Midterm Exam #2

INSTRUCTIONS:

- Put your NAME and SBU ID # on this exam booklet in the space provided.
- This is a CLOSED-BOOK exam, which TERMINATES AT 12:50PM (80 minutes). NO ELECTRONIC DEVICES, including calculators, may be used during the exam.
- Please place ALL ANSWERS IN THIS BOOKLET, on the sheet where the corresponding question is printed.
- THINK BEFORE YOU WRITE. A partial solution can get you partial credit, but too much extraneous information can prevent me from finding your correct solution.
- SOME QUESTIONS ARE HARDER THAN OTHERS, and you might not have time to answer all questions completely. LOOK OVER ALL THE QUESTIONS BEFORE STARTING, and work first on those that will get you the most credit fastest. Use the number of points listed for each question as a guide.

Question:	1	2	3	4	5	6	7	Total
Points:	10	10	10	10	10	10	20	80
Score:	10	10	10	16	10	10	13	73

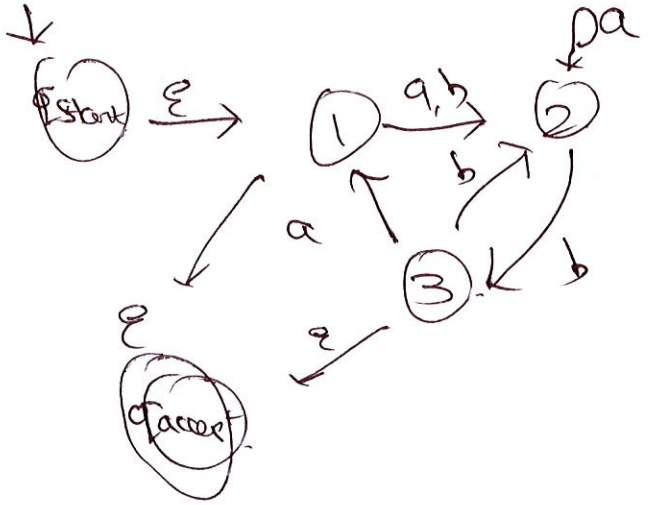
Note: Point values have been assigned so that you should expect to be answering roughly one point per minute.

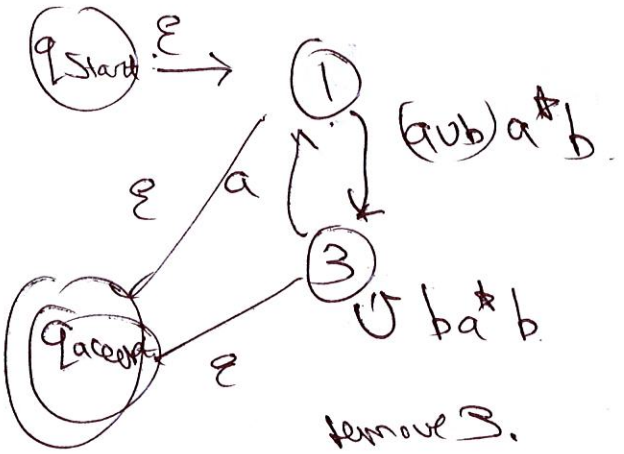
NAME and SBU ID#: Deep Diwakar 115060128.

$q_{rip} = 2$.

1-2-3

1-3-23.



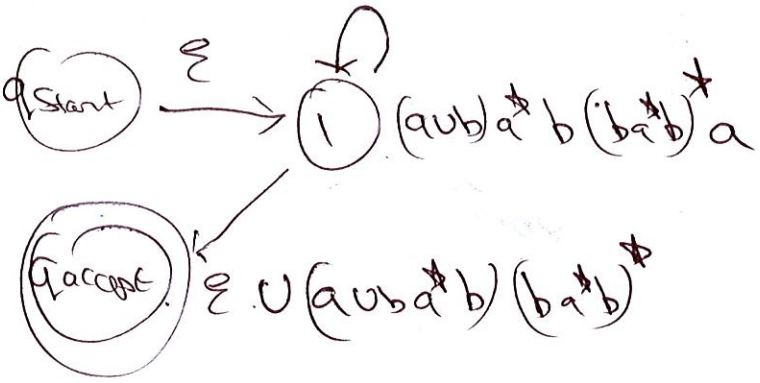


remove 3.

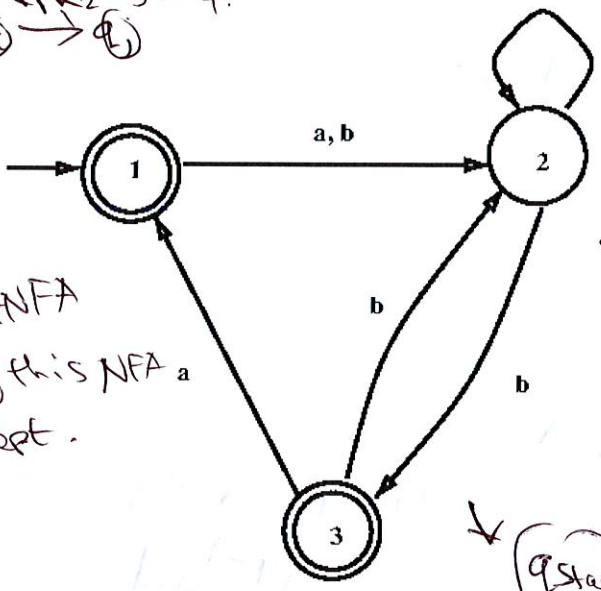
$q_{rip} = 3$.

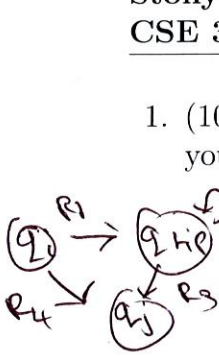
1-3-1

1-3-qaccept



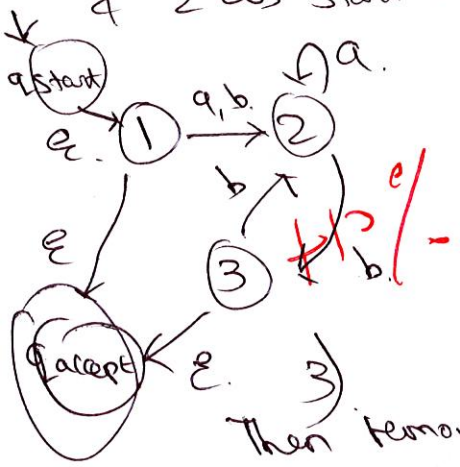
1. (10 points) Find a regular expression equivalent to the following NFA. For full credit, you must include some kind of plausible description of how you obtained your result.



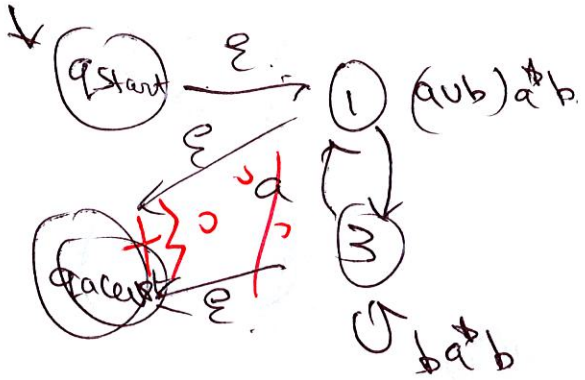


Seq to 

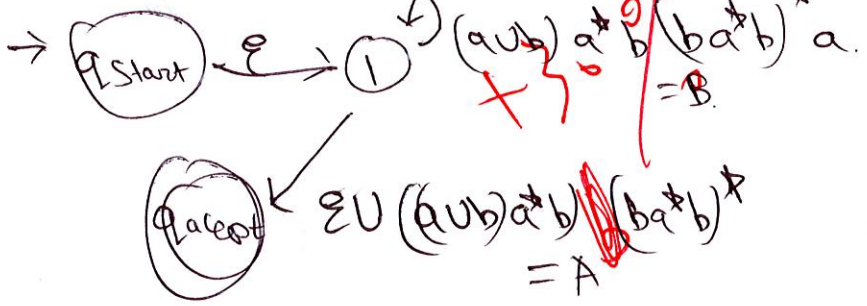
1) We will construct a GNFA with 5 states, 3 of this NFA & 2 as start & accept.



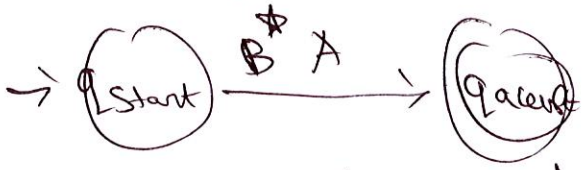
2) We will rip out states one after other. first we remove (2).



3) Then remove state (3). we will get.



4) Then remove (1) also.
we get.



where $B = (aUb)a^*b(ba^*b)^*a$

& $A = \varepsilon U (aUb)a^*b)(ba^*b)^*$

& the Reg ex is B^*A for the NFA.

Explanation

then 

$R_1 R_2^* R_3 \cup R_4$

NAME: Deven Diwakar.

$0^m 1^n \quad m \neq n.$

$0^p 1^{p+1}.$

$k > 0$

$0^p 1^{p+1}$: $0^p 1^p.$

~~$0^p 1^{p-1}$~~

~~$0^p 1^k$~~

$0^p 1^{p!+p}$

$x = 0^a \qquad a+b \leq p. \quad b \leq p.$

$y = 0^b \qquad (a+b)\ p! - p$ ~~$= a b$~~

~~$b!$~~ : ~~b~~ $\dfrac{p!}{b}$

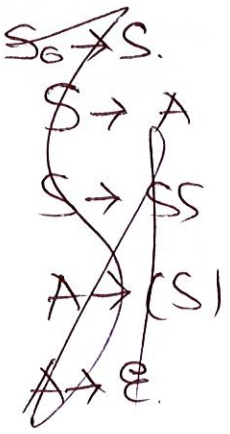
~~p~~

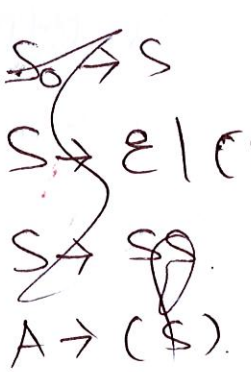
2. (10 points) Prove that the following language is not regular:

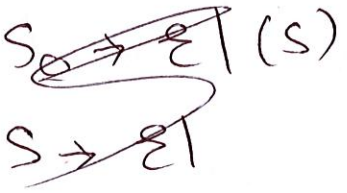
$$L = \{0^m 1^n \mid m \neq n\}.$$

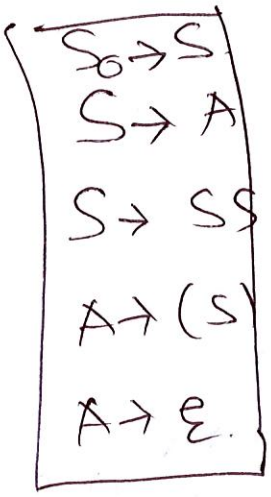
10

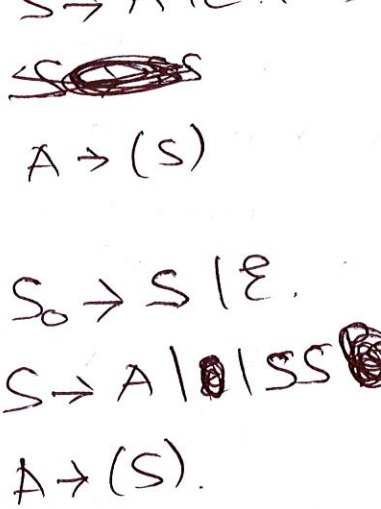
We will use Proof by contradiction.
Let L be regular ⇒ By pumping lemma we get p as pumping length.
Let $w = 0^p 1^{p!+p}$ Note $|w| \geq p$ & $m = p$ & $n = p! + p \Rightarrow m \neq n$.
Thus $w \in L$.
Thus we get x, y, z s.t $xy^i z \in L$ $i \geq 0$ &, $|xy| \leq p$ & $|y| > 0$.
Thus y must be of the form 0^b where $b \leq p$.
Now we will pump up string y to such a i s.t.
No of 0's = no of 1's. we need extra p! 0's for this to happen. with a single pump of y we add b 0's. Thus we need to pump it by $\frac{p!}{b}$ times which is an integer since $b \leq p$.
This makes no. of 0's = no of 1's & that string is not in L by def.
Thus we get a contradiction.
Thus L is not regular.











$S_0 \to \varepsilon \mid 1$

3. (10 points) Consider the following context-free grammar G:

10

$$\begin{aligned} S &\to A \\ S &\to SS \\ A &\to (S) \\ A &\to \epsilon \end{aligned}$$

Give a grammar G' that is equivalent to G, but which is in Chomsky Normal Form.

1) Add $S_0 \to S$

$S_0 \to S$
$S \to A \mid SS$
$A \to (S) \mid \varepsilon$

2) $A \to \varepsilon$ remove

$S_0 \to S$
$S \to A \mid SS \mid \varepsilon$
$A \to (S)$

remove $S \to \varepsilon$.

$S_0 \to S \mid \varepsilon$
$S \to A \mid SS \mid S$
$A \to (S)$

3) remove $S \to A$.

$S_0 \to S \mid \varepsilon$
$S \to SS \mid S \mid (S)$
$A \to (S)$

remove $S \to S$

$S_0 \to S \mid \varepsilon$
$S \to SS \mid (S)$
$A \to (S)$

Remove $S_0 \to S$

$S_0 \to SS \mid (S) \mid \varepsilon$
$S \to SS \mid (S)$
$A \to (S)$

4) Convert remaining

$L \to ($ $U_1 \to LS$
$R \to)$.

$S_0 \to SS \mid U_1 R \mid \varepsilon$
$S \to SS \mid U_1 R$
$A \to U_1 R$
$L \to ($
$U_1 \to LS$
$R \to)$

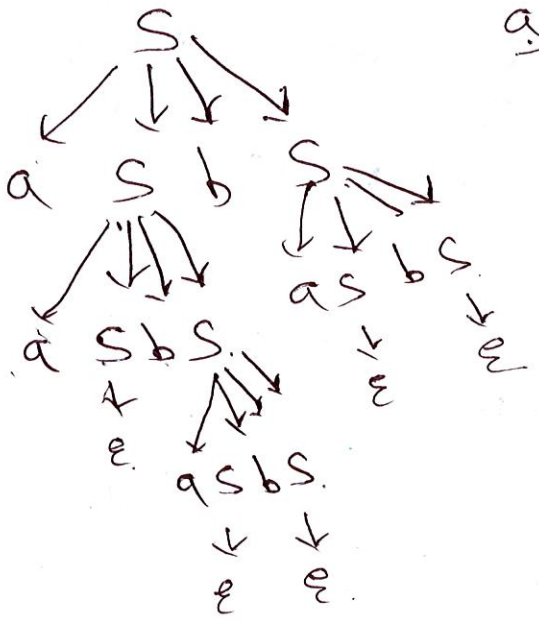
This is in Chomsky Normal Form.

$S \to \varepsilon \mid aS \mid aSbS$ aabbab

$S \Rightarrow aSbS \Rightarrow aaSbSbS$ aabababb

$\Rightarrow aabSbS \Rightarrow aabaSbSbS$

aaabaSbSbS



4. Consider the following CFG: 10

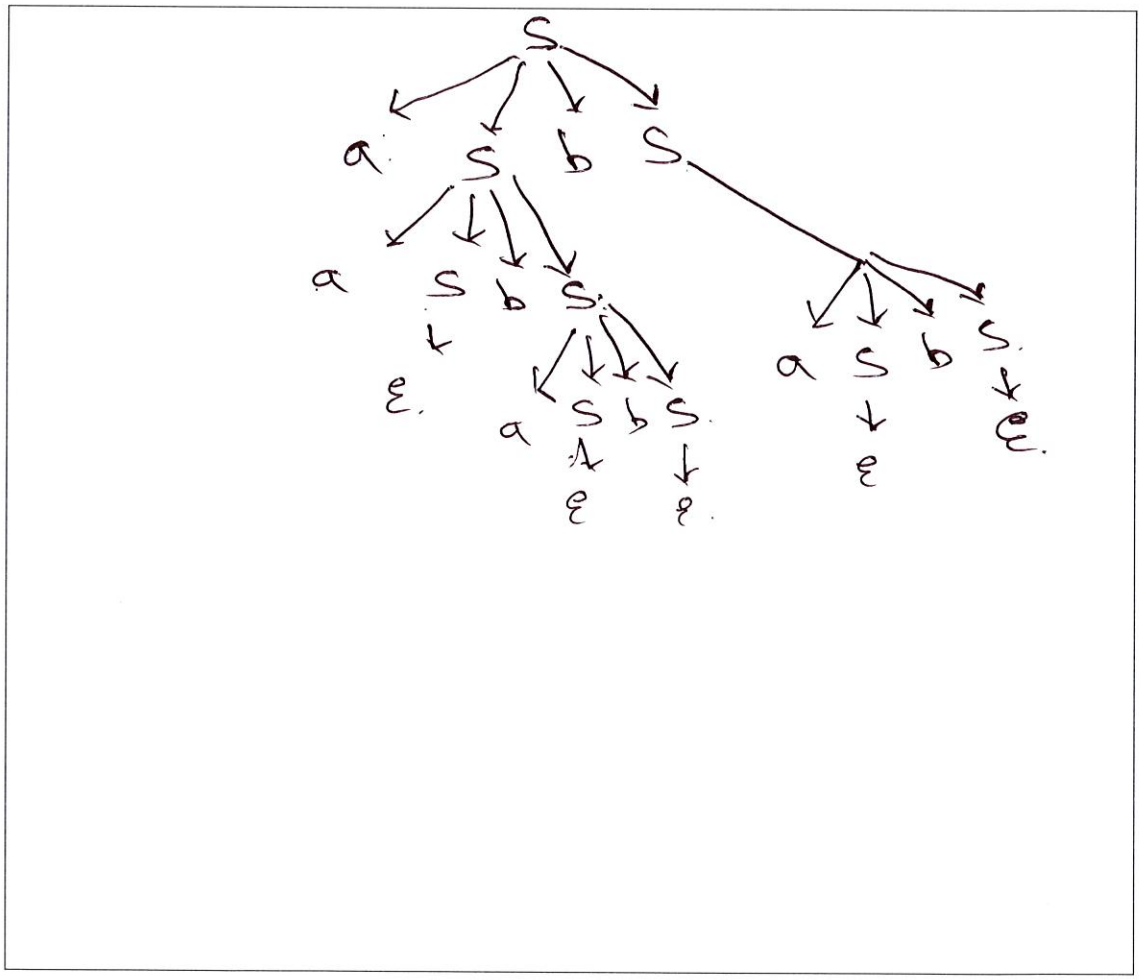
$$S \rightarrow \epsilon \mid aS \mid aSbS$$

(a) (4 points) Give a *leftmost* derivation for the following string: *aababbab*.

$S \Rightarrow aSbS \Rightarrow aaSbSbS \Rightarrow aabSbS \Rightarrow aabaSbSbS \Rightarrow aababSbS \Rightarrow aababbS \Rightarrow aababbaSbS \Rightarrow aababbabS \Rightarrow aababbab$

Note we only used $S \rightarrow \varepsilon \mid aSbS$ since #a = #b
$S \rightarrow aS$ will increase #a. 4

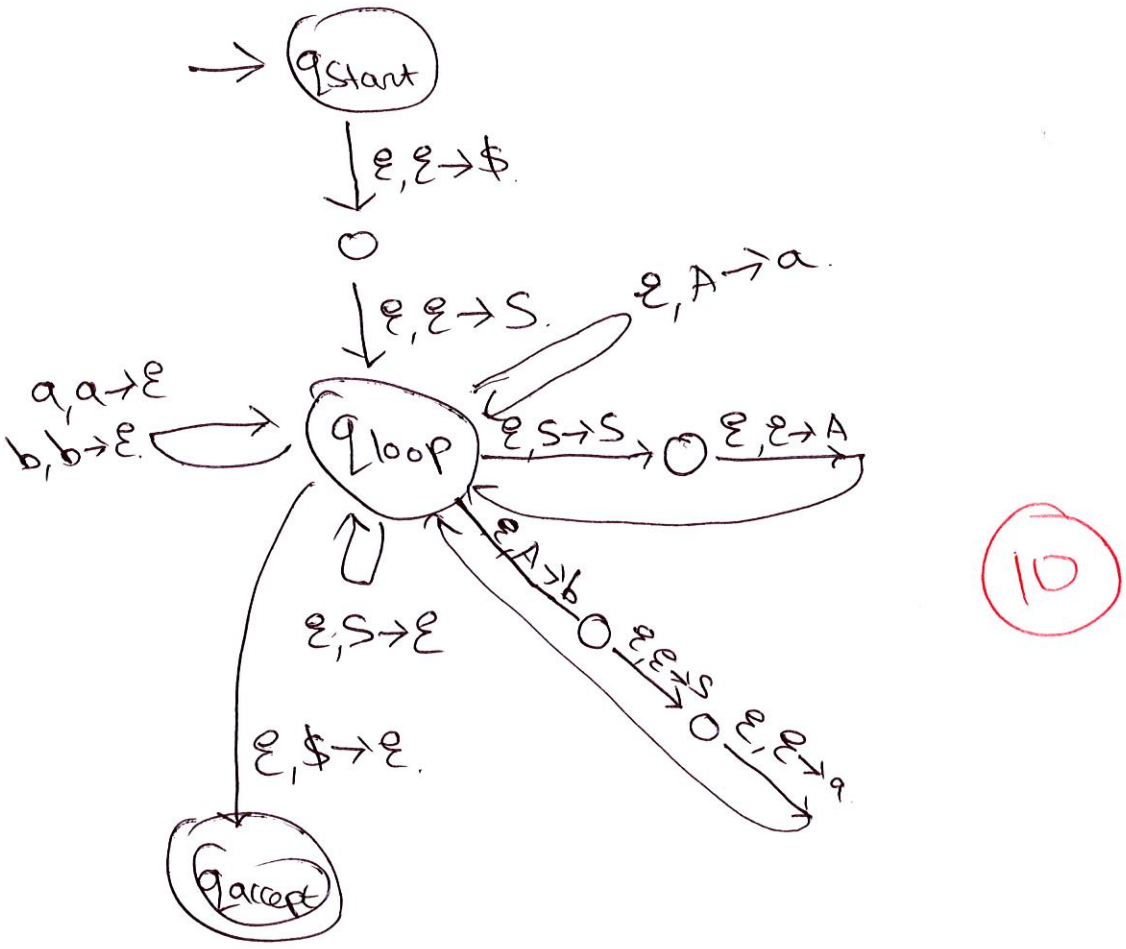
(b) (6 points) Draw the parse tree corresponding to your derivation:



6

5. (10 points) Draw the transition diagram for a pushdown automaton that recognizes the language generated by the following grammar:

$$S \rightarrow \epsilon$$
$$S \rightarrow AS$$
$$A \rightarrow aSb$$
$$A \rightarrow a$$



10

a, ε → a
b, ε → b
c, ε → c
ε, $ → ε
ε, ε → ε
ε, ε → ε
ε, ε → ε

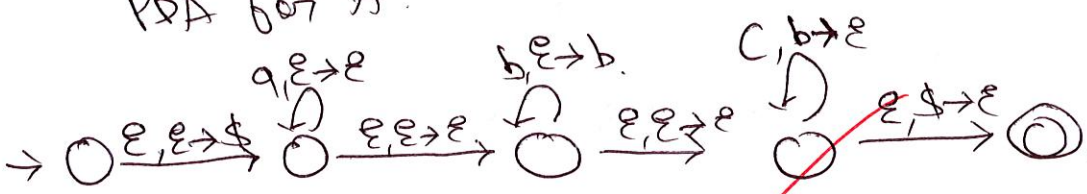
6. Example 2.36 in Sipser shows that the language $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free.

   (a) (4 points) Use the languages $A = \{a^m b^n c^n \mid m, n \geq 0\}$ and $B = \{a^n b^n c^m \mid m, n \geq 0\}$ together with Sipser Example 2.36 to show that the class of context-free languages is not closed under intersection.

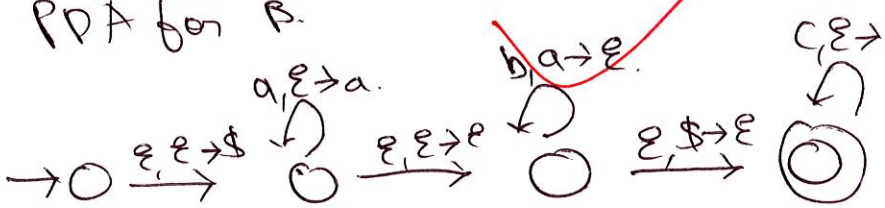
4

We have A & B as context free languages.

PDA for A.



PDA for B.



We have $L = \{a^n b^n c^n \mid n \geq 0\}$ which is A∩B. Because $m = n$. If class of context free languages were closed under intersection then L would be CFL. But Ex 2.36 shows it is not. Here class of CFL is NOT closed under intersection

$A \wedge B \equiv \overline{\bar{A} \cup \bar{B}}$.

$A = 1234$ 1-10

$B = 23$ 5-10.

$\bar{B} = 1, 4, 5, 6, 7, 8, 9, 10$

1, 4, 5, 6, 7, 8, 9, 10

(b) (6 points) Use part (a) and DeMorgan's law to show that the class of context-free languages is not closed under complementation.

We have $A \cap B = \overline{\bar{A} \cup \bar{B}}$

Let class CFL be closed under complementation. We will show contradiction

Let A & B be as in part A.

A is CFL $\Rightarrow$ $\bar{A}$ is CFL; B is CFL $\Rightarrow$ $\bar{B}$ is CFL

$\bar{A}$, $\bar{B}$ both CFL $\Rightarrow$ $\bar{A} \cup \bar{B}$ is CFL since class of CFL is closed under union. 6

Thus $\overline{\bar{A} \cup \bar{B}}$ is also CFL since ~~CFL~~ class of ~~complementation~~ CFL is closed under complementation.

But By De Morgans Laws $A \cap B = \overline{\bar{A} \cup \bar{B}}$.

So ~~it~~ implies $A \cap B$ is CFL. which is not the case as per Ex2.36.

Thus class of Context Free languages ~~can't~~ is not closed under complementation.

Ind. $k=0$ 11 00

$k=1$ ww

$k-1$

$w = ya$ $|y| = k$

yayaya. y cannot exist i.

$|y| = k$. $k-1$. No DFA of length $k-1$ can accept yy $|y| = k$.

$w = ya$ yaya k

$w = 0^p 1 0^p 1$

7. Let $\Sigma = \{0, 1\}$. Let $WW_k = \{ww \mid w \in \Sigma^*$ and w is of length $k\}$.

   (a) (10 points) Show that for each k, no DFA with fewer than k states can recognize WW_k.

~~WW_k is not a regular language only. So no DFA can recognize it.~~

We will prove by induction.

Base Case $k=1$. Let $w = 0 \Rightarrow ww = 00$. We can't have a DFA with $k-1 = 0$ states recognize 00.

General: Let WW_k be not recognized by all DFAs of length $k-1$ & below. This is good, but induction is not required

Proof:- Let $w = w_1 \dots w_{k+1}$. Let $\exists$ a DFA of k states, which recog WW_{k+1} $q_0 \dots q_{k-1}$

Since we have $k+1$ input it will repeat somewhere. by Pigeonhole Principle. Let q_i & q_j be states with repeat when w is fed into DFA. & let $\delta(q_{i-1}, w_i) = q_i$ & $\delta(q_{j-1}, w_j) = q_j$. Thus if we use string $w_1 \dots w_i w_j \dots w_{k+1}$ is a string which should also be accepted. $\therefore$ it is less than k. But this contradicts our ind. Hyp.

Induction is not really used

$w = w_1 w_2 \dots w_i \underbrace{w_{i+1} \dots w_j} w_{j+1} \dots w_{k+1}$

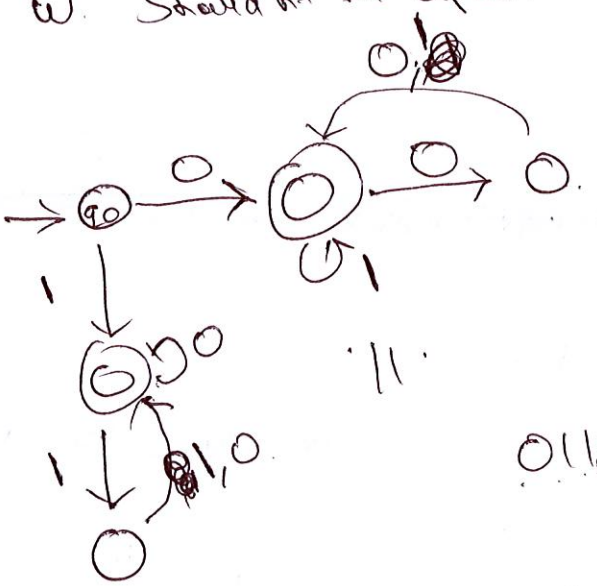
remove since

$q = q_0 q_1 \dots q_i \underbrace{q_{i+1} \dots q_j} q_{j+1} \dots q_{k-1}$

removing this will also get us an accepted state using a smaller string.

Thus we have $\forall k$, no DFA with fewer than k states can recognise WW_k.

w. Should not be a palindrom



00

0, 0...

11

1

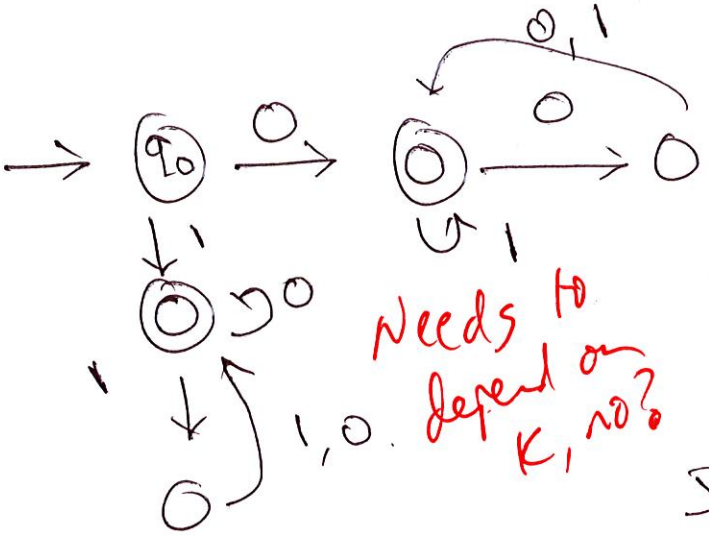
0110

0111

01110

(b) (10 points) Describe a much smaller NFA for $\overline{WW_k}$, the complement of WW_k. (**Note:** A clear description, including an analysis of the number of states, is required for full credit. It is not necessary to give a formal description.)

Any odd length string should be accepted. ✓



Needs to depend on k, no?

4

input, we accept.

with 2 accept states

You need to explain how to check that the 1st + second halves are different

The NFA is like this

Whenever we get ~~equal 0's~~ a palindrome & even length, we reject.

? If any odd string is

We have 5 states required.

ww is not a "palindrome" — ww^R is

(scratch paper)

(scratch paper)