

A Study of T-Avoiding Elements in Coxeter Groups

Taryn Laird

Northern Arizona University
Department of Mathematics and Statistics

Colloquium
May 3, 2016

Definition

A **Coxeter system** consists of a group W (called a **Coxeter group**) generated by a set S of involutions with presentation

$$W = \langle S \mid s^2 = e, (st)^{m(s,t)} = e \rangle$$

where $m(s, t) \geq 2$ for all $s \neq t$.

Definition

A **Coxeter system** consists of a group W (called a **Coxeter group**) generated by a set S of involutions with presentation

$$W = \langle S \mid s^2 = e, (st)^{m(s,t)} = e \rangle$$

where $m(s, t) \geq 2$ for all $s \neq t$.

Comment

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

$$m(s, t) = 2 \implies st = ts \quad \} \quad \text{commutations}$$

Definition

A **Coxeter system** consists of a group W (called a **Coxeter group**) generated by a set S of involutions with presentation

$$W = \langle S \mid s^2 = e, (st)^{m(s,t)} = e \rangle$$

where $m(s, t) \geq 2$ for all $s \neq t$.

Comment

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

$$m(s, t) = 2 \implies st = ts \quad \} \quad \text{commutations}$$

$$\left. \begin{array}{l} m(s, t) = 3 \implies sts = tst \\ m(s, t) = 4 \implies stst = tsts \\ \vdots \end{array} \right\} \quad \text{braid relations}$$

Definition

We can encode (W, S) with a unique Coxeter graph Γ having:

- vertex set S ;
- edges $\{s, t\}$ labeled $m(s, t)$ whenever $m(s, t) \geq 3$;

Definition

We can encode (W, S) with a unique Coxeter graph Γ having:

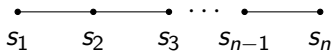
- vertex set S ;
- edges $\{s, t\}$ labeled $m(s, t)$ whenever $m(s, t) \geq 3$;

Comments

- If $m(s, t) = 3$, we omit label.
- If s and t are not connected in Γ , then s and t commute.
- Given Γ , we can uniquely reconstruct the corresponding (W, S) .

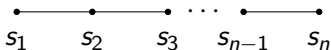
Coxeter Groups of Type A

Coxeter groups of type A_n ($n \geq 1$) are defined by:



Coxeter Groups of Type A

Coxeter groups of type A_n ($n \geq 1$) are defined by:

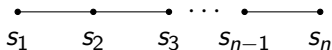


Then $W(A_n)$ is generated by $\{s_1, s_2, \dots, s_n\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$.

Coxeter Groups of Type A

Coxeter groups of type A_n ($n \geq 1$) are defined by:



Then $W(A_n)$ is generated by $\{s_1, s_2, \dots, s_n\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$.

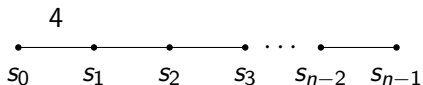
$W(A_n)$ is isomorphic to the symmetric group, Sym_{n+1} , under the correspondence

$$s_i \mapsto (i, i + 1),$$

where $(i, i + 1)$ is the adjacent transposition exchanging i and $i + 1$.

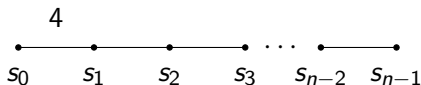
Coxeter Groups of Type B

Coxeter groups of type B_n ($n \geq 2$) are defined by:



Coxeter Groups of Type B

Coxeter groups of type B_n ($n \geq 2$) are defined by:

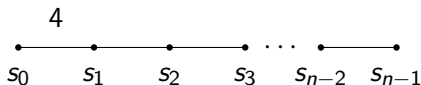


Then $W(B_n)$ is generated by $\{s_0, s_1, \dots, s_{n-1}\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$ and $1 < i, j \leq n - 1$,
4. $s_0 s_1 s_0 s_1 = s_1 s_0 s_1 s_0$.

Coxeter Groups of Type B

Coxeter groups of type B_n ($n \geq 2$) are defined by:



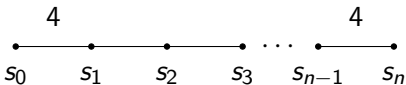
Then $W(B_n)$ is generated by $\{s_0, s_1, \dots, s_{n-1}\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$ and $1 < i, j \leq n - 1$,
4. $s_0 s_1 s_0 s_1 = s_1 s_0 s_1 s_0$.

$W(B_n) \cong \text{Sym}_n^B$ is a finite group of order $n!2^n$.

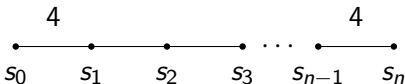
Coxeter Groups of Type Affine C

Coxeter groups of type \tilde{C}_n ($n \geq 2$) are defined by:



Coxeter Groups of Type Affine C

Coxeter groups of type \tilde{C}_n ($n \geq 2$) are defined by:

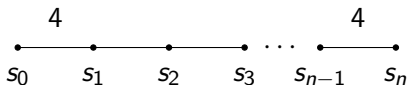


Here, we see that $W(\tilde{C}_n)$ is generated by $\{s_0, \dots, s_n\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$ and $1 < i, j < n$,
4. $s_i s_j s_i s_j = s_j s_i s_j s_i$ if $\{i, j\} = \{0, 1\}$ or $\{n - 1, n\}$.

Coxeter Groups of Type Affine C

Coxeter groups of type \tilde{C}_n ($n \geq 2$) are defined by:



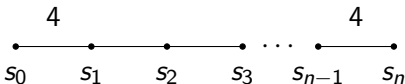
Here, we see that $W(\tilde{C}_n)$ is generated by $\{s_0, \dots, s_n\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$ and $1 < i, j < n$,
4. $s_i s_j s_i s_j = s_j s_i s_j s_i$ if $\{i, j\} = \{0, 1\}$ or $\{n - 1, n\}$.

$W(\tilde{C}_n)$ is an infinite group.

Coxeter Groups of Type Affine C

Coxeter groups of type \tilde{C}_n ($n \geq 2$) are defined by:



Here, we see that $W(\tilde{C}_n)$ is generated by $\{s_0, \dots, s_n\}$ and is subject to defining relations

1. $s_i^2 = e$ for all i ,
2. $s_i s_j = s_j s_i$ if $|i - j| > 1$,
3. $s_i s_j s_i = s_j s_i s_j$ if $|i - j| = 1$ and $1 < i, j < n$,
4. $s_i s_j s_i s_j = s_j s_i s_j s_i$ if $\{i, j\} = \{0, 1\}$ or $\{n - 1, n\}$.

$W(\tilde{C}_n)$ is an infinite group.

Comment

We can obtain $W(A_n)$ and $W(B_n)$ from $W(\tilde{C}_n)$ by removing the appropriate generators and corresponding relations. In fact, we can obtain $W(B_n)$ in two ways.

Definition

A word $s_{x_1}s_{x_2}\cdots s_{x_m}$ is called an **expression** for $w \in W$ if it is equal to w when considered as a group element.

Definition

A word $s_{x_1}s_{x_2}\cdots s_{x_m}$ is called an **expression** for $w \in W$ if it is equal to w when considered as a group element.

We define the **length** of w , $\ell(w)$, to be the smallest m for which w is a product of m generators, such an expression is called **reduced**.

Definition

A word $s_{x_1}s_{x_2}\cdots s_{x_m}$ is called an **expression** for $w \in W$ if it is equal to w when considered as a group element.

We define the **length** of w , $\ell(w)$, to be the smallest m for which w is a product of m generators, such an expression is called **reduced**.

Given $w \in W$, if we wish to emphasize a fixed, possibly reduced, expression for w , we represent it as

$$\overline{w} = s_{x_1}s_{x_2}\cdots s_{x_m}.$$

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Definition

We define the **left** (respectively, **right**) **descent set** w as follows:

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

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Definition

We define the **left** (respectively, **right**) **descent set** w as follows:

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

Example

Let $\overline{w} = s_2 s_1 s_2 s_3 s_1$ be a fixed expression for $w \in W(A_3)$. We see that

$$s_2 s_1 s_2 s_3 s_1$$

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Definition

We define the **left** (respectively, **right**) **descent set** w as follows:

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

Example

Let $\overline{w} = s_2 s_1 s_2 s_3 s_1$ be a fixed expression for $w \in W(A_3)$. We see that

$$s_2 s_1 s_2 s_3 s_1 = s_1 s_2 s_1 s_3 s_1$$

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Definition

We define the **left** (respectively, **right**) **descent set** w as follows:

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

Example

Let $\overline{w} = s_2 s_1 s_2 s_3 s_1$ be a fixed expression for $w \in W(A_3)$. We see that

$$s_2 s_1 s_2 s_3 s_1 = s_1 s_2 s_1 s_3 s_1 = s_1 s_2 s_1 s_1 s_3$$

Matsumoto's Theorem and Support

Theorem (Matsumoto)

Any two reduced expressions for $w \in W$ differ by a sequence of commutations and braid moves.

Definition

We define $\text{supp}(w)$ to be the set of generators appearing in any reduced expression for w . This is well-defined by Matsumoto's Theorem.

Definition

We define the **left** (respectively, **right**) **descent set** w as follows:

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

Example

Let $\overline{w} = s_2 s_1 s_2 s_3 s_1$ be a fixed expression for $w \in W(A_3)$. We see that

$$s_2 s_1 s_2 s_3 s_1 = s_1 s_2 s_1 s_3 s_1 = s_1 s_2 s_1 s_1 s_3 = s_1 s_2 s_3$$

Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.

Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.

Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

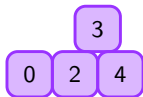
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

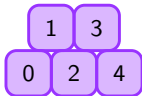
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

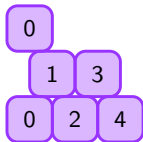
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

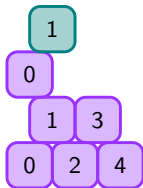
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

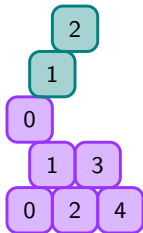
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

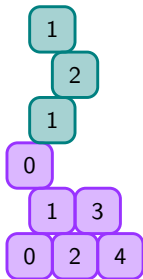
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

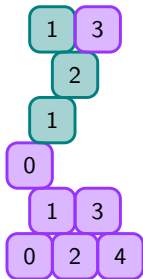
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

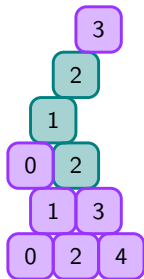
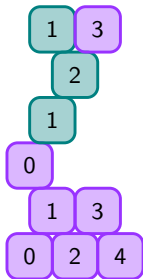
Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Every reduced expression \overline{w} can be represented by a labeled partially ordered set (poset) called a heap, denoted $H(\overline{w})$. Heaps provide a visual representation of a reduced expression while preserving the relations among the generators.

Example

Let $\overline{w} = s_3 s_1 s_2 s_1 s_0 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(B_6)$.



Fully Commutative Elements

Definition

Let (W, S) be a Coxeter system of type Γ . We say that $w \in W(\Gamma)$ is **fully commutative** (FC) if any two reduced expressions for w can be transformed into each other via iterated commutations. The set of FC elements is denoted $FC(\Gamma)$.

Fully Commutative Elements

Definition

Let (W, S) be a Coxeter system of type Γ . We say that $w \in W(\Gamma)$ is **fully commutative** (FC) if any two reduced expressions for w can be transformed into each other via iterated commutations. The set of FC elements is denoted $FC(\Gamma)$.

Theorem (Stembridge)

$w \in FC(\Gamma)$ if and only if no reduced expression for w contains a braid.

Fully Commutative Elements

Definition

Let (W, S) be a Coxeter system of type Γ . We say that $w \in W(\Gamma)$ is **fully commutative** (FC) if any two reduced expressions for w can be transformed into each other via iterated commutations. The set of FC elements is denoted $FC(\Gamma)$.

Theorem (Stembridge)

$w \in FC(\Gamma)$ if and only if no reduced expression for w contains a braid.

Comment

It turns out that $W(\tilde{C}_n)$ contains an infinite number of FC elements, while $W(A_n)$ and $W(B_n)$ do not.

Fully Commutative Elements

Comment

The elements of $\text{FC}(\tilde{C}_n)$ are precisely those whose reduced expressions avoid the consecutive subwords $s_i s_j s_i$ for $m(s_i, s_j) = 3$, $s_0 s_1 s_0 s_1$, and $s_{n-1} s_n s_{n-1} s_n$.

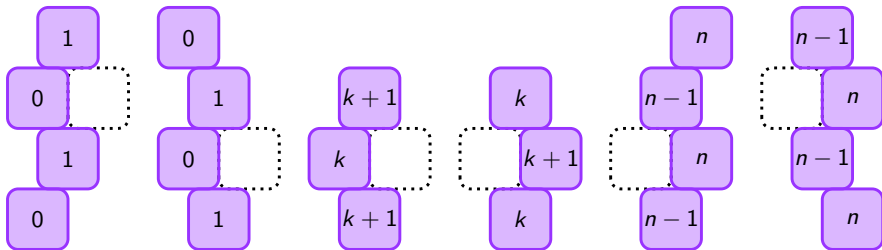
Fully Commutative Elements

Comment

The elements of $FC(\tilde{C}_n)$ are precisely those whose reduced expressions avoid the consecutive subwords $s_i s_j s_i$ for $m(s_i, s_j) = 3$, $s_0 s_1 s_0 s_1$, and $s_{n-1} s_n s_{n-1} s_n$.

Theorem

Let $w \in FC(\tilde{C}_n)$. Then $H(w)$ can not contain any of the following convex subheaps



Theorem (Stembridge)

There is a unique heap for w if and only if w is FC.

Theorem (Stembridge)

There is a unique heap for w if and only if w is FC.

Example

Let $\overline{w} = s_0 s_2 s_4 s_3 s_2 s_1$ be a reduced expression for $w \in W(\tilde{C}_4)$. We see that

$$s_0 s_2 s_4 s_3 s_2 s_1 = s_0 s_4 s_2 s_3 s_2 s_1 = s_0 s_4 s_3 s_2 s_1 s_3.$$

Since w has one of the forbidden consecutive subwords, w is **not** FC.

Fully Commutative

Theorem (Stembridge)

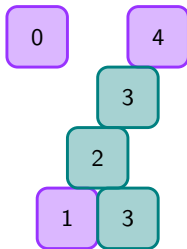
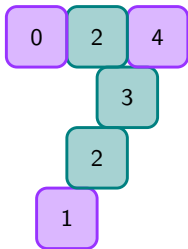
There is a unique heap for w if and only if w is FC.

Example

Let $\overline{w} = s_0 s_2 s_4 s_3 s_2 s_1$ be a reduced expression for $w \in W(\tilde{C}_4)$. We see that

$$s_0 \textcolor{red}{s_2 s_4} s_3 s_2 s_1 = s_0 s_4 \textcolor{red}{s_2 s_3 s_2} s_1 = s_0 s_4 s_3 s_2 s_1 s_3.$$

Since w has one of the forbidden consecutive subwords, w is **not** FC.

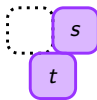
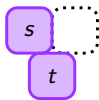


Definition

We define w to be **left star reducible by s with respect to t** if $m(s, t) \geq 3$, $s \in \mathcal{L}(w)$ and $t \in \mathcal{L}(sw)$. Analogous definition for **right star reducible**.

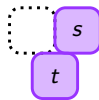
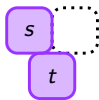
Definition

We define w to be **left star reducible by s with respect to t** if $m(s, t) \geq 3$, $s \in \mathcal{L}(w)$ and $t \in \mathcal{L}(sw)$. Analogous definition for **right star reducible**.



Definition

We define w to be **left star reducible by s with respect to t** if $m(s, t) \geq 3$, $s \in \mathcal{L}(w)$ and $t \in \mathcal{L}(sw)$. Analogous definition for **right star reducible**.

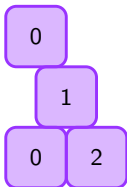


Definition

We define $W(\Gamma)$ to be **star reducible** if every element of $FC(\Gamma)$ can be reduced to a product of commuting generators via a sequence of star reductions.

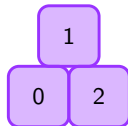
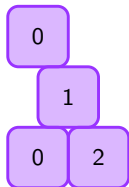
Example

Let $\overline{w} = s_0 s_1 s_0 s_2$ be a reduced expression for $w \in W(B_4)$.



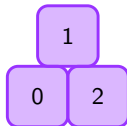
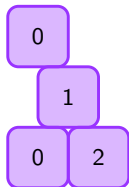
Example

Let $\overline{w} = s_0 s_1 s_0 s_2$ be a reduced expression for $w \in W(B_4)$.



Example

Let $\overline{w} = s_0 s_1 s_0 s_2$ be a reduced expression for $w \in W(B_4)$.



Theorem (Green)

There is a complete list of star reducible Coxeter systems. These include Coxeter systems of 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 $I_2(m)$ ($m \geq 3$), type \tilde{A}_n ($n \geq 3$ and n even), and type \tilde{C}_n ($n \geq 3$ and n odd).

Definition

We define w to have **Property T** if and only if there exists a reduced product for w such that $w = stu$ or $w = uts$ where $m(s, t) \geq 3$.

Definition

We define w to have **Property T** if and only if there exists a reduced product for w such that $w = stu$ or $w = uts$ where $m(s, t) \geq 3$.

We say w is **T-avoiding** if w does not have Property T.

Definition

We define w to have **Property T** if and only if there exists a reduced product for w such that $w = stu$ or $w = uts$ where $m(s, t) \geq 3$.

We say w is **T-avoiding** if w does not have Property T.

Proposition

A product of commuting generators is T-avoiding.

Definition

We define w to have **Property T** if and only if there exists a reduced product for w such that $w = stu$ or $w = uts$ where $m(s, t) \geq 3$.

We say w is **T-avoiding** if w does not have Property T.

Proposition

A product of commuting generators is T-avoiding.

Definition

We define w to be a **trivial T-avoiding** element if w is a product of commuting generators. Otherwise, we say w is a **non-trivial T-avoiding** element.

Examples of Property T and T-avoiding

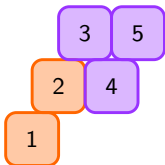
Example

Let $\overline{w} = s_5 s_3 s_2 s_4 s_1$ be a reduced expression for $w \in W(A_5)$.

Examples of Property T and T-avoiding

Example

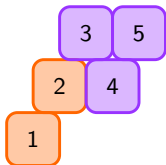
Let $\overline{w} = s_5 s_3 s_2 s_4 s_1$ be a reduced expression for $w \in W(A_5)$.



Examples of Property T and T-avoiding

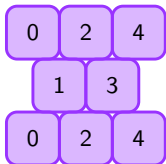
Example

Let $\overline{w} = s_5 s_3 s_2 s_4 s_1$ be a reduced expression for $w \in W(A_5)$.



Example

Let $\overline{w} = s_0 s_2 s_4 s_1 s_3 s_0 s_2 s_4$ be a reduced expression for $w \in W(\tilde{C}_4)$.



Theorem (Fan, Green)

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

Theorem (Fan, Green)

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

Corollary

There are no non-trivial T-avoiding elements in $W(A_n)$.

Known Classifications

Theorem (Fan, Green)

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

Corollary

There are no non-trivial T-avoiding elements in $W(A_n)$.

Conjecture

The only non-trivial T-avoiding elements of $W(\tilde{A}_n)$ 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 $k \in \mathbb{Z}^+$.

Known Classifications

Theorem (Fan, Green)

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

Corollary

There are no non-trivial T-avoiding elements in $W(A_n)$.

Conjecture

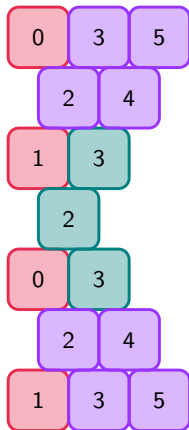
The only non-trivial T-avoiding elements of $W(\tilde{A}_n)$ for n odd are of the form $w = (s_0 s_2 \cdots s_{n-2} s_n s_1 s_3 \cdots s_{n-3} s_{n-1})^k$ for $k \in \mathbb{Z}^+$.

Theorem (Laird?)

There are no non-trivial T-avoiding elements in $W(I_2(m))$.

Theorem (Gern)

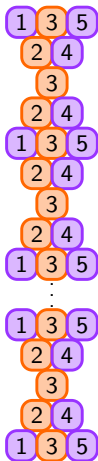
The only non-trivial T-avoiding elements in $W(D_n)$ have this pattern:



Known Classifications

Theorem (Cross, Ernst, Hills-Kimball, Quaranta)

The only non-trivial T-avoiding elements in F_5 are stacks of bowties:



Corollary (Cross, Ernst, Hills-Kimball, Quaranta)

There are no non-trivial T-avoiding elements in F_4 .

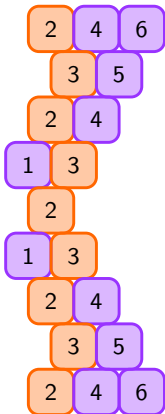
Known Classifications

Corollary (Cross, Ernst, Hills-Kimball, Quaranta)

There are no non-trivial T-avoiding elements in F_4 .

Comment

Classifying non-trivial T-avoiding elements in F_n for $n \geq 6$ gets very difficult.



Classification of T-Avoiding Elements: B_n

Theorem (Laird)

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

Classification of T-Avoiding Elements: B_n

Theorem (Laird)

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

123	<u>1</u> 23	1 <u>2</u> 3	12 <u>3</u>	<u>1</u> 23	<u>1</u> 23	1 <u>2</u> 3	1 <u>2</u> 3
132	<u>1</u> 32	1 <u>3</u> 2	13 <u>2</u>	<u>1</u> 32	<u>1</u> 32	1 <u>3</u> 2	1 <u>3</u> 2
213	<u>2</u> 13	2 <u>1</u> 3	21 <u>3</u>	<u>2</u> 13	<u>2</u> 13	2 <u>1</u> 3	2 <u>1</u> 3
231	<u>2</u> 31	2 <u>3</u> 1	23 <u>1</u>	<u>2</u> 31	<u>2</u> 31	2 <u>3</u> 1	2 <u>3</u> 1
312	<u>3</u> 12	3 <u>1</u> 2	31 <u>2</u>	<u>3</u> 12	<u>3</u> 12	3 <u>1</u> 2	3 <u>1</u> 2
321	<u>3</u> 21	3 <u>2</u> 1	32 <u>1</u>	<u>3</u> 21	<u>3</u> 21	3 <u>2</u> 1	3 <u>2</u> 1

Through a series of lemmas we were able to determine if a reduced product ends or begins with st given that w contains a certain signed consecutive pattern.

Classification of T-Avoiding Elements: \tilde{C}_n

Theorem (Laird)

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

Classification of T-Avoiding Elements: \tilde{C}_n

Theorem (Laird)

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

Theorem (Laird)

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

Classification of T-Avoiding Elements: \tilde{C}_n

Theorem (Laird)

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

Theorem (Laird)

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

Theorem (Laird)

If n is even, then the only non-trivial T-avoiding elements in Coxeter systems of type \tilde{C}_n are sandwich stacks.

Comment

Recall that Coxeter systems of Type D_n and F_n have non-trivial T-avoiding elements that are not FC. Also Coxeter systems of Type \tilde{A}_n and \tilde{C}_n for appropriate choice of n have non-trivial T-avoiding elements that are FC.

In all of the examples we have seen so far the non-trivial T-avoiding elements are either only FC or only not FC.