Depth for classical Coxeter groups

Riccardo Biagioli (Lyon 1)

Eli Bagno, Mordechai Novick (Jerusalem College of Tech.)
and Alexander Woo (U. Idaho)

SLC 75 and IICA 20
Bertinoro, 9 settembre 2015

Sorting by transpositions

One can imagine various “machines” that can sort permutations (to the identity) by swapping pairs of entries.

Machine ℓ: Can only swap adjacent entries, and every move costs 1.

Machine a: Can swap arbitrary pairs of entries, and every move costs 1.

Machine d: Can swap arbitrary pairs of entries, and a move costs the distance between the entries.

Question: Can we look at a permutation and easily tell the minimum cost to sort it?

Inversions

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**.
One optimal algorithm is to always swap the rightmost descent.

For $w = 2537146$, we have

$$2537146 \to 2531746 \to 2531476 \to 2531467 \to 2513467$$
$$\to 2153467 \to 2135467 \to 2134567 \to 1234567$$

So $\ell(w) = 8$, and we have $1 + 3 + 1 + 3 = 8$ inversions.

Inversions

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**.
One optimal algorithm is to always swap the rightmost descent.

For $w = 2537146$, we have

$$253\mathbf{71}46 \to 2531\mathbf{74}6 \to 25314\mathbf{76} \to 25\mathbf{31}467 \to 2\mathbf{51}3467$$
$$\to 21\mathbf{53}467 \to 213\mathbf{54}67 \to \mathbf{21}34567 \to 1234567$$

So $\ell(w) = 8$, and we have $1 + 3 + 1 + 3 = 8$ inversions.

Inversions

For Machine ℓ, the answer is called the **length** of the permutation, and it is equal to the **number of inversions**.
One optimal algorithm is to always swap the rightmost descent.

For $w = 2537146$, we have

$$253\mathbf{71}46 \to 2531\mathbf{74}6 \to 25314\mathbf{76} \to 25\mathbf{31}467 \to 2\mathbf{51}3467$$
$$\to 21\mathbf{53}467 \to 213\mathbf{54}67 \to \mathbf{21}34567 \to 1234567$$

So $\ell(w) = 8$, and we have $1 + 3 + 1 + 3 = 8$ inversions.

Cycles

For Machine *a*, the answer is called the **absolute length** or **reflection length**, and it is equal to n **minus the number of cycles**.

One optimal algorithm (called "straigh selection sort" by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2537146$, we have

$$253\mathbf{7}14\mathbf{6} \to 253\mathbf{6}14\mathbf{7} \to 2\mathbf{5}34\mathbf{1}67 \to \mathbf{21}34567 \to 1234567$$

So $a(w) = 4$. We have $n = 7$ and 3 cycles, since $w = (125)(476)(3)$.

Cycles

For Machine a, the answer is called the **absolute length** or **reflection length**, and it is equal to n **minus the number of cycles**.

One optimal algorithm (called “straigh selection sort” by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2537146$, we have

$$253\mathbf{7}14\mathbf{6} \to 253\mathbf{6}14\mathbf{7} \to 2\mathbf{5}34\mathbf{1}67 \to \mathbf{21}34567 \to 1234567$$

So $a(w) = 4$. We have $n = 7$ and 3 cycles, since $w = (125)(476)(3)$.

Cycles

For Machine a, the answer is called the **absolute length** or **reflection length**, and it is equal to n **minus the number of cycles**.
One optimal algorithm (called "straigh selection sort" by Knuth) is to always swap the largest misplaced entry to its correct location.

For $w = 2537146$, we have

$$253\mathbf{7}14\mathbf{6} \to 253\mathbf{6}1\mathbf{4}7 \to 2\mathbf{5}34\mathbf{1}67 \to \mathbf{21}34567 \to 1234567$$

So $a(w) = 4$. We have $n = 7$ and 3 cycles, since
$w = (125)(476)(3)$.

Sum of the sizes of exceedances

For Machine d, the answer is called the **depth**, and Petersen–Tenner showed it is equal to the **sum of the sizes of exceedances**, i.e.

$$d(w) = \sum_{w(i)>i} (w(i) - i).$$

One optimal algorithm is to always swap the rightmost exceedance with the leftmost sub-exceedance to its right.

For $w = 2537146$, we have

$$253\mathbf{71}46 \xrightarrow{1} 2531\mathbf{74}6 \xrightarrow{1} 25314\mathbf{76} \xrightarrow{1} 2\mathbf{5}3\mathbf{1}467 \xrightarrow{2} 213\mathbf{5}\mathbf{4}67$$
$$\xrightarrow{1} \mathbf{21}34567 \xrightarrow{1} 1234567$$

So $d(w) = 7$, and the sum of sizes of exceedances is $1 + 3 + 0 + 3 + 0 + 0 + 0 = 7$.

Sum of the sizes of exceedances

For Machine d, the answer is called the **depth**, and Petersen–Tenner showed it is equal to the **sum of the sizes of exceedances**, i.e.

$$d(w) = \sum_{w(i)>i} (w(i) - i).$$

One optimal algorithm is to always swap the rightmost exceedance with the leftmost sub-exceedance to its right.

For $w = 2537146$, we have

$$253\mathbf{7}\mathbf{1}46 \xrightarrow{1} 2531\mathbf{7}\mathbf{4}6 \xrightarrow{1} 25314\mathbf{7}\mathbf{6} \xrightarrow{1} 2\mathbf{5}3\mathbf{1}467 \xrightarrow{2} 213\mathbf{5}\mathbf{4}67$$
$$\xrightarrow{1} \mathbf{2}\mathbf{1}34567 \xrightarrow{1} 1234567$$

So $d(w) = 7$, and the sum of sizes of exceedances is $1 + 3 + 0 + 3 + 0 + 0 + 0 = 7$.

Sum of the sizes of exceedances

For Machine d, the answer is called the **depth**, and Petersen–Tenner showed it is equal to the **sum of the sizes of exceedances**, i.e.

$$d(w) = \sum_{w(i)>i} (w(i) - i).$$

One optimal algorithm is to always swap the rightmost exceedance with the leftmost sub-exceedance to its right.

For $w = 2537146$, we have

$$253\mathbf{71}46 \xrightarrow{1} 2531\mathbf{74}6 \xrightarrow{1} 25314\mathbf{76} \xrightarrow{1} 2\mathbf{5}3\mathbf{1}467 \xrightarrow{2} 213\mathbf{54}67$$
$$\xrightarrow{1} \mathbf{21}34567 \xrightarrow{1} 1234567$$

So $d(w) = 7$, and the sum of sizes of exceedances is $1 + 3 + 0 + 3 + 0 + 0 + 0 = 7$.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w) + \ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations. (Petersen–Tenner)

- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations. (Tenner)

- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w)+\ell(w)}{2} \leq d(w) \leq \ell(w).$$

• The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations. (Petersen–Tenner)

• The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations. (Tenner)

• It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w) + \ell(w)}{2} \leq d(w) \leq \ell(w).$$

- The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations. (Petersen–Tenner)

- The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations. (Tenner)

- It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

Cost Coincidences

Petersen and Tenner observed that

$$a(w) \leq \frac{a(w)+\ell(w)}{2} \leq d(w) \leq \ell(w).$$

• The permutations for which $d(w) = \ell(w)$ are the 321 avoiding permutations. (Petersen–Tenner)

• The permutations for which $d(w) = a(w)$ (and hence $a(w) = \ell(w)$) are the 321 and 3412 avoiding permutations. (Tenner)

• It seems like a hard problem to characterize the permutations for which $d(w) = (a(w) + \ell(w))/2$ by pattern avoidance.

The group B_n

A **signed permutation** is a permutation w on the set $\{\pm 1, \ldots, \pm n\}$ with the property that $w(-i) = -w(i)$ for all i.

It suffices to specify $w(i)$ for $i > 0$, so we can think of a signed permutation as a permutation with the additional property that some of the entries have a negative sign.

We denote $neg(w)$ the **number of negative entries** of w.

For example, we might have $w = 2\bar{4}3\bar{1}7\bar{5}6$. (To save space, we draw the negative signs on top of the numbers.)

Machines for B_n

- Machine ℓ can swap two adjacent entries or change the sign of the leftmost entry (each costs 1).
- Machine a can (each costs 1) :

Shuffling: swap a pair of entries at positions i and j

Double unsigning: swap a pair of entries at positions i and j and change both signs

Single unsigning: change the sign of the entry at position i

- Machine d costs

Shuffling: $j - i$ (as for permutations)

Double unsigning: $i + j - 1$

Single unsigning: i

Length for B_n

The cost for machine ℓ is the total count of the following:

- Positions $i < j$ with $w(i) > w(j)$
- Positions $i < j$ with $w(i) + w(j) < 0$
- Positions i with $w(i) < 0$

For $w = 2\bar{4}3\bar{1}7\bar{5}6$, we have
$\ell(w) = (3+1+2+1+2) + (2+3+1+1) + 3 = 19$, with sorting algorithm

$$2\bar{4}3\bar{1}\mathbf{7}\bar{\mathbf{5}}6 \to 2\bar{4}3\bar{1}\bar{5}\mathbf{76} \to 2\bar{4}3\bar{\mathbf{1}}\bar{\mathbf{5}}67 \to 2\bar{4}\mathbf{3}\bar{\mathbf{5}}\bar{1}67 \to \cdots \to \bar{\mathbf{5}}\bar{4}\bar{1}2367$$
$$\to 5\bar{4}\bar{1}2367 \to \cdots \to \bar{4}\bar{1}23567 \to 4\bar{1}23567 \to \cdots \to 1234567$$

Oddness of a signed permutation

We can have a sum $\oplus$ of signed permutations and sum decompositions defined by ignoring the signs. For example, $2\bar{4}3\bar{1}7\bar{5}6 = 2\bar{4}3\bar{1} \oplus 3\bar{1}2$ is the sum decomposition.

Given a signed permutation w, define the **oddness** of w to be the number of blocks in the sum decomposition with an odd number of signed elements, denoted $o(w)$.

The negative identity $\bar{1} \cdots \bar{n}$ is the oddest element, with oddness n.

Oddness of a signed permutation

We can have a sum $\oplus$ of signed permutations and sum decompositions defined by ignoring the signs. For example, $2\bar{4}3\bar{1}7\bar{5}6 = 2\bar{4}3\bar{1} \oplus 3\bar{1}2$ is the sum decomposition.

Given a signed permutation w, define the **oddness** of w to be the number of blocks in the sum decomposition with an odd number of signed elements, denoted $o(w)$.

The negative identity $\bar{1} \cdots \bar{n}$ is the oddest element, with oddness n.

Depth for a signed permutation

Theorem [BBNW, 2015]
We have the following formula for depth for B_n

$$d(w) = \left(\sum_{w(i)>i} (w(i) - i) \right) + \left(\sum_{w(i)<0} |w(i)| \right) + \left(\frac{o(w) - neg(w)}{2} \right).$$

Single unsigning moves are slightly expensive, and $o(w)$ counts how many times they need to be used.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at least two negative entries, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is one negative entry, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at least two negative entries, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is one negative entry, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at least two negative entries, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is one negative entry, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm for signed permutations

To sort a signed permutation w using the minimum depth, we do the following to each block in the sum decomposition:

1. If possible apply a shuffling move to positions i and j, where $x = w(i)$ is the largest positive entry in w with $x > i$, and $y = w(j)$ is the smallest entry in w with $i < j \leq x$. Repeat this step until there is no positive entry $x = w(i)$ with $x > i$.
2. If there are at least two negative entries, apply a double unsigning move at positions i and j, where $x = w(i)$ and $y = w(j)$ are the two negative entries of largest absolute value in w, and go back to Step 1.
3. If there is one negative entry, apply a single unsigning move the negative entry, and go back to Step 1.

Algorithm example

For $w = 2\bar{4}3\bar{1}7\bar{5}6 = [2\bar{4}3\bar{1}] \oplus [3\bar{1}2]$, the formula gives
$d(w) = (1+2) + (4+1+5) + (1-3)/2 = 12$

$$2\bar{4}3\bar{1}\_\mathbf{7\bar{5}}6 \xrightarrow{1} 2\bar{4}3\bar{1}\_\bar{5}\mathbf{76} \xrightarrow{1} 2\bar{4}3\bar{1}\_\mathbf{\bar{5}}67 \xrightarrow{5} \mathbf{2\bar{4}}3\bar{1}\_567$$
$$\xrightarrow{1} \mathbf{\bar{4}}23\mathbf{\bar{1}}\_567 \xrightarrow{4} 1234567$$

The group D_n

Consider

$$D_n = \{w \in B_n \mid neg(w) \equiv 0 \text{ (mod) } 2\}.$$

• Machine ℓ :

The double unsigning move swapping the leftmost entries is now a move for Machine ℓ, single unsigning moves are banned !

• Machine d :

The costs for double unsigning moves for Machine d go down by 1, hence it is equal to $i + j - 2$.

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block have an even number of negative entries, while a **type B decomposition** does not.

If $w = \bar{2}134\bar{5}\bar{7}\bar{8}6$, then
$w = \bar{2}134\bar{5} \oplus \bar{2}\bar{3}\bar{1}$ is the **type D decomposition**,
$w = \bar{2}1 \oplus 1 \oplus 1 \oplus \bar{1} \oplus \bar{2}\bar{3}\bar{1}$ is the **type B decomposition**.

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block have an even number of negative entries, while a **type B decomposition** does not.

If $w = \bar{2}1345\bar{7}\bar{8}\bar{6}$, then
$w = \bar{2}134\bar{5} \oplus \overline{23}1$ is the **type D decomposition**,
$w = \bar{2}1 \oplus 1 \oplus 1 \oplus \bar{1} \oplus \overline{23}1$ is the **type B decomposition**.

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Sum decompositions for D_n

For D_n, we need to distinguish between two types of sum decompositions. A **type D decomposition** requires that each block have an even number of negative entries, while a **type B decomposition** does not.

If $w = \bar{2}1345\bar{5}\bar{7}\bar{8}6$, then
$w = \bar{2}134\bar{5} \oplus \bar{2}\bar{3}1$ is the **type D decomposition**,
$w = \bar{2}1 \oplus 1 \oplus 1 \oplus \bar{1} \oplus \bar{2}\bar{3}1$ is the **type B decomposition**.

Define **oddness** in type D (denoted $o^D(w)$) as the number of type B blocks minus the number of type D blocks (so $o^D(w) = 3$).

Depth for an even signed permutation

Theorem [BBNW, 2015]
We have the following formula for depth for D_n

$$d(w) = \left(\sum_{w(i)>i} (w(i) - i) \right) + \left(\sum_{w(i)<0} |w(i)| \right) + \left(o^D(w) - neg(w) \right).$$

The D-oddness counts the "wasted" moves that are needed to join type B blocks so that we can perform the needed double unsigning moves.

Minimizing over products

Let (W, S) be a Coxeter group, and T its **set of reflections**

$$T := \{wsw^{-1} \mid s \in S, w \in W\}$$

We can rephrase the definition of $\ell(w)$ and $a(w)$ as

$$\ell(w) = \min\{k \in \mathbb{N} \mid w = s_1 \cdots s_k \text{ for } s_i \in S\}$$

and

$$a(w) = \min\{k \in \mathbb{N} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\}$$

Depth in terms of roots

Let $\Phi = \Phi^+ \cup \Phi^-$ be the **root system** for (W, S).
The depth $dp(\beta)$ of a positive root $\beta \in \Phi^+$ is defined as

$$dp(\beta) = \min\{k \mid s_1 \cdots s_k(\beta) \in \Phi^-, s_j \in S\}.$$

There is a bijection between positive roots and reflections, and denote by t_β the reflection corresponding to the root β.
For any $w \in W$ Petersen and Tenner defined

$$d(w) = \min\left\{\sum_{i=1}^{k} dp(\beta_i) \mid w = t_{\beta_1} \cdots t_{\beta_k},\ \ t_{\beta_i} \in T\right\}.$$

Algebraic meaning and algebraic motivation

Since for any reflection one has

$$d(t_\beta) = dp(\beta) = \frac{1+\ell(t_\beta)}{2}, \quad \text{(these are the costs of the machines } d\text{)}$$

then

$$d(w) = \min\left\{\sum_{i=1}^{k} \frac{1+\ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T\right\}.$$

Algebraic meaning and algebraic motivation

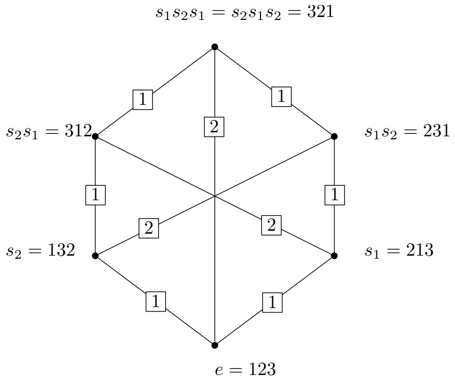
Since for any reflection one has

$$d(t_\beta) = dp(\beta) = \frac{1+\ell(t_\beta)}{2}, \quad \text{(these are the costs of the machines } d\text{)}$$

then

$$d(w) = \min\left\{ \sum_{i=1}^{k} \frac{1+\ell(t_i)}{2} \mid w = t_1 \cdots t_k \text{ for } t_i \in T \right\}.$$

Undirected paths in the weighted Bruhat order



BRUHAT GRAPH OF S_3

This means that the depth of w is equal to the minimal cost of an undirected path going from e to w in the Bruhat graph of W where each edge is labeled by

$$t \longrightarrow (1 + \ell(t))/2$$

Increasing paths, reduced factorizations, and weak order

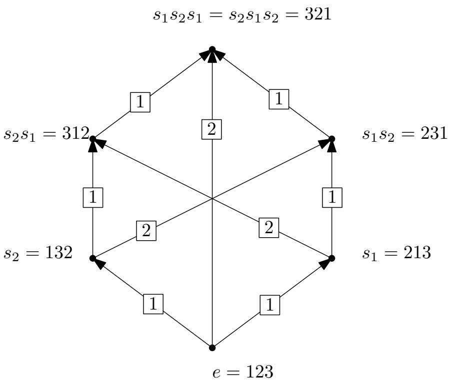
Our algorithms provide factorizations

$$w = t_1 \cdots t_k \text{ such that } d(w) = d(t_1) + \cdots + d(t_k)$$

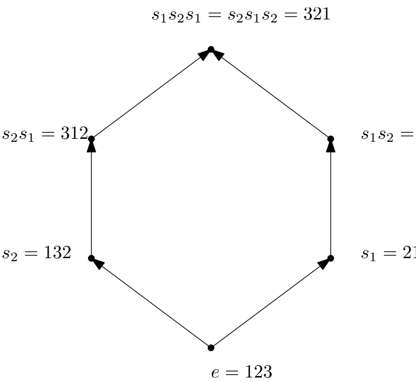
with the properties that:

- $\ell(w) = \ell(t_1) + \cdots + \ell(t_k)$. When this happens we say that the depth is **realized by a reduced factorization**.
- Hence we can restrict our checking only to **increasing paths** in the Bruhat graph.
- Moreover $e \prec t_1 \prec t_1 t_2 \prec \ldots \prec t_1 t_2 \cdots t_k$, where $\prec$ denotes the **weak Bruhat order**.

Directed paths in the weighted Bruhat order



DIRECTED BRUHAT GRAPH OF S_3



WEAK BRUHAT ORDER OF S_3

Reduced reflection length

Define the **reduced reflection length** $a'(w)$ as

$$a'(w) = \min\left\{ k \in \mathbb{N} \mid w = t_1 \cdots t_k \text{ for } t_i \in T \text{ with } \ell(w) = \sum_{i=1}^{k} \ell(t_i) \right\}$$

Since in classical Coxeter groups, depth can always be realized by a reduced factorization, we have

$$d(w) = \min_{t_1 \cdots t_k} \frac{\sum_i 1 + \ell(t_i)}{2} = \frac{a'(w) + \ell(w)}{2}.$$

Reduced reflection length

Define the **reduced reflection length** $a'(w)$ as

$$a'(w) = \min \left\{ k \in \mathbb{N} \mid w = t_1 \cdots t_k \text{ for } t_i \in T \text{ with } \ \ell(w) = \sum_{i=1}^{k} \ell(t_i) \right\}$$

Since in classical Coxeter groups, depth can always be realized by a reduced factorization, we have

$$d(w) = \min_{t_1 \cdots t_k} \frac{\sum_i 1 + \ell(t_i)}{2} = \frac{a'(w) + \ell(w)}{2}.$$

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem [BBNW, 2015]
$d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n, B_n, and D_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem [BBNW, 2015]
$d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n, B_n, and D_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Comparing length and depth

An element in a Coxeter group (W, S) is **short-braid-avoiding** if no reduced decomposition (product of simple reflections realizing w) has a consecutive subexpression $s_i s_j s_i$, with $s_i, s_j \in S$.

Theorem [BBNW, 2015]
$d(w) = \ell(w)$ if and only if the depth of w is realized by a reduced factorization and w is short-braid-avoiding.

Since the depth is always realized by a reduced factorization in S_n, B_n, and D_n, this shows that $d(w) = \ell(w)$ in those groups if and only if w is short-braid-avoiding.

Short-braid-avoidance in B_n and D_n

For permutations, this reproves the Petersen–Tenner theorem that $d(w) = \ell(w)$ if and only if w is **fully commutative**, which is characterized by Billey-Jockusch-Stanley avoiding 321.

In B_n, short-braid-avoiding is equivalent to Stembridge's notion of **fully commutative top-and-bottom**, which is characterized by avoiding $1\bar{2}$, $\bar{1}\bar{2}$, $\bar{2}\bar{1}$, $\bar{3}21$, $\bar{3}2\bar{1}$, and 321

In D_n (and any simply-laced group), short-braid-avoiding is equivalent to being **fully commutative**, which is characterized by Billey-Postnikov avoiding 321. (This is avoiding 321 as a permutation of $\{\pm 1, \ldots, \pm n\}$, not allowing the simultaneous use of opposite entries.)

Achieving the lower bound

The elements for which $a(w) = d(w)$ (and hence both are equal to $\ell(w)$) are the **boolean elements**, where no reduced decomposition has any simple reflection more than once. These are characterized by avoiding 10 patterns for B_n and 20 for D_n (Tenner).

The more general question of when $d(w) = (a(w) + \ell(w))/2$ seems hard and is not characterized by pattern avoidance.

Problems

- How many elements of B_n and D_n have depth k ?
- Find the generating function for depth in B_n or D_n (See Guay-Paquet–Petersen for S_n)
- Characterize depth for affine Coxeter groups.
- Is depth realized by a reduced factorization into reflections for all elements in all Coxeter groups?
- Is there a characterization or a formula for the reduced absolute length $a'(w)$ for general Coxeter groups ?

Thank you

Thank you for your attention!