Denis R[redacted]

COMP 330 - Assignment 5

Q1) L1 is context free and this is a grammar for it:

$$L_1 = \{ a^m b^n k^k d^j \mid m,n,k,j \geq 0 \text{ and } m=n \text{ and } k=j \}$$

$$\Sigma = \{a, b, c, d\}$$

$$V = \{S, A, C\}$$

$$S = AC \mid \varepsilon$$

$$A = aAb \mid \varepsilon$$

$$C = cCd \mid \varepsilon$$

$$L_2 = \{ a^m b^n k^k d^j \mid m,n,k,j \geq 0 \text{ and } m = k \text{ and } n = j \}$$

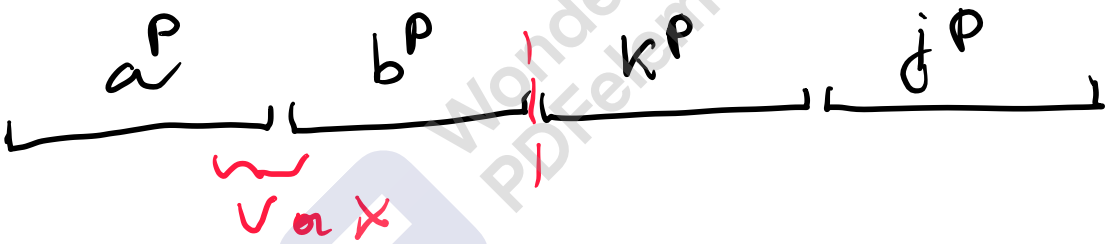
L2 is not context free. Proof by pumping lemma.

Demon chooses p>0

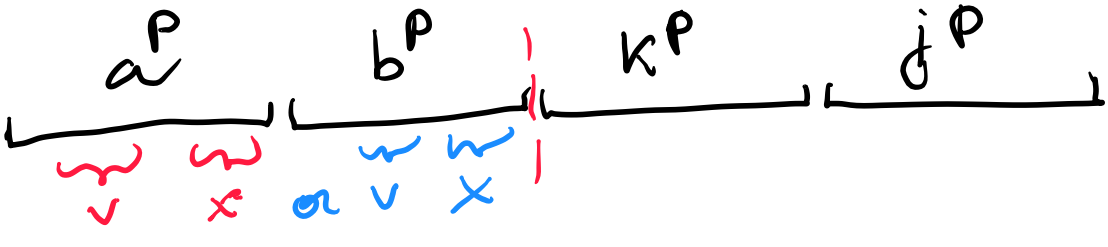
I choose a^p b^p k^p d^p

How can the demon break up the string into u, v, w, x, y parts:

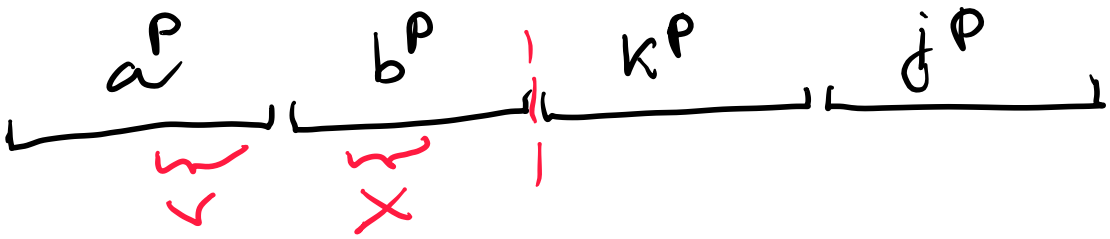
A)v or x straddles the a letter boundary. Let i=2 and the count of a's != k's and b's != j's.




B)v and x are is the same block. Let i=2 then either the count of a's != k's or b's != j's.



C)v and x are in different letter blocks. Let i=2 then both the count of a's != k's and b's != j's.




Therefore L2 is not context free.

Q2-Part 1) Consider L:

$$L = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i \neq j \text{ or } j \neq k \}$$

Here is the grammar for L:

$$\Sigma = \{ a, b, c \}$$

$$V = \{ S, A, A_1, A_2, B_1, B_2, C \}$$

$$S = aA_1C \mid A_2bC \mid AbB_1 \mid AB_2c$$

$$A_1 = aA_1 \mid aA_1b \mid \varepsilon$$

$$A_2 = A_2b \mid aA_2b \mid \varepsilon$$

$$B_1 = bB_1 \mid bB_1c \mid \varepsilon$$

$$B_2 = B_2c \mid bB_2c \mid \varepsilon$$

$$C = Cc \mid \varepsilon$$
$$A = Aa \mid \varepsilon$$

The non-terminal symbol S has 4 rules: from left to right they allow for a's>b's + arbitrary c's; a's <b's + aribitrary c's; arbitrary a's + b's>c's; arbitrary a's + b's <c's.

A1 only allows for increasing a's or a's and b's. Since we go in with 1 more a then b already we are guaranteed to have a's >b's. A2 does the same for a's < b's. B1 enforces more b's than c's similairly to A1 and B2 mirrors A2's behavior, keeping b's < c's. The symbols A and C allow to add arbitrary a's and c's respectively to branches where their letter count is irrelevant.

Thus L is context-free.

Q2-Part 2)

Take L complement.

$$\overline{L} = \{a^i b^j c^k \mid i,j,k \geq 0 \text{ and } i = j \text{ and } j = k\}$$

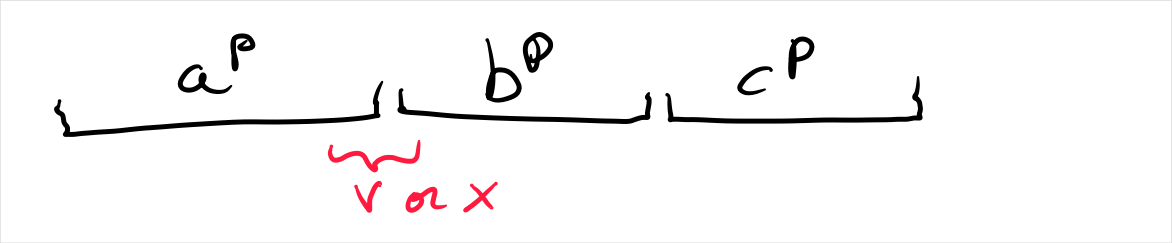
We will show this is not context free by the pumping lemma.

Demon choose p>0.

I choose a^p b^p c^p

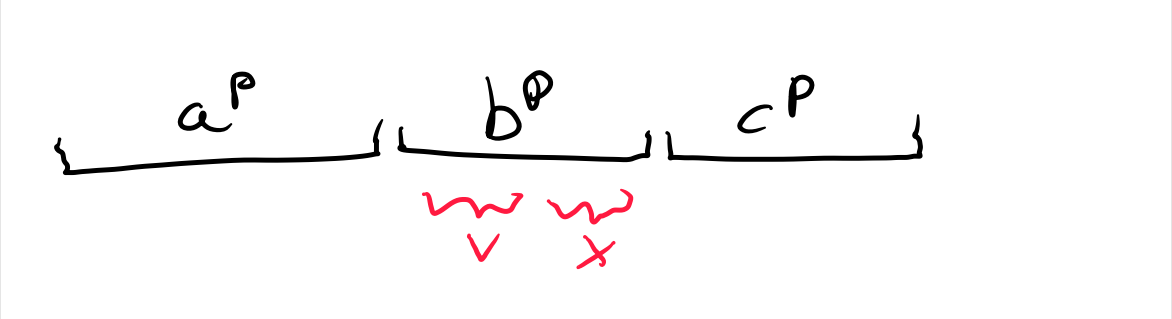
Have to analyze how the demon might break up the string into u, v, w, x, y:

A) v or x crosses a boundary between 2 blocks. Then let i=2 and number of a's != b's




B) v and x are both in the first block of a's and contain only a's. Then let i=2 and the 3 blocks don't have the same length anymore.


C) Same argument as B) but v and x are both in the second block and contain only b's.



D) v contains only a's and x only b's (i.e. both are in separate blocks), thus one block is untouched by the pumping. Let i=2 and again the blocks don't have the same length anymore.

Therefore by the pumping lemma we have shown L complement is not context-free and thus L is not deterministic context-free.

Q3) Let L=(abc)*. This is a regular language and thus a CFL too.

Take perm(L) which effectively shuffles the a's, b's and c's in all the different possible orderings. Clearly no matter the ordering #a's = b's = c's.

Recall that the intersection of 2 CFLs must be a CFL too.

Let L2 = a*b*c* This is clearly regular and a CFL. perm(L) intersection L2 = { a^n b^n c^n | n >=0 }. However we have proved in class that this isn't a CFL therefore either perm(L) or L2 must not be CFLs. The only option is for perm(L) to not be a CFL.

Q4)

$$L_2 = \{ a^{i+j} b^{j+k} c^{k+l} d^{i+l} \mid i, j, k, l \geq 0 \}$$

$$\Sigma = \{a, b, c, d\}$$

$$V = \{S, A, B, C\}$$

$$S = aSd \mid ABC \mid \varepsilon$$

$$A = aAb \mid \varepsilon$$

$$B = bBc \mid \varepsilon$$

$$C = cCd \mid \varepsilon$$

Q5) Let L be the language that recognizes palindromes. This language is a CFL given by this grammar:

$$\Sigma = \{a_1, a_2, \ldots, a_n\}$$

$$S = \{S, A_1, A_2, \ldots A_n, B\}$$

$$S = A_1 \mid A_2 \mid \ldots \mid A_n \mid B \mid \varepsilon$$

$$A_1 = a_1 S a_1$$

$$A_2 = a_2 S a_2$$

$$\vdots$$

$$A_n = a_n S a_n$$

$$B = a_1 \mid a_2 \mid \ldots \mid a_n$$



Since L is a CFL then there exists a PDA for it. Take this PDA and change all the original accept states to reject states and vice versa. Now the machine accepts the opposite of what it used to, the complement of the set of palindrome or in other words the non-palindromes. This happens because in the original machine every reject state represented a non-palindrome. Since all these states are accept states then all non-palindromes are accepted. Conversely the original accept states which used to accept palindromes now reject them.
Therefore we have a PDA for the non-palindromes and it is a CFL.