USN

RV COLLEGE OF ENGINEERING®
(An Autonomous Institution affiliated to VTU)
IV Semester B. E. Fast Track Examinations Oct-2020
Computer Science and Engineering
THEORY OF COMPUTATION

Time: 03 Hours *Maximum Marks: 100*

Instructions to candidates:
1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2, 7 and 8 are compulsory. Answer any one full question from 3 and 4 & one full question from 5 and 6

PART-A

1 1.1 Find a string of minimum length in $\{0,1\}^*$ not in the language corresponding to the regular expression $0^*(100^*)^*1^*$ 01

1.2 Consider the two regular expressions
$r=0^*+1^*$ and $s=01^*+10^*+1^*0+(0^*1)^*$
Find a string corresponds to s but not in r 01

1.3 An NFA with states 1 to 5 and input alphabet $\{a,b\}$ has the following transition table.

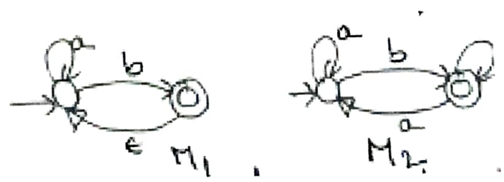
q	δ (q,a)	δ(q,b)
→1	{1,2}	{1}
2	{3}	{3}
3	{4}	{4}
4	{4}	{∅}
5	Φ	{5}

Draw the transition diagram and calculate $\delta^*(1,abaab)$ 02

1.4 Let $M=(Q,\Sigma,\delta,\ q_0.F)$ be an NFA. Show that for any $q\in Q$ and a Σ, $\delta^*(q,a)=\ \delta(q,a)$ 01

1.5 Suppose M is an NFA-$\in$ accepting $\subseteq\Sigma^*$. Describe how to modify M to obtain an NFA-$\in$ recognizing rev(L) = $\{x^r|x\}\in$L} 01

1.6 Consider two NFA-$\in$ below. Decide whether the two NFA-$\in$ accept the same language and give reasons for your answer.

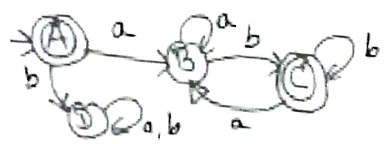


01

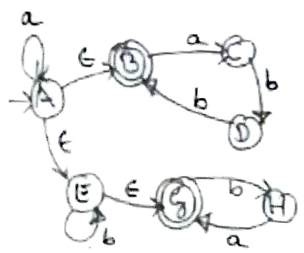
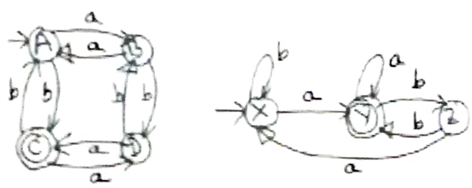
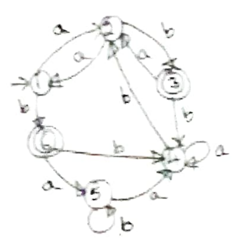
1.7 Describe the decision algorithm to answer the following equation "Given a regular expression γ and $DFA\ M$, are the corresponding language are same?" 01

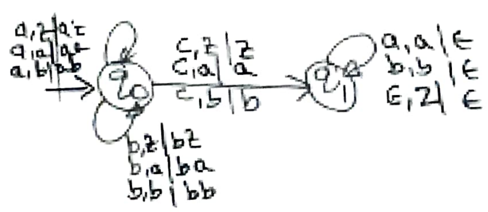
1.8 Consider the CFG with productions
$S\rightarrow aSbScS|aScSbS|bSaScS|bScSaS|cSbSaS|\in$.
Does this grammar generates the language.
$L=\{x|x\in\{a,b,c\}^*\ \&\ n_a(x)=n_b(x)=n_{c(x)}\}$. Justify your answer. 02

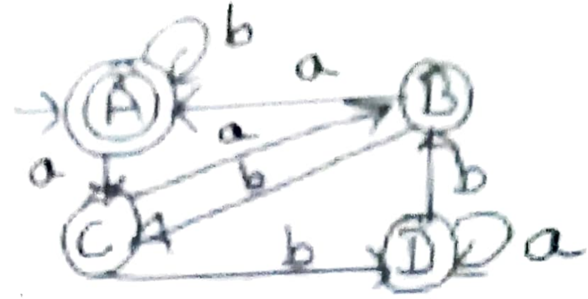
1.9 Show that CFG with productions $S\rightarrow\ a|Sa|bSS|SSb|SbS$ is ambiguous 01

	1.10	In the CFG below, identify the null productions and the unit productions. $S \rightarrow ABCBCDA, A \rightarrow CD, B \rightarrow Cb, C \rightarrow a \mid \epsilon, D \rightarrow bD \mid \epsilon$.	01
	1.11	Show that if L is accepted by a PDA in which no symbols are ever removed from the stack then L is regular	01
	1.12	If L is CFL then there exists a $DPDA$ which accepts L. is this statement true or false. Justify your answer.	01
	1.13	Describe the language generated by the regular grammar with productions $S \rightarrow aA \mid bC \mid b, A \rightarrow aS \mid bB, B \rightarrow aC \backslash bA \mid a, C \rightarrow aB \mid bS$.	01
	1.14	Given below a DFA accepting the language L, find a regular grammar generating $L - \{\epsilon\}$. 	01
	1.15	Give the unrestricted grammar to generate the language L over $\Sigma = \{a,b,c\}$, where $L = \{w \mid n_a(w) = n_b(w) = n_c(w)\}$.	02
	1.16	Give the transition diagram of a turing machine that accepts $L = \{w \mid w \in \{a,b\}^* \,\&\, w \in \{a,b\}^* aba$.	02

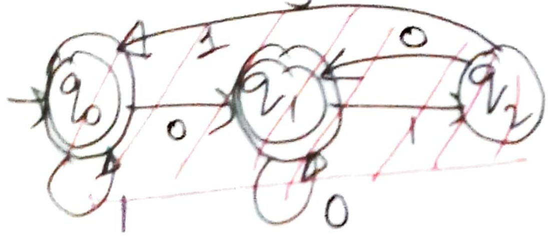
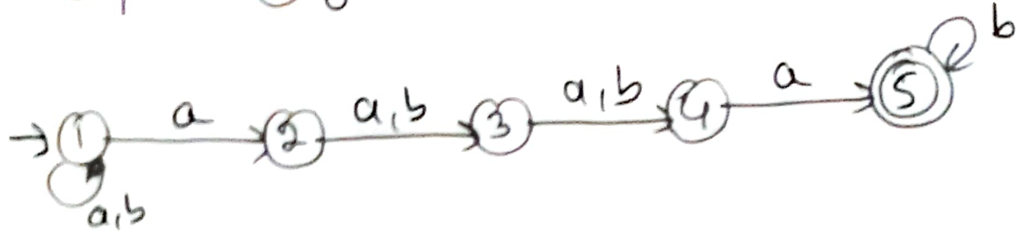
PART-B

2	a	Find the equivalent DFA to the given NFA-ϵ whose transition diagram is as follows. 	08
	b	Transition diagrams for two $DFA's$ $M1$ and $M2$ are shown below. Draw the DFA recognizing each of the following languages. i) $L1 \cup L2$ ii) $L1 - L2$  M1 M2	04
	c	Find a minimum state DFA for the below DFA 	04

3	a	State and prove pumping lemma for regular languages. Apply this lemma to show that $L = \{0^n \mid n\ is\ prime\}$ is not regular	08
	b	Find the CFG to generate each of the following languages: i) $L = \{a^i b^j c^k \mid i = j + k\}$ ii) $L = \{a^i b^j c^k \mid i = j\ or\ j = k\}$	04
	c	Simplify the below CFG with productions $S \to ABCBCDA, A \to CD, B \to Cb, C \to a \mid \epsilon, D \to bD \mid \epsilon$	04
		OR	
4	a	Find the language generated by the below grammar $S \to SS \mid bTT \mid TbT \mid TTb \mid \epsilon$ $T \to aS \mid SaS \mid Sa \mid a$	04
	b	Show that the CFG with productions $S \to S(S) \mid \epsilon$ is unambiguous.	04
	c	For the CFG , G given below, find a CFG, $G^{|}$ in GNF generating $L(G) - \{\epsilon\}$. $S \to AaA \mid cA \mid BaB, A \to aaBa \mid CDA \mid aa \mid DC, B \to bB \mid bAB \mid bb \mid aS, C \to Ca \mid bC \mid D, D \to bD \mid \epsilon$.	08
5	a	Let L be $L(M1)$ for some PDA with final state, $M1 = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, prove that there exists an empty stack PDA, $M2$ such that $L(M1) = L(M2)$. Construct final state PDA to accept $L = \{a^i b^j c^k \mid i = j\ or\ j = k\}$. Convert it into equivalent empty stack PDA.	09
	b	Define $DPDA$. Construct $DPDA$ to accept the language $L = \{x \mid x \in \{a, b\}^*\ \&\ n_a(x) > n_b(x)\}$. Illustrate the operation of this machine on the input string abbabaa	07
		OR	
6	a	How to find an equivalent CFG from a given PDA. Find the equivalent CFG to the PDA whose transition diagram is shown below. Show that the string $abacaba$ is accepted by the PDA and it is generated by the equivalent CFG.  	08
	b	Apply pumping lemma for CFL to show that the language $L = \{a^i b^j c^k \mid i < j < k$ is not context free.	03
	c	Consider two languages over $\Sigma = \{a, b, c\}$. $L1 = \{a^i b^j c^k \mid i < j$ and $L2 = \{a^i b^j c^k \mid i < k$. Show that $L1$ and $L2$ are context free language but $L1 \cap L2$ and L1 are not context free.	05

7	a	Construct LBA to accept the language $L = \{ww \mid w \in \{a,b\}^*\}$. Show that the string abbabb is accepted	08
	b	Find the equivalent left linear grammar to the language accepted by the following DFA 	05
	c	Show that all regular grammars are linear but every linear grammars need not be regular.	03
8	a	Construct a turing machine to accept the language $L = \{w \mid w \in \{a,b\}^* \,\&\, n_a(w) = n_b(w)\}$. Trace the machine for the string *abbbabaa*.	08
	b	Find the unrestricted grammar to generate the language $L = \{w \mid w \in \{a,b\}^*\}$. Show the derivation for the string *baabaa*.	04
	c	If $L1$ and $L2$ are recursively enumerable language over Σ then $L1 \cup L2$ and $L1 \cap L2$ are also recursively enumerable.	04

COURSE CODE: 16CS46 COURSE: Theory of Computation UG

Question No	PART – A	Marks
1.1	110	1
1.2	10 or any other string	1
1.3		2
1.4		2
	$\delta^*(1, abaab)$	
	$\delta^*(\delta(1,a), baab)$	
	$\delta^*(\{1,2\}, baab)$	
	$\delta^*(\delta(\{1,2\}, b), aab)$	
	$\delta^*(\{1,3\}, aab)$	
	$\delta^*(\delta(\{1,3\}, a), ab)$	
	$\delta^*(\{1,2,4\}, ab)$	
	$\delta^*(\delta(\{1,2,4\}, a), b)$	
	$\delta^*(\{1,2,5\}, b)$	
	$\delta^*(\delta(\{1,2,5\}, b, \epsilon)$	
	$\delta^*(\{1,3,5\}, \epsilon)$	
	$= \{1,3,5\}$ since 5 is the final state, the string abaab is accepted.	1
1.5	Since the string here is a single i/p symbol and hence $\delta^*(q,a) = \delta(q,a)$.	

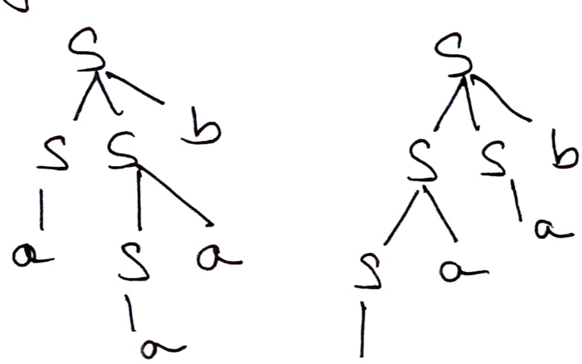
1.6. * reverse all the transitions
* Make the start state as final state
* Introduce a new start state and add transition upon ϵ to each final state of the given NFA-ϵ from this new start state.

1.7 YES: Both accepts the same language. for any string $x \in \{a,b\}^*$ which is accepted by M_1 is also accepted by M_2 — 1

1.8 YES it is decidable.
For the given r, construct NFA-ϵ using Kleen's theorem part-I. Convert this NFA-ϵ to DFA, let it be M'. Apply minimization algorithm on M' & M. If both the minimized DFAs are isomorphic to each other, the corresponding languages are same. —

1.9 NO: for example aabbcc can't be derived from the given grammar. Students may argue upon any other string. — 2

1.10 Let $x = aaab$, two parse trees for this string are — 1



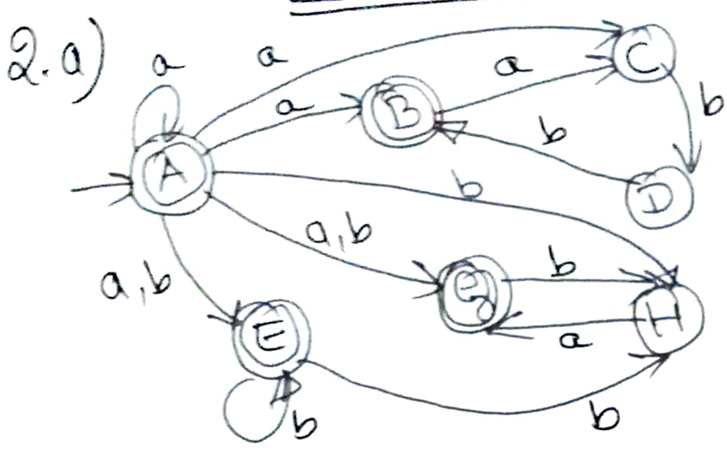
Students may consider any other string.

...E CODE: 16CS~~66~~ COURSE: Theory of Computation UG

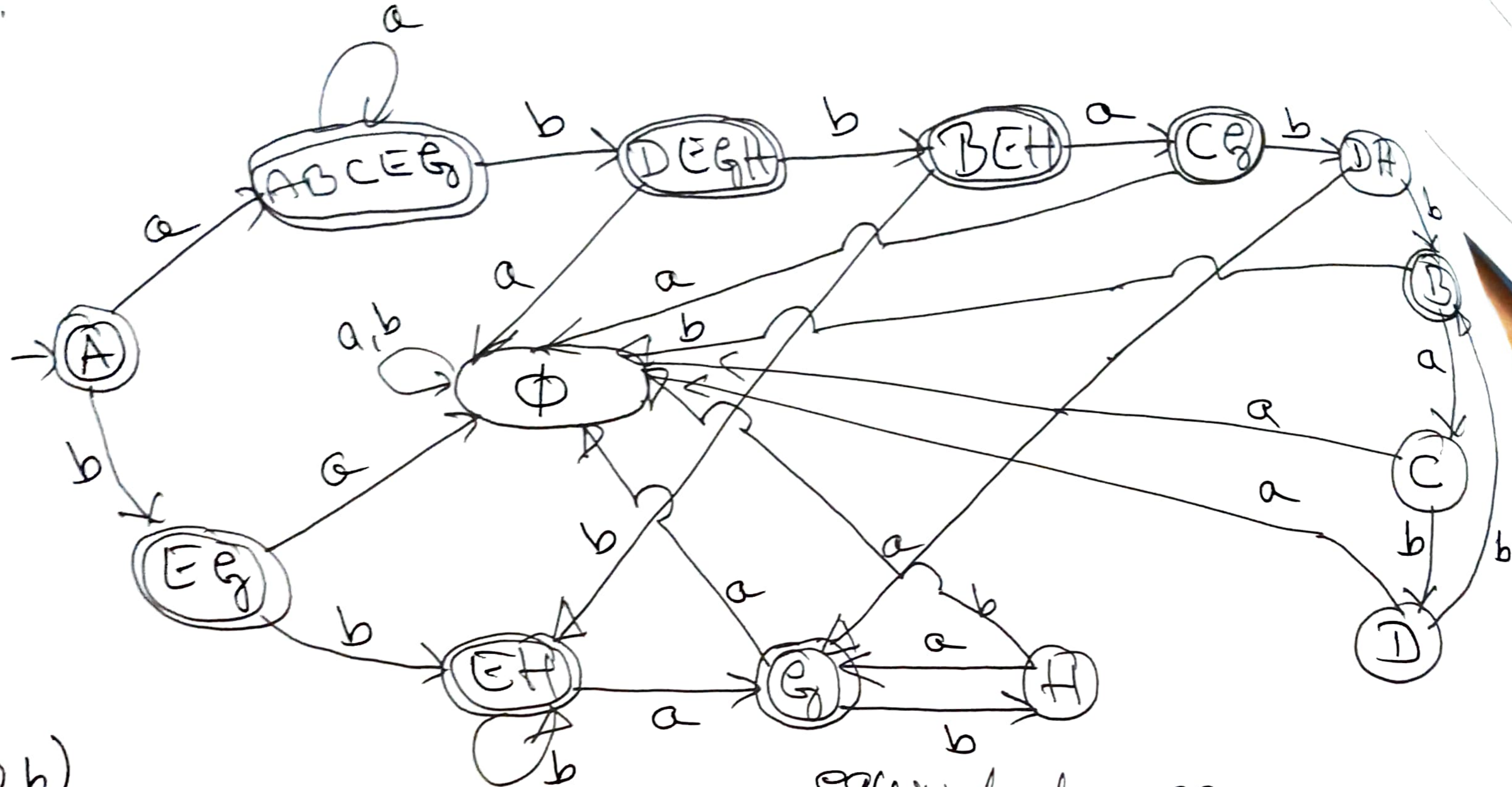
...n		Marks
1.11	Null productions are $D \to \epsilon, C \to \epsilon, A \to \epsilon$ Unit productions are $A \to C, A \to D$	1
1.12	19 ~~steps~~ required.	1
1.13	For every regular language we can construct DFA, For each DFA we can construct equivalent DPDA in which no symbols are pushed or poped from the stack, i.e if $\delta(q,a) = p$ is the transition in DFA, then $\delta(q,a,Z) = (p,Z)$ becomes the transition in DPDA.	1
1.14	NO: For all CFL, DPDA does not exists. For ex, $L = \{w \mid w \in \{a,b\}^* \text{ \& } w \text{ is a palindrome}\}$. It is not possible to build DPDA to this language. Students may choose any other languages as a counter example.	1
1.15	$L = \{w \mid w \in \{a,b\}^*$ and w has even number of a's & odd no. of b's$\}$	1
1.16	$A \to aB \mid bD,\ B \to aB \mid bC,\ C \to aB \mid bC \mid \epsilon,$ $D \to aD \mid bD$	1
1.17	$S \to ABCS \mid ABC,\ AB \to BA,\ AC \to CA,\ BC \to CB,$ $BA \to AB,\ CA \to AC,\ CB \to BC,\ A \to a,\ B \to b,\ C \to c.$	2
1.18		2

PART-B

2.a)

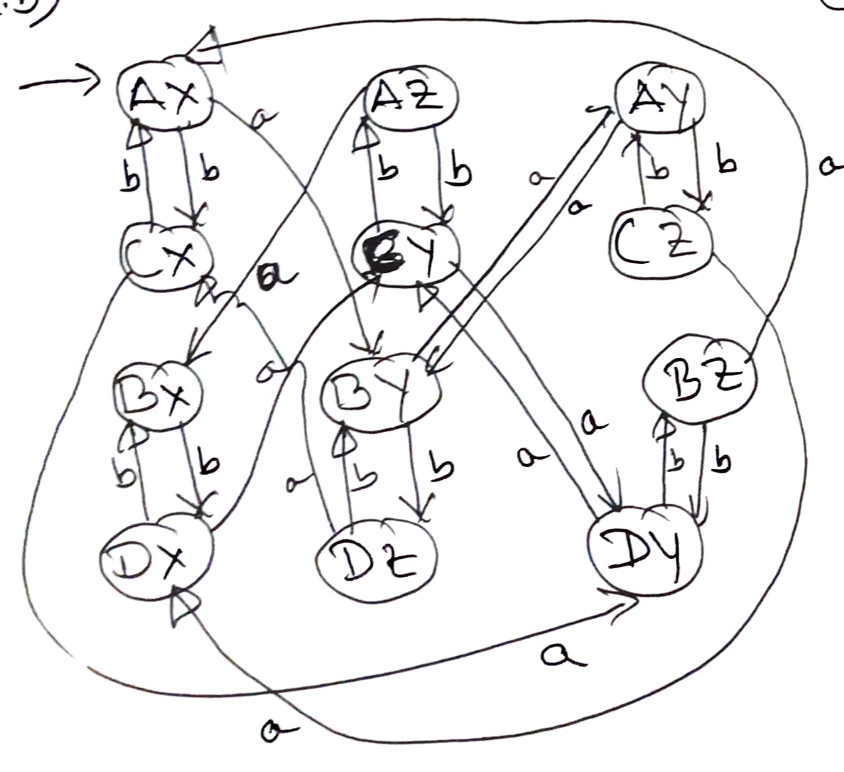


equivalent NFA



equivalent DFA

2.b)



— 4

For $L_1 \cup L_2$ consider AX, AY, AZ, BY, CY & DY as final states

For $L_1 - L_2$ consider AX and AZ as final states

$3+3=6$

Page-4

	Marks

2c)

2	2				
3	1	1			
4	2	2	1		
5	2	2	1	2	
6	1	1	2	1	1
	1	2	3	4	5

— 2

Minimization not possible because all states are distinguishable — 2

3. a) Statement of the pumping lemma — 2
proof of the lemma — 3
By applying lemma to show the given language is not regular — 3

b) i) $S \to aSc \mid aAb \mid \epsilon$
$A \to aAb \mid \epsilon$ — 2
(or) any other grammar

ii) $S \to AB \mid CD$
$A \to aAb \mid \epsilon$, $B \to cB \mid \epsilon$
$C \to aC \mid \epsilon$, $D \to bDc \mid \epsilon$ — 2
(or) any other grammar

3.c) Here, variables A, C, D are null
By eliminating null productions, the
$S \to ABCBCDA$ gives additional 3
productions, along with the following p

$A \to CD \mid C \mid D$
$B \to Cb \mid b$
$C \to a$
$D \to bD \mid b$

eliminate Unit productions $A \to C$, & $A \to$

$A \to CD \mid bD \mid a \mid b$
$B \to Cb \mid b$
$C \to a$
$D \to bD \mid b$ — 4

4.a) $L(G) = \{ w \mid w \in \{a,b\}^+ \mid n_a(x) = 2n_b(x) \}$ — 4

b) This grammar generates balanced parenthesis. Any valid balanced parenthesis can be generated in only one way. — 4
(or) any other argument may be considered.

c) Here S, A, C & D are nullable

$S \to AaA \mid CA \mid A \mid BaB \mid Aa \mid aA \mid a \mid C$
$A \to aaBa \mid CDA \mid CA \mid DA \mid A \mid aa \mid C \mid D$
$B \to bB \mid bAB \mid bB \mid bb \mid aS \mid a$
$C \to Ca \mid bC \mid a \mid b$ — 1
$D \to bD \mid b$

	Marks
Now, eliminate the unit productions	
$S \to A \mid C$, $A \to A \mid C \mid D$.	
$\therefore S \to AaA \mid CA \mid BaB \mid Aa \mid aA \mid a \mid aaBa \mid CDA \mid CA \mid DA \mid aa \mid Ca \mid bC \mid a \mid b \mid bD$	
$A \to aaBa \mid CDA \mid CA \mid DA \mid aa \mid Ca \mid bC \mid bD \mid a \mid b$	1
$B \to bB \mid bAB \mid bB \mid bb \mid aS \mid a$	
$C \to Ca \mid bC \mid a \mid b$	
$D \to bD \mid b$	
Eliminate left recursion in $C \to Ca$	
$C \to bCE \mid aE \mid bE \mid bC \mid a \mid b$	1
$E \to aE \mid a$	
By substitution ensure that the right hand side of each production has only variables or a terminal followed by a variables.	
$S \to AXA \mid CA \mid BXB \mid AX \mid aA \mid a \mid aXBX \mid CDA \mid CA \mid DA \mid aX \mid CX \mid bC \mid bD \mid a \mid b$	
$A \to aXBX \mid CDA \mid CA \mid DA \mid aX \mid CX \mid bC \mid bD \mid a \mid b$	2
$B \to bB \mid bAB \mid bY \mid aS \mid a$	
$C \to bCE \mid aE \mid bE \mid bC \mid a \mid b$	
$D \to bD \mid b$, $E \to aE \mid a$, $X \to a$, $Y \to b$	

B, C, D, E, X & Y productions are in GNF. Substitute these in A productions

$A \rightarrow aXBX \mid bCEDA \mid aEDA \mid bEDA \mid bCDA \mid aDA \mid bDA \mid bCEA \mid aEA \mid bEA \mid bCA \mid bA \mid bDA \mid aX \mid bC \mid bD \mid a \mid b \mid bCEX \mid aEX \mid bEX \mid aX \mid bX$

$B \rightarrow bB \mid bAB \mid bY \mid aS \mid a$

$C \rightarrow bCE \mid aE \mid bE \mid bC \mid a \mid b$

$D \rightarrow bD \mid b$

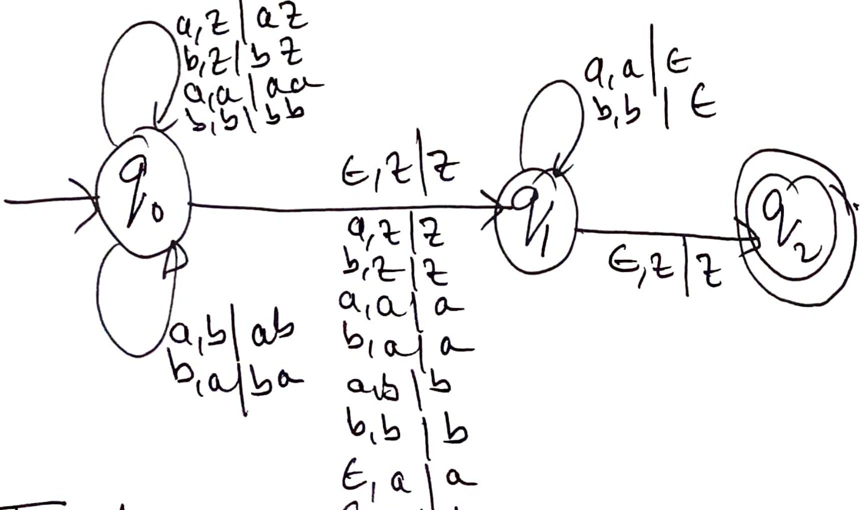
$E \rightarrow aE \mid a$

$X \rightarrow a$

$Y \rightarrow b$

Now, substitute the above productions in S productions, we will get 87 more productions associated with S.

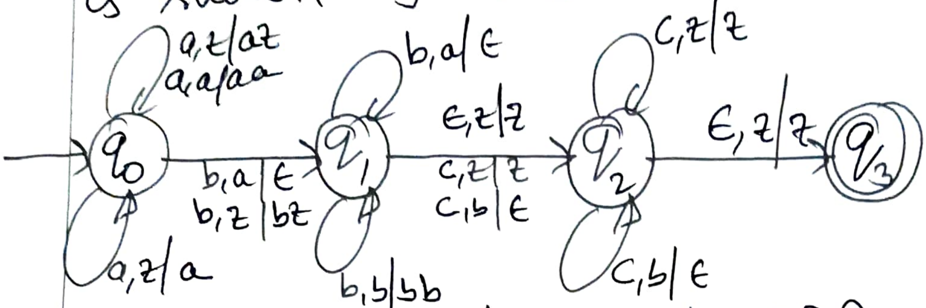
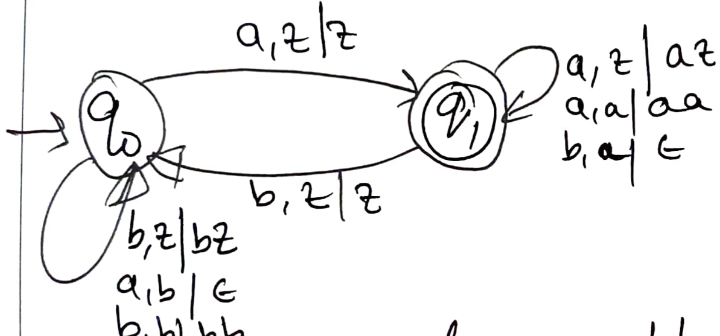
5.a) The transition diagram for the PDA to accept palindrome is as below.



——— 3

To show baab string is accepted by this PDA by using ID's

——— 2

	Marks
5) procedure to convert final state PDA to empty stack PDA — 8M	2
Final state PDA for the given language is shown by the transition diagram	
	4
To write empty stack PDA	2
5.c) Definition of DPDA 8M	2
DPDA to the given language is	
	2
To show the string abbabaa is accepted by the above DPDA	2
6.a) Steps to be followed to construct CFG from the given PDA	2
productions of the equivalent grammar ~~for the given PDA~~ are listed below	
$S \to [q_0, Z, q]$	
$[q_0,Z,q] \to a[q_0,a,P][P,Z,q] \mid b[q_0,b,P][P,Z,q]$	
$[q_0,a,q] \to a[q_0,a,P][P,a,q] \mid b[q_0,b,P][P,a,q]$	4
$[q_0,b,q] \to a[q_0,a,P][P,b,q] \mid b[q_0,b,P][P,b,q]$	
$[q_0,Z,q] \to c[q_1,Z,q]$, $[q_0,a,q] \to c[q_1,a,q]$	
$[q_0,b,q] \to c[q_1,b,q]$, $[q_1,a,q] \to a$, $[q_1,b,q] \to b$	
$[q_1,Z,q] \to \epsilon$	

where p and q can be substituted by q_0 and q_1, which leads to 35 productions in total.
To show abacaba is accepted by PDA
To show the string abacaba is derived from CFG.

6.b) To show $L = \{a^i b^j c^k \mid i < j < k\}$ is not CFL, by applying pumping Lemma for CFL — 3

6.c) Let G_1 is

$S \to ABC$, $A \to aAb \mid \epsilon$, $B \to bB \mid b$, $C \to cC \mid \epsilon$ — 1

& G_2 is

$S \to AC$, $A \to aAc \mid B$, $B \to bB \mid \epsilon$, $C \to cC \mid c$

G_1 and G_2 generates L_1 and L_2 respectively. — 1

$L_1 \cap L_2 = \{a^i b^j c^k \mid i < j \ \& \ i < k\}$

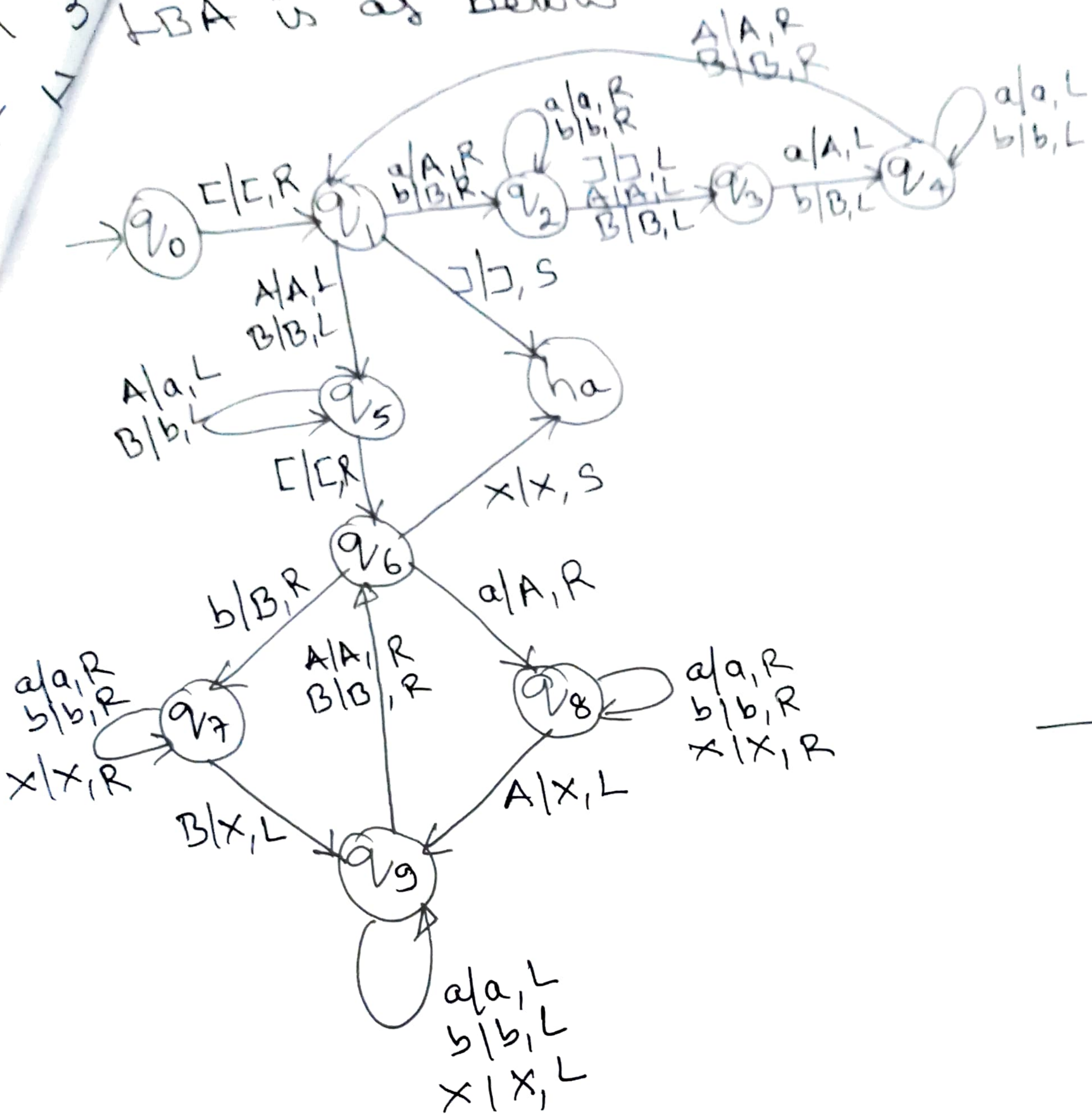
W.K.T, $L_1 \cap L_2$ is not CFL or by applying pumping lemma we can show that $L_1 \cap L_2$ is not CFL. — 2

$L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$

$\therefore \overline{L_1}$ is not CFL. — 1

e: 16CS46 Subject: Theory of Computation

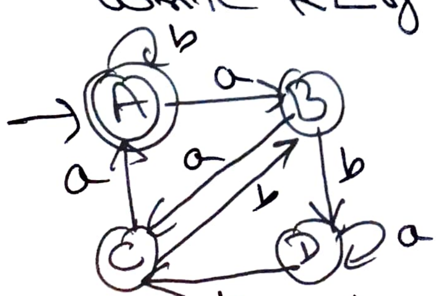
Transition diagram for the required LBA is as below



—— 5

By tracing the string abbabb, using the above LBA and to show it is accepted —— 3

7.b) Reverse the given DFA and write RLG to the reversed automata.



Reversed automata

$A \to aB \mid bA \mid \epsilon$

$B \to aC \mid bD$

$C \to aA \mid bB$

$D \to aD \mid bC$

RLG to this automata

$\Rightarrow$

$A \to Ba \mid Ab \mid \epsilon$

$B \to Ca \mid Db$

$C \to Aa \mid Bb$

$D \to Da \mid Cb$

$1+1+2 = 4$

equivalent LLG for this RLG.

7.c) Regular grammars are either right linear or left linear grammars whose productions are of the form

$A \to \alpha \mid \alpha B$ or $A \to \alpha \mid B\alpha$

where $\alpha \in T^*$.

Since these grammars contain only one variable on the right side of each production they are linear.

Now, consider a grammar G with the production $S \to aSb \mid ab$.

The language of this grammar is

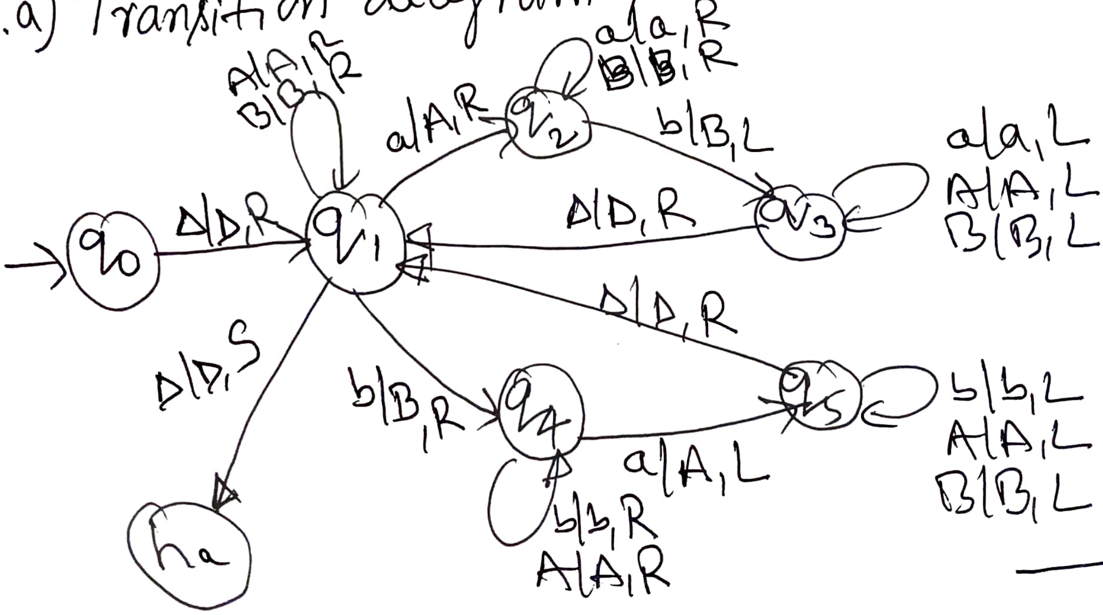
$L(G) = \{ a^i b^i \mid i \geq 1 \}$

W.K.T this language is not regular. —— 1

7.d)

$L(G) = \{ a^i \mid i \geq 1 \}$ —— 2

8.a) Transition diagram for the TM is as below



—— 5

Tracing the TM for the string abbbabaa —— 3

No: 16CSA6. Subject: Theory of Computation

$S \rightarrow FM$

$F \rightarrow FaA \mid FbB$

$Aa \rightarrow aA$, $Ab \rightarrow bA$, $Ba \rightarrow aB$, $Bb \rightarrow bB$

$AM \rightarrow Ma$, $BM \rightarrow Mb$ — 2

$F \rightarrow \epsilon$, $M \rightarrow \epsilon$

The derivation for the string baabaa from the above grammar is as below.

$S \Rightarrow FM$; $S \rightarrow FM$

$\Rightarrow FaAM$; $F \rightarrow FaA$

$\Rightarrow FaAaAM$; $F \rightarrow FaA$

$\Rightarrow FbBaAaAM$; $F \rightarrow FbB$

$\Rightarrow FbaBAaAM$; $Ba \rightarrow aB$

$\Rightarrow FbaBaAAM$; $Aa \rightarrow aA$

$\Rightarrow FbaBaAMa$; $AM \rightarrow Ma$ — 2

$\Rightarrow FbaBaMaa$; $AM \rightarrow Ma$

$\Rightarrow FbaaBMaa$; $Ba \rightarrow aB$

$\Rightarrow FbaaMbaa$; $BM \rightarrow Mb$

$\Rightarrow Fbaa\epsilon baa$; $M \rightarrow \epsilon$

$\Rightarrow baabaa$; $F \rightarrow \epsilon$

8. c) To show $L_1 \cup L_2$ is r.e.L — 2

To show $L_1 \cap L_2$ is r.e.L — 2