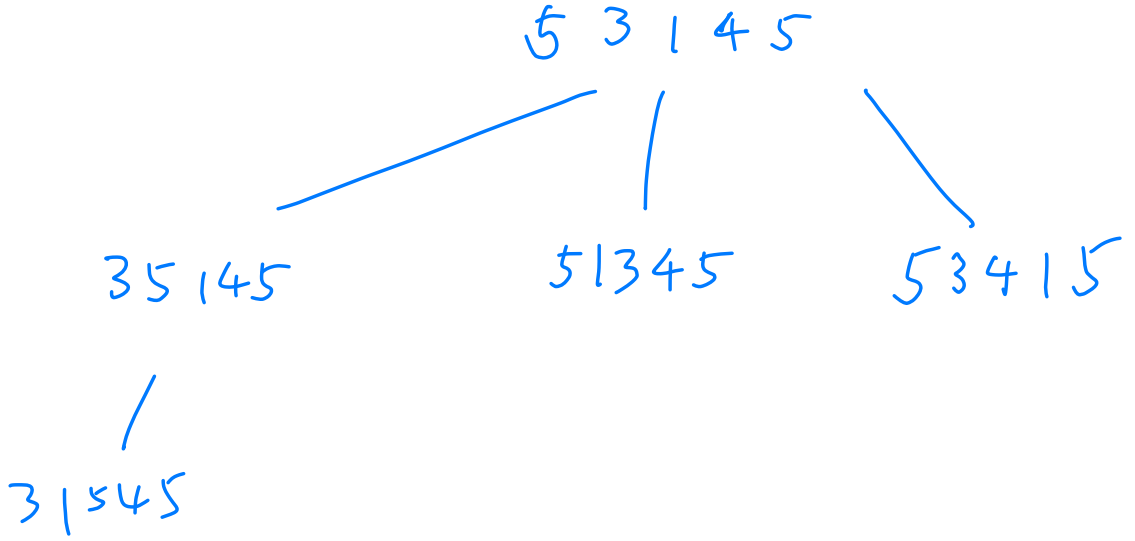
is_FC- breadth , $\underbrace{(5,3,1,4,5)}_{w}$. State

Line	
55-56	no long braid in w, so move in
58	checked = {w} queue = [w]
59	queue is not empty. So go on
60	word = queue.pop( ) → word = w → rightmost in queue; queue = [ ]
61-70	for i ... : commute neighbors in word wherever possible. once a truly new word is generated, immediately check for long braids. If a long braid is present, done; else, add the truly new word to checked, and put the new word to the very left of the queue.

What should happen.



What happens in the code.

58 | word = 53145, queue = [ ]

word = 53145 queue = [ ].

61-70:

i = 0. new_word = 35145, no long braid.

checked = {w, 35145}

queue = [35145,]

i = 1 new_word = 51345, no long braid.

checked = {w, 35145, 51345}

queue = [51345, 35145]

i = 2 new_word = 53415, no long braid

checked = {w, 35145, 51345, 53415}

queue = [53415, 51345, 35145].

queue = [53415, ... , ...]

59 queue not empty, so run 60- 70 again.

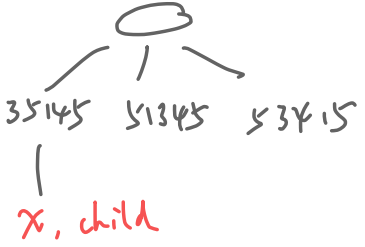
60 word = queue.pop( ) → word = 35145 → rightmost
queue = [53415, ...]

61-70 i = 1. (truly) new_word = 3|545. long braid. Done.

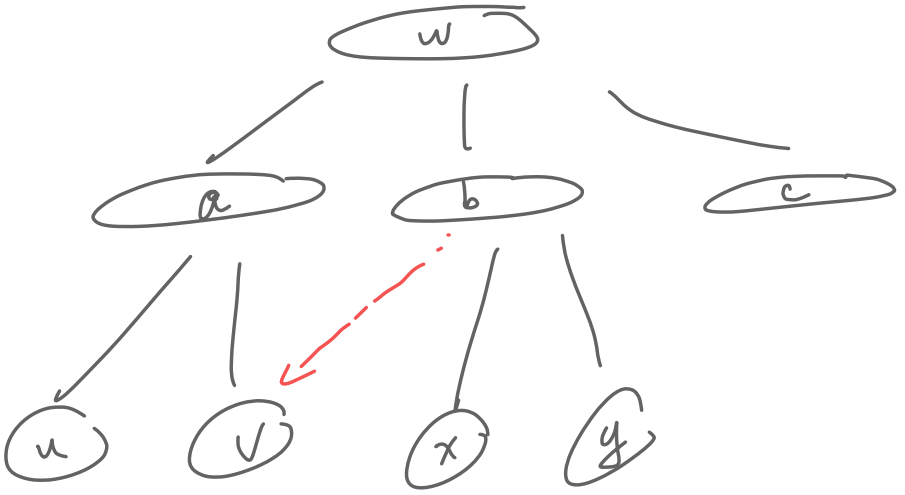
But, if (for all truly) new_word = x (child of 35145) has no long braid

then checked.add(x).

queue = [x, 53415, 51345].
child



Ex. If the commutation class of w 'looks like'



where all words contain no long braid. How does the program run. How does the 'depth' code run?

Done with FC_tests.

Next ? :

Given the Coxeter type. eg. A_4: 

generate all FC words up to a certain length l.

$l = 2$. all ij where $i \neq j$ should be generated.

$l = 3$. more interesting. for example

1 3 1 should not be an output

1 2 1 , 2 3 2 should not be output

$l = 4$ more complicated.

Naive idea: A

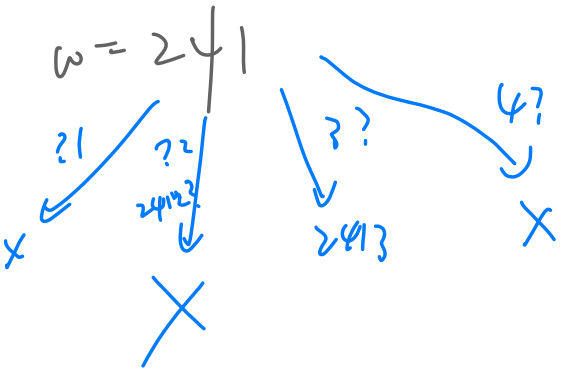
generate all words up to length l, then check FC. A_4

but can we do better?

Rough Idea. Maybe each admitted word w can be viewed as a node on a graph and it leads to all other admissible words obtained by adding one letter to the right of w?

B

eg



Algo. Todo: check if a word is reduced → didn't matter much for FC test may matter more now.

More on reducedness / FC test.

C. FC test + checking for 'ss' $\stackrel{?}{=}$ Reduced + FC test

ToDo: – Sarah : C

– All : A+B . pretend we already have is_reduced test if never say.