2)|, |w(3)|)$ are in the same relative order as $(|-2|, |3|, |-1|)$, and $\text{sgn}(w(1)) = \text{sgn}(-2)$, $\text{sgn}(w(2)) = \text{sgn}(3)$, and $\text{sgn}(w(3)) = \text{sgn}(-1)$. However, w avoids the signed consecutive pattern $\underline{1}2\underline{3}$.

Occasionally, we will need to factor a reduced expression w for $w \in W(B_n)$ in a specific manner. Let $I = \{s, t\}$ for $s, t \in S(B_n)$. Define w^I as the set of all $w \in W(B_n)$ such that $\mathcal{L}(w) \cap I = \emptyset$ and define $W_I = \langle s, t \rangle$. In [10], it is shown that any element $w \in W(B_n)$ can be written as $w = w_I w^I$ reduced where $w_I \in W_I$ and $w^I \in W^I$.

4.2 Classification of T-Avoiding Elements in Type B_n

Time permitting we will streamline the corollaries into the lemmas themselves or as a remark at the end of the lemmas

In this section we will classify the T-avoiding elements in Coxeter groups of type B_n . Our main result in this section is Theorem 4.2.17. First we need some preparatory lemmas.

Lemma 4.2.1. Let $s, t \in S(B_n)$ such that $m(s, t) = 3$. Then w has a reduced expression ending in sts if and only if w has the consecutive pattern 321.

Proof. Let $i \geq 1$, let $I = \{s_i, s_{i+1}\}$ and write $w = w^I w_I$. Note that since $m(s_i, s_{i+1}) = 3$, $s_0 \notin I$. Observe that if w has a reduced expression ending in the product of two non-commuting generators $s_i s_{i+1}$ or $s_{i+1} s_i$, then we have $w_I \in \{s_i s_{i+1}, s_{i+1} s_i\}$.

Suppose w has the consecutive pattern 321. Then there is some i such that $w(i) > w(i+1) > w(i+2)$. By Proposition 4.1.4, $s_i, s_{i+1} \in \mathcal{R}(w)$. Since $W(B_n)$ is a simply laced Coxeter group, w ends in $s_i s_{i+1} s_{i+1}$ or $s_{i+1} s_i s_{i+1}$.

Conversely, suppose w ends in $s_i s_{i+1} s_i$. This implies that either $w_I = s_i s_{i+1}$ or $w_I = s_{i+1} s_i$ which implies that $s_i, s_{i+1} \in \mathcal{R}(w)$. Since $s_i, s_{i+1} \in \mathcal{R}(w)$, we see that $w(i) > w(i+1) > w(i+2)$ by Proposition 4.1.4. Thus w has the consecutive pattern 321. Therefore, w has a reduced expression ending in sts if and only if w has the consecutive pattern 321. \square

Corollary 4.2.2. Let $s, t \in S(B_n)$ such that $m(s, t) = 3$. Then w has a reduced expression beginning with sts if and only if w^{-1} has the consecutive pattern 321.

Proof. Let $s, t \in S(B_n)$ such that $m(s, t) = 3$, and $s_0 \notin \{s, t\}$. We know that w has no reduced expressions beginning with sts if and only if w^{-1} has no reduced expression ending with sts which by Theorem 4.2.2 happens only if w^{-1} avoids the consecutive pattern 321. \square

Lemma 4.2.3. If $i \neq 0$, then w has a reduced expression ending in $s_i s_{i+1}$ if and only if w has the consecutive pattern 231.

Proof. Suppose that w has the consecutive pattern 231. Then there is some i such that $w(i+1) > w(i) > w(i+2)$. By Proposition 4.1.4, $s_{i+1} \in \mathcal{R}(w)$. Now multiplying on the right by s_{i+1} we see that $ws_{i+1}(i+1) = w(i+2)$ and $ws_{i+1}(i) = w(i)$. We know that $w(i+2) < w(i)$, which implies that $s_i \in \mathcal{R}(ws_{i+1})$, and hence w has a reduced expression that ends in $s_i s_{i+1}$.

Conversely, suppose that w has a reduced expression ending in $s_i s_{i+1}$. Then $w(i+2) < w(i+1)$ and $w(i) < w(i+1)$. Since $s_i \in \mathcal{R}(ws_{i+1})$ we have $w(i+2) = ws_{i+1}(i+1) < ws_{i+1}(i) = w(i)$. Thus we have that $w(i+1) > w(i) > w(i+2)$. Hence w has the consecutive pattern 231.

Therefore, w has a reduced expression ending in st if and only if w has the consecutive pattern 231. \square

Corollary 4.2.4. If $i \neq 0$, then w has a reduced expression beginning with $s_i s_{i+1}$ if and only if w^{-1} has the consecutive pattern 231.

Proof. Let $s, t \in S(B_n)$ such that $m(s, t) = 3$, and $s_0 \notin \{s, t\}$. We know that w has no reduced expressions beginning with st if and only if w^{-1} has no reduced expression ending with st which by Theorem 4.2.2 happens only if w^{-1} avoids the consecutive pattern 231. \square

Lemma 4.2.5. If $i \neq 0$, then w has a reduced expression ending in $s_{i+1} s_i$ if and only if w has the consecutive pattern 312.

Proof. Suppose that w has the consecutive pattern 312. Then there is some i such that $w(i) > w(i+2) > w(i+1)$. By Proposition 4.1.4 we see that $s_i \in \mathcal{R}(w)$. Multiplying on the right by s_i we get $ws_i(i+1) = w(i)$ and $ws_i(i+2) = w(i+2)$.

By above $w(i) > w(i+2)$, and by Proposition 4.1.4 $s_{i+1} \in \mathcal{R}(ws_i)$. This implies that w has a reduced expression ending in $s_{i+1}s_i$.

Conversely suppose w ends in a reduced expression with $s_{i+1}s_i$. Then $w_I = s_{i+1}s_i$. We see that $w(i) > w(i+1)$ and $w(i+2) > w(i+1)$. Since $s_{i+1} \in \mathcal{R}(ws_i)$, we have $w(i+2) = ws_i(i+2) < ws_i(i+1) = w(i)$. From this we have $w(i) > w(i+2)$, so $w(i) > w(i+2) > w(i+1)$. Hence, w has the consecutive pattern 312.

Therefore, w has a reduced expression ending in ts if and only if w has the consecutive pattern 312. \square

Corollary 4.2.6. If $i \neq 0$, then w has a reduced expression beginning with $s_{i+1}s_i$ if and only if w^{-1} has the consecutive pattern 312.

Proof. Let $s, t \in S(B_n)$ such that $m(s, t) = 3$, and $s_0 \notin \{s, t\}$. We know that w has no reduced expressions beginning with ts if and only if w^{-1} has no reduced expression ending with ts which by Theorem 4.2.2 happens only if w^{-1} avoids the consecutive pattern 312. \square

Lemma 4.2.7. Let $w \in W(B_n)$. Then w has a reduced expression ending in s_1s_0 if and only if $w(0) > w(1)$ and $-w(1) > w(2)$.

Proof. Suppose $w \in W(B_n)$ such that w ends with s_1s_0 . Then $s_0 \in \mathcal{R}(w)$ and $s_1 \in \mathcal{R}(ws_0)$. This implies that $ws_0(1) > ws_0(2)$ by Proposition 4.1.4. We see that $ws_0(1) = w(-1) = -w(1)$ and $ws_0(2) = 2$. Hence $-w(1) = ws_0(1) > ws_0(2) = w(2)$. Further, since $s_0 \in \mathcal{R}(w)$, we see that $w(0) > w(1)$.

Conversely, suppose $w \in W(B_n)$ such that $w(0) > w(1)$ and $-w(1) > w(2)$. Since $w(0) > w(1)$ so $s_0 \in \mathcal{R}(w)$. Multiplying on the right by s_0 we see that $ws_0(1) = -w(1)$ and $ws_0(2) = w(2)$. Note that since $ws_0(1) = -w(1) > w(2) = ws_0(2)$, $s_1 \in \mathcal{R}(ws_0)$. Thus w ends with s_1s_0 .

Therefore, w has a reduced expression ending in s_1s_0 if and only if $w(0) > w(1)$ and $-w(1) > w(2)$. \square

Corollary 4.2.8. Let $w \in W(B_n)$. Then w has a reduced expression beginning in s_0s_1 if and only if $w^{-1}(0) > w^{-1}(1)$ and $-w^{-1}(1) > w^{-1}(2)$.

Proof. Let $w \in W(B_n)$. We know that w has no reduced expressions beginning in s_0s_1 if and only if w^{-1} has no reduced expressions ending in s_0s_1 . By Lemma 4.2.7 we know that this occurs if and only if $w^{-1}(0) > w^{-1}(1)$ and $-w^{-1}(1) > w^{-1}(2)$. \square

Lemma 4.2.9. Let $w \in W(B_n)$. Then w has a reduced expression ending in s_0s_1 if and only if $w(0) > w(2)$ and $w(1) > w(2)$.

Proof. Suppose $w \in W(B_n)$ such that w ends with s_0s_1 . Then $s_1 \in \mathcal{R}(w)$ and $s_0 \in \mathcal{R}(ws_1)$. Then $ws_1(0) > ws_1(1)$. We see that $ws_1(0) = 0$ and $ws_1(1) = w(2)$. This implies that $0 = ws_1(0) > ws_1(1) = 2$. Further, since $s_1 \in \mathcal{R}(w)$ this implies that $w(1) > w(2)$. Thus if w ends with s_0s_1 , then $w(1) > w(2)$ and $w(0) > w(2)$.

Conversely, suppose $w \in W(B_n)$ such that $w(1) > w(2)$ and $w(0) > w(2)$. This implies that $s_1 \in \mathcal{R}(w)$. Multiplying w on the right by s_1 we see that $ws_1(0) = w(0)$ and $ws_1(1) = w(2)$. Note that since $ws_1(0) = w(0) > w(2) = ws_1(1)$, $s_0 \in \mathcal{R}(ws_1)$. Thus w ends with s_0s_1 .

Therefore, w has a reduced expression ending in s_0s_1 if and only if $w(1) > w(2)$ and $w(0) > w(2)$. \square

Corollary 4.2.10. Let $w \in W(B_n)$. Then w has a reduced expression beginning in s_1s_0 if and only if $w^{-1}(0) > w^{-1}(2)$ and $w^{-1}(1) > w^{-1}(2)$.

Proof. Let $w \in W(B_n)$. We know that w has no reduced expressions beginning in s_1s_0 if and only if w^{-1} has no reduced expressions ending in s_1s_0 . By Lemma 4.2.7 we know that this occurs if and only if $w^{-1}(0) > w^{-1}(2)$ and $w^{-1}(1) > w^{-1}(2)$. \square

Lemma 4.2.11. Let $w \in W(B_n)$ such that each entry for w in the one-line notation is positive and both w and w^{-1} avoid the consecutive patterns 321, 231, and 312, then w is a product of commuting generators.

Proof. This follows from an appropriate translation of [8, Lemma 2.2.9]. \square

Lemma 4.2.12. Let $w \in W(B_n)$ be trivially T-avoiding and let $i \in \{1, 2, \dots, n\}$. Then w satisfies all the following conditions:

- (1) $w(j) > \min\{w(i-1), w(i)\}$ for all $j > i$;
- (2) $w(k) < \max\{w(i-1), w(i)\}$ for all $k < i-1$;
- (3) If $w(i), w(i+1) > 0$, then $w(j) > 0$ for all $j \geq i$;
- (4) If $w(i), w(i+1) < 0$, then $w(j) < 0$ for all $j \leq i+1$.

Proof. Suppose there is some least $j > i$ such that $w(j) \leq \min\{w(i-1), w(i)\}$. Note that $j > i$ so $j \neq i$, and $j \neq i-1$ so $w(j) < \min\{w(i-1), w(i)\}$. Note that j is the least so $w(j-2) \geq \min\{w(i+1), w(i)\} > w(j)$. This implies that either $w(j-1) > w(j-2) > w(j)$ or $w(j-2) > w(j-1) > w(j)$, which implies w has the consecutive pattern 231 or 321, which is a contradiction to w being a non-trivial T-avoiding element by Lemmas 4.2.1 and 4.2.5. Thus proving (1).

Suppose there exists a maximal $k < i-1$ such that $w \geq \max\{w(i-1), w(i)\}$. Note that $k < i-1$ so $k \neq i$ and $k \neq i-1$. Then $w(k) > \max\{w(i-1), w(i)\}$. Since

k is maximal $w(k+1) \leq \max\{w(i-1), w(i)\}$ and $w(k+2) \leq \max\{w(i-1), w(i)\}$. This implies that either $w(k+2) < w(k+1) < w(k)$ or $w(k+1) < w(k+2) < w(k)$, which implies w has the consecutive pattern 321 or 312, which is a contradiction to w being a non-trivial T-avoiding element by Lemmas 4.2.1 and 4.2.3. Thus proving (2).

It is easy to see that assertion (1) implies (3) and assertion (2) implies (4). \square

Lemma 4.2.13. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{231}$. Then w has Property T.

Proof. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{231}$.

Case (1): Suppose w has the one-line notation $w = [\underline{2}, 3, 1]$. This implies that $w = s_1 s_0 s_2$. Clearly, w begins with a product of non-commuting generators. Thus w has Property T.

Case (2): Suppose that w has the one-line notation $w = [\underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $i = 4, 5, \dots, n$. We now consider the possible signed consecutive pattern $bc*$. The following are the possibilities: 312 , $31\underline{2}$, 321 , $32\underline{1}$, 213 , or $21\underline{3}$. We know that b and c must be positive since they are positive in w and we also know that $b > c$ by the original signed consecutive pattern. Note that by Lemmas 4.2.1, 4.2.3, and 4.2.7 all of these patterns imply that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (3): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $i = 1, 2, \dots, n-3$. We now consider the possible signed consecutive pattern $*\underline{ab}$. The following are the possibilities: 123 , $\underline{123}$, 213 , $\underline{213}$, 312 , or $\underline{312}$. Note that by Lemmas 4.2.3, 4.2.7, and 4.2.9 all of these patterns implies that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (4): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $|w(i)| \neq a, b, c$. In this case we can apply either Case (2) or Case (3) and we can conclude that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Therefore, if $w \in W(B_n)$ contains the consecutive pattern $\underline{231}$, then w has Property T. \square

Lemma 4.2.14. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{231}$. Then w has Property T.

Proof. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{231}$.

Case (1): Suppose w has the one-line notation $w = [\underline{2}, 3, \underline{1}]$. This implies that $w = s_0 s_1 s_0 s_2$. Clearly, w begins with a product of non-commuting generators. Thus w has Property T.

Case (2): Suppose that w has the one-line notation $w = [\underline{a}, b, \underline{c}, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $i = 4, 5, \dots, n$. We now consider the possible signed consecutive pattern $b\underline{c}*$. The following are the possibilities: $\underline{312}$, $\underline{312}$, $\underline{321}$, $\underline{321}$, $\underline{213}$, or $\underline{213}$. We know that b must be positive since it is positive in w , c must be negative since it is negative in w , and we also know that $|b| > |c|$ by the original signed consecutive pattern. Note that by Lemmas 4.2.1, 4.2.3, and 4.2.7 all of these patterns imply that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (3): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, \underline{c}]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $i = 1, 2, \dots, n - 3$. We now consider the possible signed consecutive pattern $*\underline{ab}$. The following are the possibilities: $\underline{123}$, $\underline{123}$, $\underline{213}$, $\underline{21}$, $\underline{312}$, or $\underline{312}$. We know that a must be negative, b must be positive and $|a| < |b|$ by the original signed permutation. Note that by Lemmas 4.2.3, 4.2.7, and 4.2.9 all of these patterns implies that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (4): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{231}$, and $*$ indicates unknown values for $w(i)$ for $|w(i)| \neq a, b, c$. In this case we can apply either Case (2) or Case (3) and we can conclude that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Therefore, if $w \in W(B_n)$ contains the consecutive pattern $\underline{231}$, then w has Property T. \square

Lemma 4.2.15. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{123}$. Then either w has Property T or is a trivial T-avoiding element.

Proof. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{123}$.

Case (1): Suppose w has the one-line notation $w = [\underline{123}]$. This implies that $w = s_0$. Clearly, w is a trivial T-avoiding element as it is a single generator.

Case (2): Suppose that w has the one-line notation $w = [\underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{123}$, and $*$ indicates unknown values for $w(i)$ for $i = 4, 5, \dots, n$. We now consider the possible signed consecutive patterns $b\underline{c}*$. The following are the possibilities: $\underline{231}$, $\underline{231}$, $\underline{132}$, $\underline{132}$, $\underline{123}$, $\underline{123}$. We know that b and c are positive, and we also know that $|b| < |c|$ by the original signed consecutive pattern. Note that by Lemmas 4.2.1, 4.2.3, and 4.2.7 all of these patterns imply that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (3): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{123}$, and $*$ indicates unknown values for $w(i)$ for $i = 1, 2, \dots, n - 3$. We now consider the possible signed consecutive patterns $*\underline{ab}$. The following are the possibilities: $\underline{312}$, $\underline{312}$, $\underline{213}$, $\underline{213}$, $\underline{123}$, or $\underline{1}, \underline{23}$.

We know that a must be negative, b must be positive and $|a| < |b|$ by the original signed permutation. Note that by Lemmas 4.2.3, 4.2.7, and 4.2.9 all of these patterns implies that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (4): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{123}$, and $*$ indicates unknown values for $w(i)$ for $|w(i)| \neq a, b, c$. In this case we can apply either Case (2) or Case (3) and we can conclude that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Therefore, if $w \in W(B_n)$ contains the consecutive pattern $\underline{123}$, then w has Property T or is a trivial T-avoiding element. \square

Lemma 4.2.16. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{132}$. Then either w has Property T or is a trivial T-avoiding element.

Proof. Let $w \in W(B_n)$ such that w has the consecutive pattern $\underline{132}$.

Case (1): Suppose w has the one-line notation $w = [\underline{132}]$. This implies that $w = s_0 s_2$. Clearly, w is a trivial T-avoiding element as it is a product of commuting generators.

Case (2): Suppose that w has the one-line notation $w = [\underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{132}$, and $*$ indicates unknown values for $w(i)$ for $i = 4, 5, \dots, n$. We now consider the possible signed consecutive pattern $bc*$. The following are the possibilities: 231 , $23\underline{1}$, 132 , $13\underline{2}$, 123 , or $12\underline{3}$. We know that b and c are positive, and we also know that $|b| < |c|$ by the original signed consecutive pattern. Note that by Lemmas 4.2.1, 4.2.3, and 4.2.7 all of these patterns imply that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (3): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{132}$, and $*$ indicates unknown values for $w(i)$ for $i = 1, 2, \dots, n-3$. We now consider the possible signed consecutive pattern $*\underline{ab}$. The following are the possibilities: $3\underline{12}$, $\underline{312}$, $2\underline{13}$, $\underline{213}$, $32\underline{1}$, or $\underline{321}$. We know that a must be negative, b must be positive and $|a| < |b|$ by the original signed permutation. Note that by Lemmas 4.2.3, 4.2.7, and 4.2.9 all of these patterns implies that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Case (4): Suppose that w has the one-line notation $w = [*, \dots, *, \underline{a}, b, c, *, \dots, *]$ where \underline{abc} corresponds to the signed consecutive pattern $\underline{132}$, and $*$ indicates unknown values for $w(i)$ for $w(i) \neq a, b, c$. In this case we can apply either Case (2) or Case (3) and we can conclude that w ends or begins with a product of noncommuting generators. Thus w has Property T.

Therefore, if $w \in W(B_n)$ contains the consecutive pattern $\underline{132}$, then w has Property T or is a trivial T-avoiding element. \square

We now are ready to tackle one of the main results of this thesis.

Theorem 4.2.17. There are no non-trivial T-avoiding elements in $W(B_n)$.

Proof. Suppose that $w \in W(B_n)$ is a non-trivial T-avoiding element. There are $2^3 \cdot 3! = 48$ possible choices of signed consecutive patterns for $w(1)w(2)w(3)$ where $w = [w(1), w(2), w(3), *, \dots, *]$.

123	<u>123</u>	<u>123</u>	<u>123</u>	<u>123</u>	<u>123</u>	<u>123</u>	<u>123</u>
132	<u>132</u>	<u>132</u>	<u>132</u>	<u>132</u>	<u>132</u>	<u>132</u>	<u>132</u>
213	<u>213</u>	<u>213</u>	<u>213</u>	<u>213</u>	<u>213</u>	<u>213</u>	<u>213</u>
231	<u>231</u>	<u>231</u>	<u>231</u>	<u>231</u>	<u>231</u>	<u>231</u>	<u>231</u>
312	<u>312</u>	<u>312</u>	<u>312</u>	<u>312</u>	<u>312</u>	<u>312</u>	<u>312</u>
321	<u>321</u>	<u>321</u>	<u>321</u>	<u>321</u>	<u>321</u>	<u>321</u>	<u>321</u>

We can use Lemma 4.2.1 and Corollary 4.2.2 to eliminate the signed consecutive patterns highlighted in **turquoise**. We can use Lemma 4.2.5 and Corollary 4.2.4 to eliminate the signed consecutive patterns highlighted in **red**. We can use Lemma 4.2.3 and Corollary 4.2.6 to eliminate the consecutive patterns highlighted in **green**. We can use Lemma 4.2.7 and Corollary 4.2.8 to eliminate the signed consecutive patterns highlighted in **yellow**. We can use Lemma 4.2.9 and Corollary 4.2.10 to eliminate signed consecutive patterns highlighted in **brown**. We can use Lemma 4.2.11 to eliminate the signed consecutive patterns highlighted in **blue**. We can use Lemmas 4.2.13 and 4.2.14 to eliminate signed consecutive patterns highlighted in **purple**. Finally we can use Lemmas 4.2.15 and 4.2.16 to eliminate signed consecutive patterns highlighted in **orange**. Since all of the above patterns are eliminated as possibilities for $w(1)w(2)w(3)$ and there are no other signed consecutive patterns that are possible for these positions, w is not be a non-trivial T-avoiding element in the Coxeter group of type B_n . \square

The upshot of Theorem 4.2.17 is that the only T-avoiding elements in Coxeter systems of type B_n are products of commuting generators.

Chapter 5

T-Avoiding Elements in Type \tilde{C}_n

5.1 Classification of T-Avoiding Elements in $W(\tilde{C}_n)$

In this section we will classify the T-avoiding elements in Coxeter systems of type \tilde{C}_n . Since $W(A_n)$ and $W(B_n)$ are parabolic subgroups of $W(\tilde{C}_n)$ and these groups have no non-trivial T-avoiding elements, any non-trivial T-avoiding elements of $W(\tilde{C}_n)$ must have full support. We will first show that there are no non-trivial T-avoiding elements that are not FC in $W(\tilde{C}_n)$.

Theorem 5.1.1. There are no non-trivial T-avoiding elements in $W(\tilde{C}_n)/\text{FC}(\tilde{C}_n)$.

Proof. We proceed by contradiction. Let $w \in W(\tilde{C}_n)/\text{FC}(\tilde{C}_n)$ such that w has full support and w is T-avoiding. Consider all possible heaps for w that have been constructed with the blocks in the lowest possible vertical position. That is, the heaps have been constructed in such a way that all blocks that are fully exposed to the bottom are in the same vertical position and all blocks after are placed as low as possible in the heap. Given this construction we label rows of our heap from the bottom up. That is, the row in which all of the blocks that are fully exposed to the bottom are in is row 1 and all following rows are labeled naturally. Choose a heap that contains a lowest braid. That is, choose a heap where the braid is as low as possible in the heap, where we consider the height of the braid to be determined by the lowest row number that a braid is involved in. Note that this implies means the subheap “below” the braid corresponds to an FC element.

Case (0): Suppose the braid is located in the bottom row of the heap. That is, the first block of the heap is one of the elements that is fully exposed to the bottom. Then w has the fixed reduced product $w = u s_i s_{i+1} s_i s_{i+2} v$ where v is a product of commuting generators and does not contain s_i and s_{i+2} . Here we have highlighted the braid in teal. Since the braid is in the bottom-most row of the heap s_i and s_{i+2} could be included in v ; however, for emphasis we write them individually. Applying

the braid move we obtain the reduced product $w = us_{i+1}s_is_{i+1}s_{i+2}v$. In applying the braid move, s_i is no longer in v while s_{i+2} is still in v . This implies we could write $w = us_{i+1}s + ius_{i+1}s_{i+2}$ which clearly has Property T. This is a contradiction to the way that we chose w . Thus w can not have a braid in the bottom-most row of the heap. From here forward we assume that the braid is located above the first row of the heap.

Case (1): Suppose the braid does not contain $0, 1$ or $n-1, n$. Subcase (a): Suppose w has the fixed reduced product $w = us_k s_{k-1} s_k s_{k-1} s_{k-2} v$ or $w = us_k s_{k-1} s_k s_{k+1} s_{k-1} s_{k-2} v$ where v is FC and the braid is highlighted in teal. Applying the braid move we obtain the reduced product $w = us_{k-1} s_k s_{k-1} s_{k-2} s_{k-1} v$ or $w = us_{k-1} s_k s_{k-1} s_{k-2} s_{k-1} s_{k+1} v$. Notice that the braid is now located next to v having moved closer to the bottom in the heap. This is a contradiction to choosing a heap with the lowest braid. Therefore w does not have the reduced product $w = us_k s_{k-1} s_k s_{k-1} s_{k-2} v$ or $w = us_k s_{k-1} s_k s_{k+1} s_{k-1} s_{k-2} v$. Visually we see this in the subheap of w represented in Figure 5.1, where s_{k+1} is represented as a striped block. The striped block represents the possibility that s_{k+1} may or may not be in our fixed reduced product, but illustrates that it being there does not affect our result. Notice how there are two braids located in Figure 5.1(b), the braid that starts in purple and ends with teal and the braid that is fully highlighted in teal.

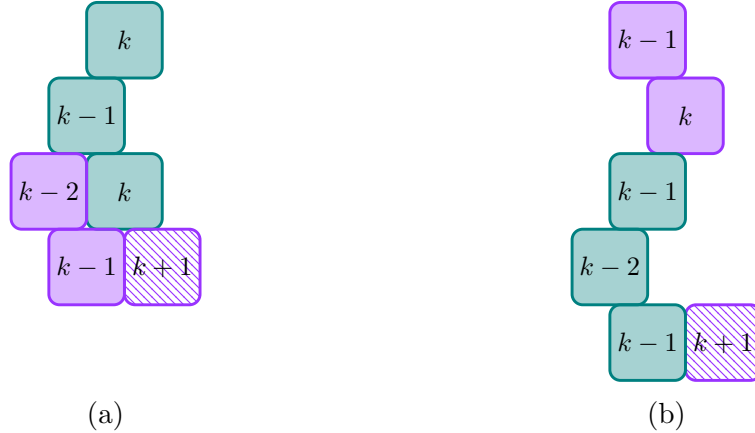


Figure 5.1: Visual representation of the heap configuration discussed in Case (1)a.

Subcase (b): Suppose w has the fixed reduced product $w = us_k s_{k-1} s_k s_{k+1} v$ where v is FC and does not contain s_{k-2} and s_{k-1} in its left descent set. Again we have highlighted the braid in teal. Applying the braid move we obtain a new reduced product $w = us_{k-1} s_k s_{k-1} s_{k+1} v$. Notice that the braid is now able to be written next to v whereas it previously was not. This again contradicts choosing a heap with the braid in the lowest location. Visually we see this in the subheap of w represented

in Figure 5.2. Notice how in Figure 5.2(b) the braid is located next to the block for s_{k+1} , whereas in Figure 5.2(a) the braid is below the block for s_{k+1} .



Figure 5.2: Visual representation of the heap configuration discussed in Case (1)b.

Case (2): Suppose the braid contains 2 or $n-2$. Without loss of generality we will take the braid to contain 2, the other argument is symmetric to the one presented here. Notice that if the braid is of the form $s_2s_3s_2$ we are in Case (1) seen above as a result we assume that the braid we refer to in the following subcases does not involve $s_2s_3s_2$ to start. Subcase (a): Suppose w has the fixed reduced product $w = us_2s_1s_2s_0s_1s_0v$, where v is FC and does not contain s_2 in the left descent set. Again we highlight the braid for emphasis in teal. Applying the braid move we obtain the reduced product $w = us_1s_2s_1s_0s_1s_0v$. Notice that the braid is now able to touch v as it was not before. This contradicts our original choice of heap and as a result we know that the reduced product $w = us_2s_1s_2s_0s_1s_0v$ has Property T and not an expression we are interested in. Visually we see this in the subheap of w represented in Figure 5.3. Notice how there are two braids located in Figure 5.3(b). The braid in the original heap that is highlighted in teal and ends with the block for s_1 highlighted in orange. The braid highlighted in orange did not appear in our original heap seen in Figure 5.3(a) and is below the original braid in the heap. This is a contradiction to the choice of the original heap, thus w is not the reduced product $w = us_2s_1s_2s_0s_1s_0v$.

Subcase (b): Suppose w has the fixed reduced product $w = us_2s_1s_2s_0s_1s_3s_2v$ where v is FC. Again we have highlighted the braid in teal. Applying the braid move we end up with the reduced product $w = us_1s_2s_1s_0s_1s_3s_2v$. Notice this time the braid does not force a higher braid. Visually we see this in the subheap of w represented in Figure 5.4. In Figures 5.4(a) and 5.4(b) we see that the braid actually moves higher in the heap.

Since we assumed that w does not have Property T, we know that u in the fixed reduced product that we have for w is non-trivial. That is, it contains some generators. Given our original reduced fixed expression for w we add a new row to our heap if we

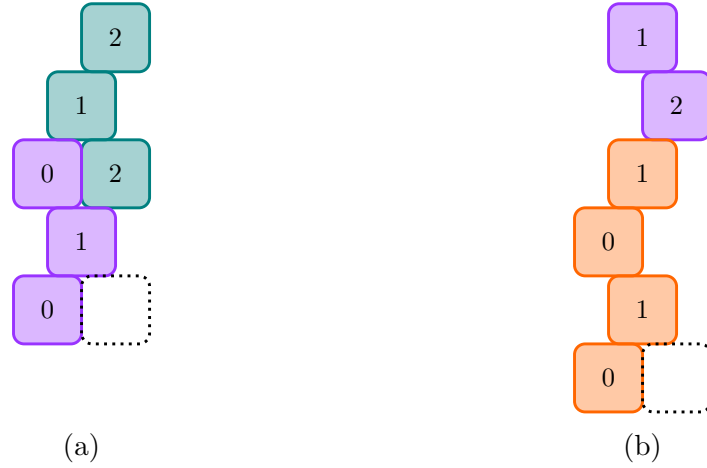


Figure 5.3: Visual representation of the heap configuration discussed in Case (2)a.



Figure 5.4: Visual representation of the heap configuration discussed in Case (2b).

were to add s_0 to the new row we would have a braid appear higher in the heap so we will not add s_0 . This forces us to add s_2 and we get the subheap configuration seen in Figure 5.5. Again this will not be the bottom row in our heap, since the group element clearly has Property T, so we know there must be another level in our heap. Notice that if s_1 is in the next level this would create a braid in our heap that will not be blocked, so s_1 is not there. This implies that s_3 will be in the next level, however in doing so we will also need to have s_4 in the level below as otherwise a braid appears lower in the heap than our original choice of heap. The resulting subheap is seen in Figure 5.6.

We once again have the same issue arise that this will not be the bottom level of our configuration as w would clearly have Property T at the bottom. Iterating the logic from above we see that our original expression must have the subheap seen in Figure 5.7.

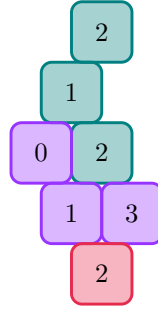


Figure 5.5

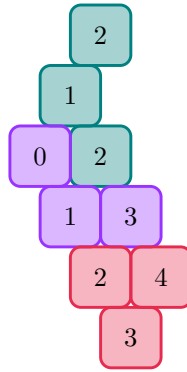


Figure 5.6

Notice that again if the bottom row of the heap in Figure 5.7 were the bottom row of the heap we would have Property T, since we could remove the block for s_{n-2} and the block for s_{n-3} would now be fully exposed to the bottom. Thus this is not the bottom row of the heap or we would have a contradiction to the way in which our heap was chosen. With this in mind we know that s_{n-1} is in the heap which will still have Property T. As a result of this logic we see that original expression must have the subheap seen in Figure 5.8.

Recall that v is FC by assumption, where a portion of v is the **red** in the above heap. In [4, Lemma 3.3] Ernst proved that an FC element of this sort has the blank space in the middle completely filled in. That is, The empty triangle shape that was created in our heap is actually has every heap block filled in. This forces our heap to look like the one seen in Figure 5.9(a) where all of the blocks in the middle of the red v are filled in. As a result of this we now have s_0 in our heap. After applying the braid move to the **teal** braid in Figure 5.9(a). This leads to the subheap seen in Figure 5.9(b), where a new **orange** braid appeared. This implies that for the fixed reduced product $w = us_2s_1s_2s_0s_1s_3s_2v$ there is a heap with a braid that is lower in the heap, a contradiction to the way in which we chose the initial heap for w . Thus

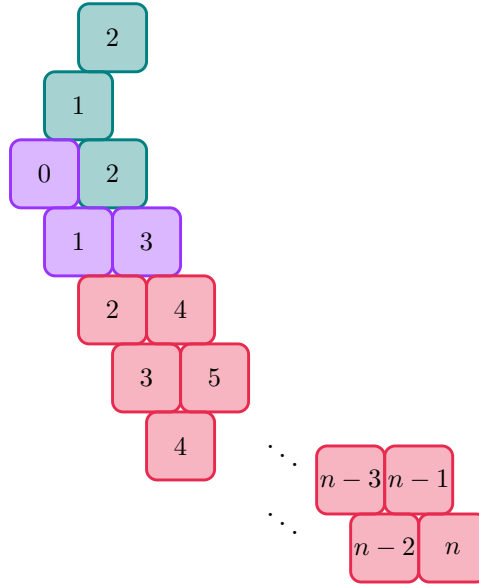


Figure 5.7

w will not be the reduced product $w = us_2s_1s_2s_0s_1s_3s_2v$.

Case (3): Suppose the braid contains 1 or $n-1$. Without loss of generality we will assume the braid contains 1, as the other argument is symmetric to the one presented here. Subcase (a): Suppose w has the reduced product $w = us_0s_1s_0s_1s_2v$ where v is FC and does not contain s_0 in its left descent set. Notice that the braid is highlighted in orange. Applying the braid move leads to the reduced product $w = us_1s_0s_1s_0s_2v$. Notice that the braid is now able to be in the same level of the heap as s_2 whereas it previously was not. That is, the block for s_0 will now be in the row that s_2 is in, whereas in our original expression the block representing s_1 was stuck in the row above s_2 as they do not commute. Visually we see this in the subheap of w represented in Figure 5.10. Notice how the braid in Figure 5.10(b) is located next to the block for s_2 whereas in Figure 5.10(a) the braid is stuck above the block for s_2 . This is a contradiction to picking the heap with the lowest braid.

Subcase (b): Suppose w has the reduced product $w = us_1s_2s_1s_0v$ where v is FC and does not contain s_2 in the left descent set. Notice that the braid is highlighted in teal. Applying the braid move leads to the reduced product $w = us_2s_1s_2s_0v$. Notice that the braid is now able to be located in the same level of the heap as s_0 whereas it previously was not. Visually we see this in the subheap of w represented in Figure 5.11. Notice how the braid in Figure 5.11(b) is located next to the block for s_0 , but the braid in Figure 5.11(a) it is stuck above the block for s_0 . This is a contradiction to the way in which we picked our heap. Thus w will not be the reduced

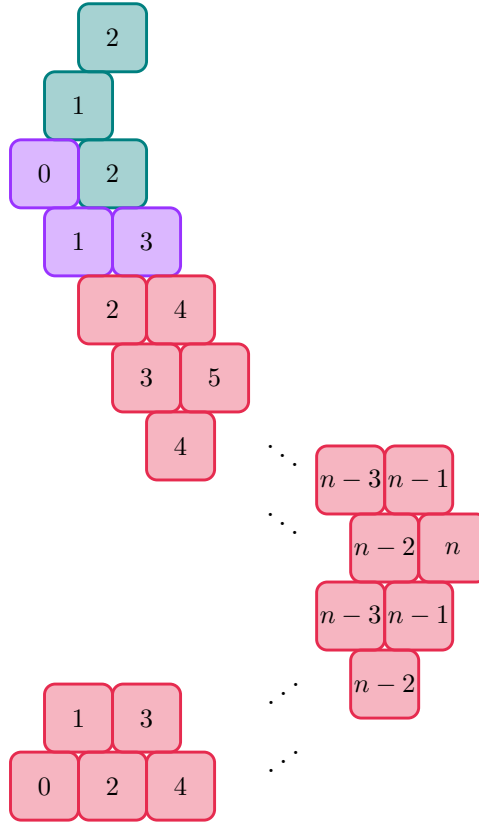


Figure 5.8

product $w = us_1s_2s_1s_0v$.

Case (4): Suppose the braid contains 0 or n . Without loss of generality we will assume the braid contains 0, as the other argument is symmetric to the one presented here. Suppose w has the fixed reduced product $w = us_1s_0s_1s_0s_2s_1v$ where v is an FC element. Notice that we have highlighted the braid in orange. Applying the braid move we obtain the reduced product $w = us_0s_1s_0s_1s_2s_1v$. In applying the braid the resulting reduced product now has the braid highlighted in teal. Notice that this braid is located next to v which is located lower in the heap than our original w . Visually we see this in the subheap of w represented in Figure 5.12. We see in Figure 5.12(b) the braid in teal is located below the braid that starts in orange and ends with s_1 . It is clear that this braid is lower than the orange braid seen in Figure 5.12(a). Thus w will not have the reduced product $w = us_1s_0s_1s_0s_2s_1v$.

From this we see there is no possible way to find a reduced expression in $W(\tilde{C}_n)/\text{FC}(\tilde{C}_n)$ with full support and does not have Property T. Thus there are no non-trivial T-avoiding elements in $W(\tilde{C}_n)/\text{FC}(\tilde{C}_n)$. \square

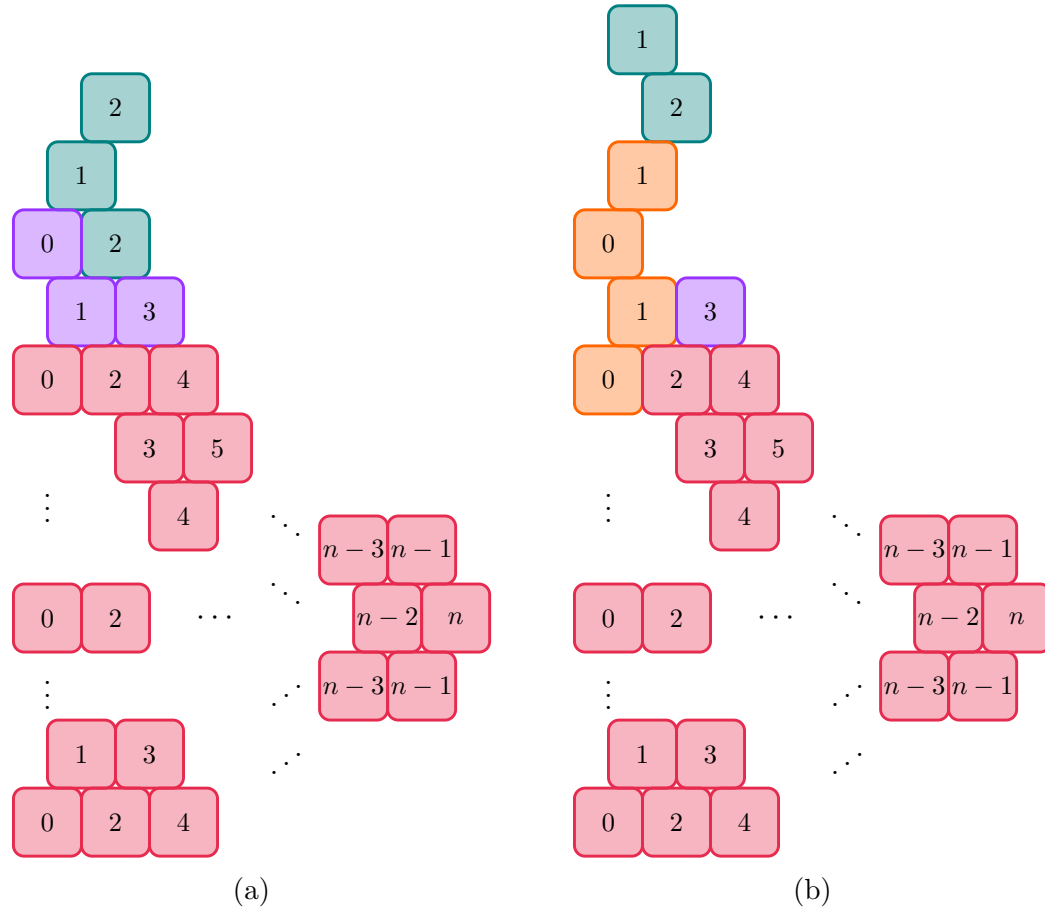


Figure 5.9: Visual Representation of the heap configuration discussed in Case (2b).



Figure 5.10: Visual representation of the heap configuration discussed in Case (3)a.

We will now consider the elements in $W(\tilde{C}_n)$ that are FC where we first will



Figure 5.11: Visual representation of the heap configuration discussed in Case (3)b.



Figure 5.12: Visual representation of the heap configuration discussed in Case (4).

classify non-trivial T-avoiding elements in $W(\tilde{C}_n)$ for n odd and then proceed to the classification for n even.

Theorem 5.1.2. If n is odd, then there are no non-trivial T-avoiding elements in the Coxeter system of type \tilde{C}_n .

Proof. Consider the Coxeter system of type \tilde{C}_n . By Theorem 5.1.1 we know that $W(\tilde{C}_n)$ contains no non-trivial T-avoiding elements that are not FC. Recall $W(\tilde{C}_n)$ is a star reducible Coxeter group, which implies that $W(\tilde{C}_n)$ contains no non-trivial T-avoiding elements that are FC. Thus as $W(\tilde{C}_n)$ has no non-trivial T-avoiding elements that are FC or not FC, $W(\tilde{C}_n)$ has no non-trivial T-avoiding elements. \square

We next will classify the non-trivial T-avoiding elements in the Coxeter system of type \tilde{C}_n for n even. Recall that $W(\tilde{C}_n)$ for n even is not a star reducible Coxeter group. In Theorem 5.1.1 we showed that $W(\tilde{C}_n)$ does not have non-trivial T-avoiding elements that are not FC. This leaves us with only the FC elements to check.

Theorem 5.1.3. If n is even, then the only non-trivial T-avoiding elements in $W(\tilde{C}_n)$ are sandwich stacks.

Proof. Let $w \in W(\tilde{C}_n)$. By Theorem 5.1.1, we know that w is an FC element. Further, we can restrict our search down to non-trivial non-cancellable elements as they are not star reducible, and thus do not contain Property T. In Section 2.3 we stated the classification of the only non-cancellable element with full support. Recall this to be sandwich stacks. Thus the only non-trivial T-avoiding elements in $W(\tilde{C}_n)$ for n odd are sandwich stacks whose top and bottom rows in the heap consist of s_0, s_2, \dots, s_n . \square

5.2 Future Work

In Sections 3.1–3.4, we relayed the known results involving T-avoiding elements in types $\tilde{A}_n, A_n, D_n, F_4, F_5$, and proved results involving T-avoiding elements in type $I_2(m)$. It remains to be shown that the conjecture in Section 3.1 regarding the classification of the non-trivial T-avoiding elements in type \tilde{A}_n holds. The classification of non-trivial T-avoiding elements in Coxeter systems of type F_n for $n \geq 6$ still remains open.

We also mentioned several other Coxeter systems in Figures 1.2 and 1.3. The classification of non-trivial T-avoiding elements in the Coxeter systems of type E_n remains an open problem. However, we do know that these groups have non-trivial T-avoiding elements as $W(D_n)$ (which has non-trivial T-avoiding elements) is a parabolic subgroup of $W(E_n)$. The classification of non-trivial T-avoiding elements in the Coxeter systems of type H_n is also an open problem.

A majority of the irreducible affine Coxeter systems currently do not have a classification of the non-trivial T-avoiding elements. Specifically, Coxeter systems of type $\tilde{B}_n, \tilde{D}_n, \tilde{E}_6, \tilde{E}_7, \tilde{E}_8$, and \tilde{G}_4 do not have a classification. Future work could include classifying the non-trivial T-avoiding elements of the Coxeter systems mentioned above.

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] A Björner and F Brenti. *Combinatorics of Coxeter groups*. 2005.
- [3] D.C. Ernst. Non-cancellable elements in type affine C Coxeter groups. *Int. Electron. J. Algebr.*, 8, 2010.
- [4] D.C. Ernst. Diagram calculus for a type affine C Temperley–Lieb algebra, II. [arXiv1101.4215v1](#), 2012.
- [5] C.K. Fan. Structure of a Hecke algebra quotient. *J. Amer. Math. Soc.*, 10:139–167, 1997.
- [6] C.K. Fan and R.M. Green. On the affine Temperley–Lieb algebras. *Jour. L.M.S.*, 60:366–380, 1999.
- [7] M. Geck and G. Pfeiffer. *Characters of finite Coxeter groups and Iwahori–Hecke algebras*. 2000.
- [8] T. Gern. *Leading Coefficients of Kazhdan–Lusztig Polynomials in Type D*. PhD, University of Colorado, 2013.
- [9] R.M. Green. Star reducible Coxeter groups. *Glas. Math. J.*, 48:583–609, 2006.
- [10] J.E. Humphreys. *Reflection Groups and Coxeter Groups*. 1990.
- [11] D. Kazhdan and G. Lusztig. Representations of Coxeter groups and Hecke algebras. *Inven. Math.*, 53:165–184, 1979.
- [12] G. Lusztig. Cells in affine Weyl groups, I. In *Collection*, pages 255–287. 1985.
- [13] J.R. Stembridge. On the fully commutative elements of Coxeter groups. *J. Algebr. Comb.*, 5:353–385, 1996.