Last meeting: obtained all FC elts in a Coxeter gp W (S_4)

by

generating all words up to max length

$\downarrow$ isFC test

FC words

$\downarrow$

FC elts

Next?: generate FC elts in the first place instead of using a filtering test.

$\downarrow$

Today: Preparation for this.

1. Descents $\left((W,S) \text{ a Coxeter system.} \right)$

Facts: (1). $\forall w \in W$, $s \in S$, $\ell(ws) \in \{\ell(w)+1, \ell(w)-1\}$.

e.g. A_5, $\ell(\underbrace{s_1 s_2}_{w}) = 2$ $\quad \ell(s_1 s_2 s_3) = 3$

$\ell(s_1 s_2) = 2 \quad \ell(s_1 s_2 s_1) = 3$

$\ell(s_1 s_2) = 2 \quad \ell(s_1 \underline{s_2} \underline{s_2}) = \ell(s_1) = 1$

$\ell(\underbrace{s_1 s_5 s_4}_{w}) = 3$

$s = s_1 \quad \ell(ws) = 2$

$s = s_2 \quad \ell(ws) = 4$

$s = s_3 \quad \ell(ws) = 4$

$s = s_4 \quad \ell(ws) = 2$

$s = s_5 \quad \ell(ws) = 4.$

Similarly, $\ell(sw) \in \{\ell(w)+1, \ell(w)-1\}$.

Notation: We'll write $ws < w$ if $\ell(ws) = \ell(w) - 1$

and $ws > w$ if $\ell(ws) = \ell(w) + 1$.

"$<$", "$>$" are part of the so-called bruhat order on W.

(2). We have $sw < w$ iff w has a reduced word where s is the leftmost letter.

the obvious right-side analogs exist

In this case, we say s is a left descent of w.

Rmk: "if" is obvious, the 'only if' is the nontrivial part.

The left descent set of w is the set of all its left descents, i.e.,

the set $\mathcal{L}(w) = \{ s \in S : sw < w \}$.

2. Descent sets of FC elts

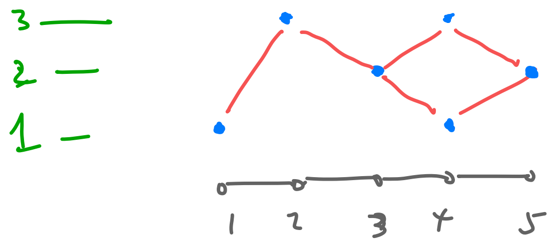
<u>Prop.</u> Let s, t be adjacent generators of W, i.e., $s, t \in S$, $m(s,t) \geq 3$. Let w be an FC elt in W. Then in all red words $\underline{w}$ of w, the subexpression of $\underline{w}$ obtained by deleting all letters not equal to s or t are all the same, i.e., does not depend on $\underline{w}$.

In particular, if $s \in \mathcal{L}(w)$, $t \notin \mathcal{L}(w)$.

<u>Corollary</u> (immediate) If $w \in W$ is FC, $\mathcal{L}(w)$ consists of pairwise commuting generators. i.e., $\forall s, s' \in \mathcal{L}(w)$, $s \neq s' \implies m(s,s') = 2$.

e.g. A_5.

$$w = \cancel{s_1}\, \underline{s_4\, s_3}\, \cancel{s_5}\, \underline{s_4}\, \cancel{s_2} \,. = s_1\, \underline{s_4}\, \underline{s_3}\, s_2\, s_5\, \underline{s_4}$$



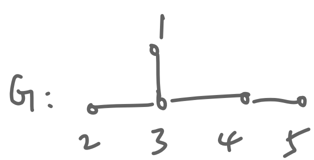
remark unrelated to the prop.

while every point a 'lowest' level, the heap is <u>not ranked</u> in the sense that $\nexists$ a function r to $\mathbb{Z}$

s.t. $x < y \Rightarrow r(y) = r(x) + 1$.

if $s = s_4$, $t = s_3$. then $w_{\{s_4, s_3\}} = s_4\, s_3\, s_4$.

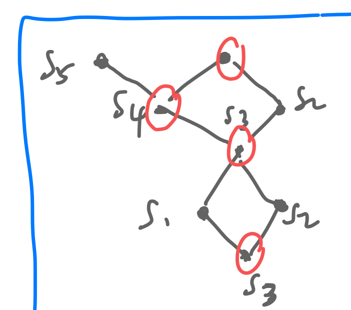
D_5



$w = s_3\, s_1\, s_2\, s_3\, s_4\, s_5\, s_2\, s_3 \rightarrow$

if $s = s_3$, $t = s_4$.

$w_{\{s_3, s_4\}} = s_3\, s_3\, s_4\, s_3$



Pf of the prop. Since w is FC, no red word of it has a contiguous subword $sts\ldots$ or $tst\ldots$ of length $m(s,t)$, so we can never change the order in which s and t appear. $\square$.

Another corollary: (Easy descent criterion for FC elts). Let $w \in W$ be FC and let $s \in S$. Then $s \in \mathcal{L}(w)$ iff in some (equiv, every) reduced word $\underline{w}$ of w, we have

checking one red word is enough!

(1) s appears in $\underline{w}$

(2) no letter in $\underline{w}$ to the left of the leftmost s is adjacent to s.

no need to discuss again

e.g. A_5, $w = s_1 s_4 s_3 s_5 s_4 s_2$. then $\mathcal{L}(w) = \underline{\{s_1, s_4\}}$. they commute

↓

FC

"left_des(w, M)"

Todo: write a function that computes the left descents of an FC elt.

input: w, an FC elt + M, the Coxeter matrix.

descent test "is_left_des_of(s, w, M)"

e.g. A_5 $w = s_1 s_2 s_1 \Longrightarrow \mathcal{L}(w) = \{s_1, s_2\}$

$v = s_2 s_1 s_3 s_2 s_3 s_4 s_1 \Longrightarrow s_1 \in \mathcal{L}(w)$

3. Canonical word of an FC elt. w

① from heaps. — consider the 'level' of each letter (see def), with the lowest level being 'Level 1'.

e.g. A_5. $w = s_1 s_4 s_3 s_5 s_4 s_2$

+1 +1 $\max\{2,3\} = 3$

1 1 2 2 3 3

+1 +1

Note: the level-1 gen. are exactly the left descents!

why ① and ② are equiv

$w = 1\,4\,3\,5\,2\,4$

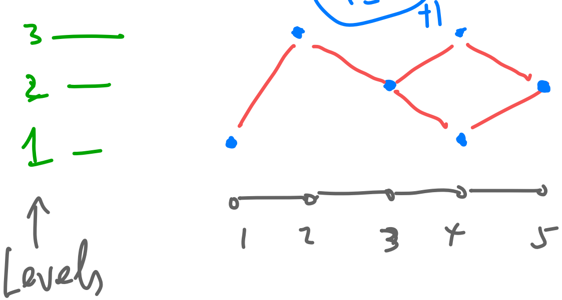
Possible ToDo:

write a canonical form function

can_word_heap(w, M)

that uses this idea.

(need to implement 'level')



— Canonical word: 'read the levels from bottom to top; for each level, arrange the generators according to their lex. order'.

② Use descents: (a) start with w. get $\mathcal{L}(w)$ (Level 1) and form the product W_1 of the elts in $\mathcal{L}(w)$ (lex.) and move it to the front of

$$w \longmapsto w = W_1 w'$$

(b) Iterate

e.g. A_5. $w = s_1 s_4 s_3 s_5 s_4 s_2 \rightarrow \mathcal{L}(w) = \{s_1, s_4\}$.

$$w = \underbrace{(s_1 s_4)}_{w_1}\big(\underbrace{s_3 s_5 s_4 s_2}_{w'}\big)$$

$$= (s_1 s_4)(s_3 s_5)(s_4 s_2)$$

$$= (s_1 s_4)(s_3 s_5)(s_2 s_4)\underline{(\quad)}$$

DONE!

ToDo: Code this. 'can_word_des(w, m)'.

In both cases, the canonical word is called the

Cartier-Foata form of w.

Rmk: Having canonical words of FC elts allows us to tell if two red reduced words express the same elt: they do iff they have the same canonical form.

Q: Say we have an FC elt w in a Coxeter gp W (the matrix M is known). Let $s \in S$. Under what conditions is sw longer than w and still FC?

$\Updownarrow$

$s \notin \mathcal{L}(w)$

??