USN					



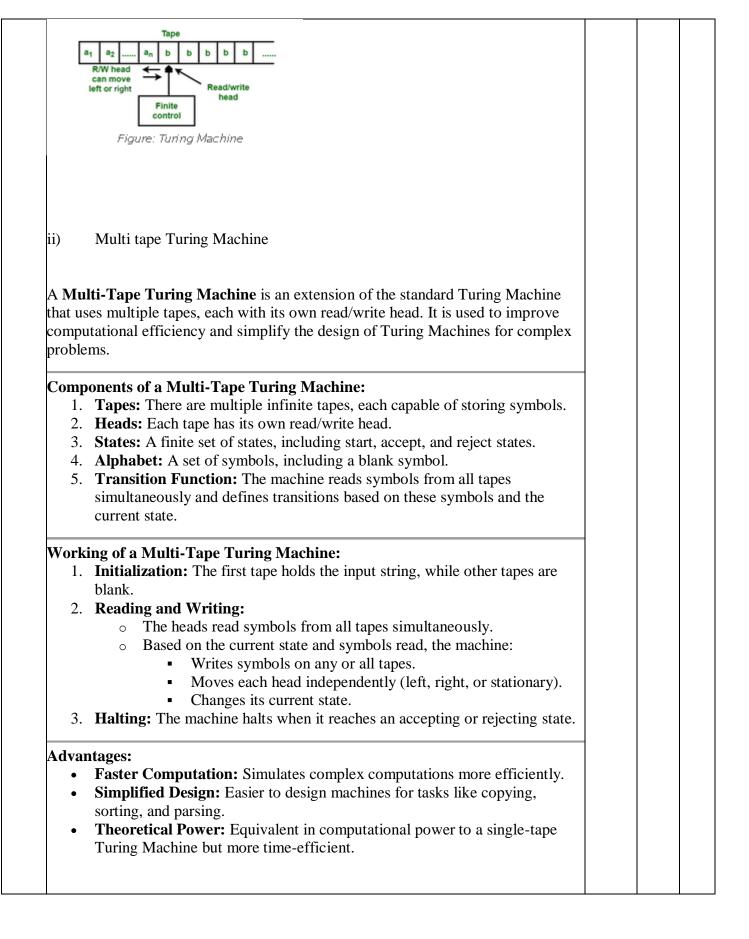
	Internal Assessment Test 2)24	T			
Sub:	Theory of Computation	Sub Code:	BCS503	Branch:	AIN AIN	1L/CSE 11	3
Date:	15/12/2024 Duration: 90min's Max Marks: 50		V/A,B CSEA	AIML(A)		OBE	E
	Answer any FIVE FULL Ques	tions		M	ARKS	СО	RB
1.(a)	Construct PDA that accepts the language $L=\{ww^R reversal of w\}$. Write ID for the string aabbaa.	w∈{a+b}* a	nd w ^R is the		6	CO3	L2 L2
	Where, q0 = Initial state qf = Final state						
	$ \varepsilon = \text{ indicates pop operation} $ $ (a, b/ab) (a, a/a) (a, a/\epsilon) (b, b/\epsilon) (b, b/\epsilon) (b, c/\epsilon) (c, c/z) (b, b/b) (b, c/b) (b, b/bb) (b, a/ba) $	qf					
	Required NPDA Let's create the ID for the string aabbaaaabbaaaabbaa 1. Initial ID (Before any steps):						
	(q0,aabbaa,Z0)(q_0, aabbaa, Z_0)(q0,aabbaa,Z0) o State: q0q_0q0 o Input: aabbaaaabbaaaabbaa o Stack: [Z0][Z_0][Z0]						
	 2. After reading the first aaa: (q0,abbaa,[a,Z0])(q_0, abbaa, [a, Z_0])(q0,abbaa,[a,Z_0) State: q0q_0q0 Input: abbaaabbaaabbaa Stack: [a,Z0][a, Z_0][a,Z0] 	0])					
	 3. After reading the second aaa: (q0,bbaa,[a,a,Z0])(q_0, bbaa, [a, a, Z_0])(q0,bbaa,[a,a o State: q0q_0q0 o Input: bbaabbaabbaa o Stack: [a,a,Z0][a, a, Z_0][a,a,Z0] 	a,Z0])					
	4. After reading the first bbb: $(q0,baa,[a,a,b,Z0])(q_0, baa, [a, a, b, Z_0])(q0,baa,[a, o State: q0q_0q0)$ \circ Input: baabaabaa	a,b,Z0])					
	 Stack: [a,a,b,Z0][a, a, b, Z_0][a,a,b,Z0] 5. After reading the second bbb: (q0,aa,[a,a,b,b,Z0])(q_0, aa, [a, a, b, b, Z_0])(q0,aa,[a o State: q0q_0q0 						
	 Input: aaaaaa Stack: [a,a,b,b,Z0][a, a, b, b, Z_0][a,a, 	b,b,Z0]					

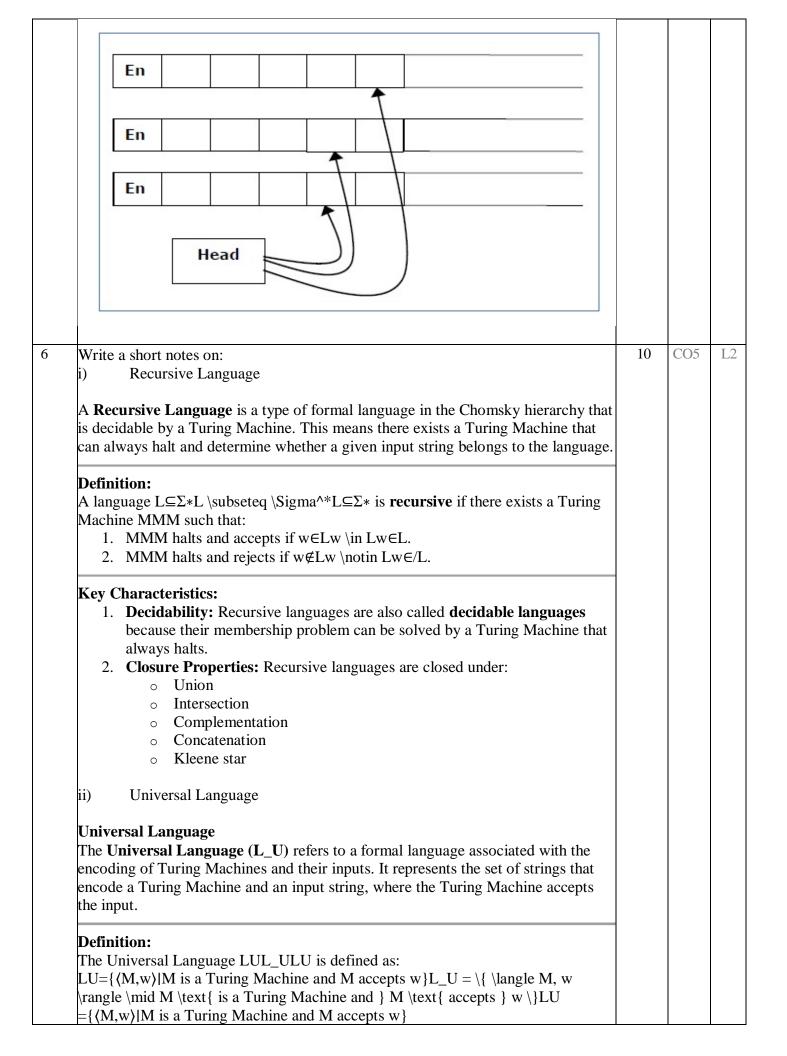
6. Transition to qi_q_1qi (start reading second half): (qi.aa, la.a, b.b. b, b, b, Z, 0)(qi.a.a, la.a, b, bZ) (qi.a.a, la.a, b, b, Z, 0)(qi.a.a, la.a, b, bZ) (qi.a.a, la.a, b, bZ, 0)(qi.a.a, la.a, b, bZ) (qi.a.a, la.a, b, bZ, 0)(qi.a.a, la.a, b, bZ) (qi.a.a, la.a, b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 0)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 2)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 2)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ, 2)(qi.a.a, la.b, bZ) (qi.a.a, la.b, bZ) (1	1	
$ \begin{vmatrix} \circ & \text{State: } q q_1 q_1 \\ \circ & \text{Ingut: aaaaaa} \\ \circ & \text{Stack: } [aa,b,b,Z0][a, a, b, b, Z, 0][a,a,b,b,Z0] \\ \hline \textbf{7. After reading the first aaa of second half: \\ (q1,a[ab,b,Z0])(q_1, a [a, b, b, Z, 0](q1,a][a,b,b,Z0] \\ \circ & \text{State: } q q_1 q_1 \\ \circ & \text{Ingut: aaa} \\ \circ & \text{Stack: } [a,b,b,Z0][a, b, b, Z, 0](q1,a][a,b,b,Z0] \\ \hline \textbf{8. After reading the second aaa: \\ (q1,c[b,b,Z0])(q_1, a [b,b, Z, 0](q1,c,[b,b,Z0] \\ \circ & \text{State: } q q_1 q_1 \\ \circ & \text{Ingut: aaa} \\ \text{Ingut: aaa} \\ \text{State: } q(q_1,q_1) \\ \hline \textbf{8. After reading the first bbi: \\ (q1,c[b,b,Z0])(q_1, a,b,b,Z,0](q1,c,[b,b,Z0] \\ \circ & \text{State: } q q_1,q_1 \\ \circ & \text{Ingut: eqssilone } \\ \text{State: } q q_1,q_1 \\ \text{State: } (20)[0,q_1,(g_1,b,Z_0)](g_1,c_1,b,Z_0)] \\ \text{Thus, the final ID shows that the PDA has accepted the string. \\ 1.(b) Convert the following CFG to PDA and give the procedure for the same. \\ \text{State: } S \rightarrow \text{AASB}[aAA A \rightarrow ABB](aA A A \rightarrow ABB](AA A \rightarrow ABB](AA A A \rightarrow ABB)(AA A A A ABB)(AA A A A A BBA (A ABB) \\ \ \ \ \begin{tabular}{lllllllllllllllllllllllllllllllllll$					
• Input: iaaiaai • Stack: [a.a.b, Z.0](a.a.b, b.Z.0](a.a.b,b.Z0) 7. After reading the first ana of second half: (q1.a.[a.b,b.Z0)(q_1, a. [a. b, b.Z_0])(q1,a,[a,b,b,Z0]) • State: (q1_q1q1 • Input: aaa • State: (a.b,b,Z0](b, b, Z_0][a.b,b,Z0] 8. After reading the second aaa: (q1,c,[b,b,Z0](b, b, Z_0][b, b, Z_0][a,b,b,Z0] • State: (q1_q1q1 • Input: clepsilon; • State: (q1_q1) • Input: clepsilon; • State: (b,Z0][b, b, Z_0][b,b,Z0] • State: (b,Z0][b, b, Z_0][b,D,Z0] • State: (q1_q1) • Input: clepsilon; • State: (b,Z0][b, b, Z_0][b,Z0] • State: (b,Z0][b, Z_0][b,Z0] • State: (b,Z0][b, Z_0][b,Z0] • State: (b,Z0][c, [vepsilon; (q1,c[20])(q, [vepsilon, [Z_0])(q1,c.[Z0]) • State: [b,Z0][b,Z_0][C0] • State: [b,Z0][b,Z_0][C0] • State: [b,Z0][c, [vepsilon; Z_0] • State: [b,Z0][c, [vepsilon; Z_0] • State: [dq_q1] • Input: clepsilon; • State: [dq_q1] • State: [dq_q2] • State: [dq_q4] • State: [dq_q4] • State: [dq_q4]		$(q1,aa,[a,a,b,b,Z0])(q_1,aa,[a,a,b,b,Z_0])(q1,aa,[a,a,b,b,Z0])$			
• Stack: [a,a,b,b,Z0][a,a,b,b,Z] 7. After reading the first aaa of second half: (q1,a,[a,b,Z0])(q_1, a, [a, b, b, Z_0])(q1,a,[a,b,b,Z0]) • State: q1q_1q1 • Input: aaa • Stack: [a,b,b,Z0](a, b, b, Z_0][a,b,b,Z0] 8. After reading the second aaa: (q1,c,[b,b,Z0](q_1, \epsilon, [b, b, Z_0])(q1,c,[b,b,Z0]) • State: q1q_1q1 • Input: c\epsilon(• State: tb,b,Z0][b, b, Z_0][b,b,Z0] 9. After reading the first blb: (q1,c,[b,Z0](q_1, \epsilon, [b, b, Z_0](b,b,Z0] • State: tb,b,Z0][b, Z_0](b,Z,0](b,C) 10. After reading the first blb: (q1,c,[b,Z0](q_1, \epsilon, [b, Z_0](q1,c,[b,Z0]) • State: tb,DZ0][b, Z_0](b,Z0] 10. After reading the second bdb: (q1,c,[b,Z0](q_1, \epsilon, [Z_0](q1,c,[b,Z0]) • State: tb,Z0][b,Z_0](b,Z0] 10. After reading the second bdb: (q1,c,[Z0](Q_1, \epsilon, [Z_0](b,Z0]) 10. After reading the second bdb: (q1,c,[Z0](Q_1, \epsilon, [Z_0](b,Z0]) • State: tc,Z0][Z_0] Thus, the final D shows that the PDA has accepted the string. 11.b) Convert the following CFG to PDA and give the procedure for the same. S ⇒ aABB aAA A ⇒ aBB aAA C ⇒ a □ Initialization: Start with the initial stack symbol Z0Z_0Z0. 8(q0,c,Z)=(q0,AAB)(deta(q_0, \epsilon, Z_0) = (q_0, S)6(q0,c,Z)=(q0,AAB) □ For S = aAABB (to aABS = aAAB: 8(q0,c,S)=(q0,AAA)(deta(q_0, \epsilon, S) = (q_0, aABB)6(q0,c,S)=(q0,aAAB) □ For S = aAABB (to aABS = AABB: 8(q0,c,A)=(q0,ABB)(deta(q_0, \epsilon, S) = (q_0, aABB)6(q0,c,A)=(q0,aBB) □ For C = aABB (to aBA → aBB: 8(q0,c,A)=(q0,ABB)(deta(q_0, \epsilon, S) = (q_0, aABB)6(q0,c,A)=(q0,aBB) □ For C = aA(b BA → aBB: 8(q0,c,C)=(q0,A)(deta(q_0, \epsilon, B) = (q_0, A)8(q0,c,A)=(q0,aBB) □ For C → aC (to aC → a: 8(q0,c,C)=(q0,A)(deta(q_0, \epsilon, B) = (q_0, A)8(q0,c,B)=(q0,A) □ For C → aC (to aC → a: 8(q0,c,C)=(q0,A)(deta(q_0, \epsilon, B) = (q_0, A)8(q0,c,B)=(q0,A) □ For C → aC (to aC → a: 8(q0,c,C)=(q0,A)(deta(q_0, \epsilon, B) = (q_0, A)8(q0,c,B)=(q0,A) □ For C → aC (to aC → a: 8(q0,c,C)=(q0,A)(deta(q_0, \epsilon, B) = (q_0, A)8(q0,c,B)=(q0,A) □ For C → aC (to aC → a:		\circ State: q1q_1q1			
7. After reading the first aaa of second half: (q1.a,[a,b,b,Z0])(q_1, [a, [a, b, b, Z_0])(q1.a, [a, b, b,Z0]) \circ State: q1q_1q1 \circ Input: aaa \circ State: q1q_1q1 \circ Input: eqsilone for both the process of the second aaa: (q1,e,[b,b,Z0])(q1,e,[b,b,Z0]) \circ State: q1q_1q1 \circ Input: eqsilone for both the process of the second aca: (q1,e,[b,Z0])(q_1, [vgsilon, [b, Z_0])(q1,e,[b,Z0]) \circ State: q1q_1q1 \circ Input: eqsilone for both the process of the second aca: (q1,e,[b,Z0])(q_1, [vgsilon, [b, Z_0])(q1,e,[b,Z0]) \circ State: q1q_1q1 \circ Input: eqsilone for both the process of the second babb: (q1,e,[b,Z0])(q_1, [vgsilon, [Z_0])(q1,e,[Z_0]) \circ State: (p20][b, Z_0][b, Z_0][b, Z_0] \circ State: (p20][b, Z_0][b, Z_0][b, Z_0] \circ State: (p20][b, Z_0][b, Z_0] \circ State: (p20][Z_0][Z_0] Thus, the final ID shows that the PDA has accepted the string. I.to) Convert the following CFG to PDA and give the procedure for the same. S \Rightarrow AABB[aAA \Rightarrow AaBB]A $C \Rightarrow a$ \Box for S \rightarrow AABB (to aABS \rightarrow aABB: $\delta(q0,c,Z0) - (q0,S) \delta(abta(q_0, [vgsilon, Z_0) - (q_0, S) \delta(q0,c, C3) - (q0, aABB) \delta(q0,c, S) - (q0, aAAB)$ \Box For S \rightarrow AABB (to aBBA \rightarrow aABB: $\delta(q0,c,A) - (q0, aABB) \delta(abta(q_0, [vgsilon, S) = (q_0, aAA) \delta(q0,c, S) - (q0, aAAB)$ \Box For S \rightarrow AABB (to aBBA \rightarrow aABB: $\delta(q0,c,A) - (q0, aABB) \delta(abta(q_0, [vgsilon, S) = (q_0, aAA) \delta(q0,c, S) - (q0, aAA)$ \Box For C \rightarrow aAA (to aABB \rightarrow aABB: $\delta(q0,c,A) - (q0, aABB) \delta(abta(q_0, [vgsilon, S) = (q_0, aAA) \delta(q0,c, C) - (q0, aBB)$ \Box For C \rightarrow aAA (to aAAB \rightarrow at BB: $\delta(q0,c,A) - (q0, a) dBBA \rightarrow aBB: \delta(q0,c,A) - (q0, a) dBBA \rightarrow aBB: \delta(q0,c,C) - (q0, a) dBBA \rightarrow aBB: \delta(q0,c,C) - (q0, a) delta(q_0, [vgsilon, A) = (q_0, a) \delta(q0,c,A) - (q0, aBB) \Box For C \rightarrow act (to aC \rightarrow a: \delta(q0,c,C) - (q0, a) delta(q_0, [vgsilon, B) = (q_0, bBB) \delta(q0,c,B) - (q0, A) \Box For C \rightarrow act (t$		o Input: aaaaaa			
$ \begin{array}{ c } (q1,a,[a,b,b,Z0])(q,1,a,[a,b,b,Z,0]) & \circ State: q1,a[q1] & \circ Input: aaa & \circ State: (a,b,b,Z0](a,b,b,Z0] & A fler reading the second aaa: (q1,s,[b,b,Z0](q,1,\epsilon, [b, b, Z,0](q1,s,[b,b,Z0]) & \circ State: q1,a[q1] & \circ Input: clepsilont & \circ State: (a,b,L,Z0](b,b,Z0] & A fler reading the first bbb: (q1,s,[b,Z0])(q,1, \epsilon, [b, b, Z,0](p1,b,Z0] & \circ State: (a,b,L,Z0](b,b,Z0] & \circ State: (b,b,Z0](b,b,Z0] & \circ State: (a,b,L,Z0](b,b,Z0] & \circ State: (b,b,Z0](b,c,D) & \circ State: (b,b,Z0](c,1, \epsilon, [b, Z,0](c,b,Z0] & \circ State: (b,b,Z0](b,d,C) & \circ State: (b,b,Z0](c,1, \epsilon, b, Z,0)(q1,s,[b,Z0] & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (c,f,G) & \circ State: (C,G)(c,C) & \circ State: (Z0](Z,0) & \circ State: (Z0)(Z,0) & \circ Sta$		• Stack: [a,a,b,b,Z0][a, a, b, b, Z_0][a,a,b,b,Z0]			
$ \begin{array}{ c } (q1,a,[a,b,b,Z0])(q,1,a,[a,b,b,Z,0]) & \circ State: q1,a[q1] & \circ Input: aaa & \circ State: (a,b,b,Z0](a,b,b,Z0] & A fler reading the second aaa: (q1,s,[b,b,Z0](q,1,\epsilon, [b, b, Z,0](q1,s,[b,b,Z0]) & \circ State: q1,a[q1] & \circ Input: clepsilont & \circ State: (a,b,L,Z0](b,b,Z0] & A fler reading the first bbb: (q1,s,[b,Z0])(q,1, \epsilon, [b, b, Z,0](p1,b,Z0] & \circ State: (a,b,L,Z0](b,b,Z0] & \circ State: (b,b,Z0](b,b,Z0] & \circ State: (a,b,L,Z0](b,b,Z0] & \circ State: (b,b,Z0](b,c,D) & \circ State: (b,b,Z0](c,1, \epsilon, [b, Z,0](c,b,Z0] & \circ State: (b,b,Z0](b,d,C) & \circ State: (b,b,Z0](c,1, \epsilon, b, Z,0)(q1,s,[b,Z0] & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (b,Z0](b,d,C) & \circ State: (c,f,G) & \circ State: (C,G)(c,C) & \circ State: (Z0](Z,0) & \circ State: (Z0)(Z,0) & \circ Sta$		7. After reading the first aaa of second half:			
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
8. After reading the second aaa: (q1,c,[b,b,Z0])(q_1, (epsilon, [b, b,Z0])(q_1,c,[b,b,Z0]) Stack: [bh,Z0](b, b, Z,0](b,b,Z0] 9. After reading the first bbb: (q1,c,[b,Z0])(q_1, (epsilon, [b, Z,0])(b,b,Z0]) 9. After reading the first bbb: (q1,c,[b,Z0])(q_1, (epsilon, [b, Z,0])(b,b,Z0]) 9. After reading the first bbb: (q1,c,[b,Z0])(q_1, (epsilon, [b, Z,0])(b,D,Z0]) 10. After reading the second bbb: (qf,c,[Z0])(q,f, (epsilon, [Z,0])(qf,c,[Z0]) 10. After reading the second bbb: (qf,c,[Z0])(q,f, (epsilon, [Z,0])(qf,c,[Z0]) 10. After reading the second bbb: (qf,c,[Z0])(q,f, (epsilon, [Z,0])(qf,c,[Z0]) Thus, the final ID shows that the PDA has accepted the string. 1.(b) Convert the following CFG to PDA and give the procedure for the same. S $\Rightarrow aABB[aA A A \Rightarrow aBB]aA A A \Rightarrow aBB]aA A A \Rightarrow aBB]aA A A \Rightarrow aBB]aA A A = aA A (to aA A = aA A) (to aA A) = (q, 0, s) \beta(q0,c, Z0) = (q0, S) (eq0,c, X) = (q0, aABB) \delta(q0,c, S) = (q0, aAABB) (eq0,c, S) = (q0, aABB) (eq0,c, S) = (q0, aAABB) (eq0,c, S) = (q0, aAABB) (eq0,c, S) = (q0, aAAB) (eq0,c, S) = (q0, aAABB) (eq0,c, A) = (q0, aBB) (eq0,c, A) = (q0, a) \delta(q0,c, A) = (q0, aABB) For A → aA A (to aA → aA A) (to aA → a A) (to aA → a) (eq0, a) (eq0, a) (eq0, a) (eq0, c, B) = (q0, A) (eq0, c, C) = (q0, A) (eq0, c, B) = (q0, A) (eq0, c, $		·			
$ \begin{array}{c c c c b,b,Z0 (q, 1, [epsilon, [b, b, Z, 0])(q1,c,[b,b,Z0) & \circ State: q1q_1q1 & \circ Input: ejepsilone & \circ State: [b,b,Z0 [b, b, Z, 0][b,b,Z0] & 0. After reading the first bbb: \\ (q1,c,[b,Z0])(q_1, [epsilon, [b, Z, 0])(q1,c,[b,Z0]) & \circ State: q1q_1q1 & \circ Input: ejepsilonc & \circ Stack: [b,Z0][b,Z0] & 10. After reading the second bbb: \\ (q1,c,[20])(q_1, [epsilon, [Z, 0])(q1,c,[Z0]) & \circ State: qfq_1qf & \circ Input: ejepsilone & \circ Stack: [b,Z0](b,Z)] & \circ State: qfq_1qf & \circ Input: ejepsilone & \circ Stack: [20][Z,0](20] & \circ State: qf0_2qf & \circ gf(q, q, qf) & \circ gf(q, qf) & \circ $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
9. After reading the first bbb: (q1, c[b,Z0])(q1, i;epsilon, [b, Z_0](p], c[b,Z0]) 		1 1			
$ \begin{array}{c c c b,20](q_1, epsilon, b, Z_0)(q_1,c_bZ_0)) \\ & \circ State: q q_1q \\ & \circ Input: (epsilon \\ & \circ Stack: [b,Z0][b, Z_0][b,Z0] \\ & 10. After reading the second bbb: \\ (qfc,C[Z0])(q, f_, epsilon, [Z_0]0(qf,c_bZ0) \\ & \circ State: qfq_fqf \\ & \circ Input: (epsilon \\ & \circ Stack: [Z0][Z_0][Z0] \\ \hline Thus, the final ID shows that the PDA has accepted the string. \\ & 1.(b) Convert the following CFG to PDA and give the procedure for the same. \\ & S \Rightarrow AABB aA \\ A \Rightarrow aBB a \\ B \Rightarrow bBB A \\ C \Rightarrow a \\ \hline Initialization: Start with the initial stack symbol Z0Z_0Z0. \\ & (q0,c,Z0)=(q0,S)(delta(q_0, epsilon, Z_0) = (q_0, S)\delta(q0,c,Z0)=(q0,S) \\ \hline For S \rightarrow aABB (xo aABBS \rightarrow aABB: \\ & \delta(q0,c,S)=(q0,aAB)(delta(q_0, epsilon, S) = (q_0, aABB)\delta(q0,c,S)=(q0,aABB) \\ \hline For A \rightarrow aBBA (to aABS \rightarrow aAB: \\ & \delta(q0,c,A)=(q0,a)AB)(delta(q_0, epsilon, S) = (q_0, aAA)\delta(q0,c,S)=(q0,aAA) \\ & \Box For A \rightarrow aBBA (to aBBA \rightarrow aBB: \\ & \delta(q0,c,A)=(q0,a)AB)(delta(q_0, epsilon, A) = (q_0, aAA)\delta(q0,c,A)=(q0,aBB) \\ & \Box For A \rightarrow aBB(to aBBA \rightarrow aBB: \\ & \delta(q0,c,A)=(q0,a)BB)(delta(q_0, epsilon, A) = (q_0, a)AB(q0,c,A)=(q0,aBB) \\ & \Box For A \rightarrow aBB(to aBBA \rightarrow aBB: \\ & \delta(q0,c,A)=(q0,a)BB)(delta(q_0, epsilon, B) = (q_0, a)A(q0,c,A)=(q0,aBB) \\ & \Box For A \rightarrow aBB(to aBBA \rightarrow aBB: \\ & \delta(q0,c,A)=(q0,a)BB(b)delta(q_0, epsilon, B) = (q_0, a)A(q0,c,A)=(q0,aBB) \\ & \Box For B \rightarrow AB (to AB A \rightarrow A: \\ & \delta(q0,c,B)=(q0,A)(delta(q_0, epsilon, B) = (q_0, a)A(q0,c,B)=(q0,A) \\ & \Box For C \rightarrow aC (to aC \rightarrow a: \\ & \delta(q0,c,C)=(q0,a)(delta(q_0, epsilon, C) = (q_0, a)\delta(q0,c,C)=(q0,a) \\ \end{array} \right) $					
$ \begin{array}{ c c c c c } & \circ & \text{State: } q _{d} q \\ & \circ & \text{Input: } c \langle \text{epsilone} \\ & \circ & \text{Stack: } [b,Z0][b,Z.0][b,Z0] \\ \hline 10. \text{ After reading the second bbb:} \\ (qf,c,[Z0])(q,f,(epsilon, [Z,0])(qf,c,[Z0]) \\ & \circ & \text{State: } qf_{d} f \\ & \circ & \text{Input: } c \langle \text{epsilone} \\ & \circ & \text{State: } [20][Z,0][Z0] \\ \hline \text{Thus, the final ID shows that the PDA has accepted the string.} \\ \hline 1.(b) & \text{Convert the following CFG to PDA and give the procedure for the same.} \\ & \Rightarrow ABB _{a} \\ & \Rightarrow \Rightarrow ABB _{a} \\ & \Rightarrow \Rightarrow BBB _{a} \\ & \mathbb{C} \rightarrow a \\ \hline & \text{Initialization: Start with the initial stack symbol Z0Z_0Z_0.} \\ & \delta(q0,c,Z0) = (q0,S) \langle \text{delta}(q,0, \langle \text{epsilon, } Z_0) = (q_0, S) \delta(q0,c,Z) = (q0,S) \\ & \text{For } S \rightarrow aABB \ (to aABS \rightarrow aABB: \\ & \delta(q0,c,S) = (q0,aABB) \langle \text{delta}(q,0, \langle \text{epsilon, } S) = (q_0, aABB) \delta(q0,c,S) = (q0,aABB) \\ & \text{For } S \rightarrow aAAS \ (to aAAS \rightarrow aAA: \\ & \delta(q0,c,S) = (q0,aA) \langle \text{delta}(q,0, \langle \text{epsilon, } S) = (q_0, aABB) \delta(q0,c,S) = (q0,aAA) \\ & \text{For } S \rightarrow aABB \ (to aABB \rightarrow aBB: \\ & \delta(q0,c,A) = (q0,aBB) \langle \text{delta}(q,0, \langle \text{epsilon, } A) = (q_0, aBB) \delta(q0,c,A) = (q0,aBB) \\ & \text{For } S \rightarrow aAB(to aA \rightarrow aBB: \\ & \delta(q0,c,A) = (q0,aBB) \langle \text{delta}(q,0, \langle \text{epsilon, } A) = (q_0,aBB) \delta(q0,c,A) = (q0,aBB) \\ & \text{For } S \rightarrow aAB(to aA \rightarrow a: \\ & \delta(q0,c,B) = (q0,A) \langle \text{delta}(q,0, \langle \text{epsilon, } B) = (q_0,a) \delta(q0,c,A) = (q0,aBB) \\ & \text{For } B \rightarrow AB \ (to a AB \rightarrow a: \\ & \delta(q0,c,B) = (q0,A) \langle \text{delta}(q,0, \langle \text{epsilon, } B) = (q_0,a) \delta(q0,c,B) = (q0,A) \\ & \text{For } S \rightarrow AB(to aA \rightarrow a: \\ & \delta(q0,c,C) = (q0,a) \langle \text{delta}(q,0, \langle \text{epsilon, } B) = (q_0,a) \delta(q0,c,B) = (q0,A) \\ & \text{For } S \rightarrow AB \ (to a A \rightarrow a: \\ & \delta(q0,c,C) = (q0,a) \langle \text{delta}(q,0, \langle \text{epsilon, } B) = (q_0,a) \delta(q0,c,B) = (q0,A) \\ & \text{For } S \rightarrow AB \ (to a A \rightarrow a: \\ & \delta(q0,c,C) = (q0,a) \langle \text{delta}(q,0, \langle \text{epsilon, } C) = (q_0,a) \delta(q0,c,C) = (q0,a) \\ & \text{For } S \rightarrow AB \ (to A \rightarrow a: \\ & \delta(q0,c,C) = (q0,a) \langle \text{delta}(q,0, \langle \text{epsilon, } C) = (q_0,a) \delta(q0,c,C) = (q0,a) \\ & \text{For } S \rightarrow AB \ (to A \rightarrow a: \\ & \delta(q0,c,C) = (q0,a) \langle \text{delta}(q,0, \langle \text{epsilon, } C) = (q_0,a) \delta(q$		8			
$\begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
$ \begin{array}{ c c c c c } & & & & & & & & & & & & & & & & & & &$		10. After reading the second bbb:			
$ \begin{array}{ c c c c c } & & & & & & & & & & & & & & & & & & &$		$(qf,\epsilon,[Z0])(q f, epsilon, [Z_0])(qf,\epsilon,[Z0])$			
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
oStack: [Z0][Z_0][Z_0]Thus, the final ID shows that the PDA has accepted the string.1.(b)Convert the following CFG to PDA and give the procedure for the same.S → aABB aAAA→aBB aB→bBB AC→a□Initialization: Start with the initial stack symbol Z0Z_0Z0. $\delta(q0,c,Z0)=(q0,S)$ (delta(q_0, vepsilon, Z_0) = (q_0, S)\delta(q0,c,Z0)=(q0,S)□For S→aABBS \to aABS→aABB: $\delta(q0,c,S)=(q0,aAB)$ (delta(q_0, vepsilon, S) = (q_0, aABB)\delta(q0,c,S)=(q0,aABB)□For S→aAAAS \to aAAS→aAA: $\delta(q0,c,S)=(q0,aAB)$ (delta(q_0, vepsilon, S) = (q_0, aABB)\delta(q0,c,S)=(q0,aAA)□For A→aBA \to aBA→aBB: $\delta(q0,c,S)=(q0,aAB)$ (delta(q_0, vepsilon, A) = (q_0, aAB)\delta(q0,c,A)=(q0,aBB)□For A→aA \to aA→a: $\delta(q0,c,A)=(q0,a)$ (delta(q_0, vepsilon, A) = (q_0, a)\delta(q0,c,A)=(q0,a)□For B→BBB \to bBBB: $\delta(q0,c,B)=(q0,bBB)$ (delta(q_0, vepsilon, B) = (q_0, bBB)\delta(q0,c,B)=(q0,bBB)□□For C→aC \to aC→a: $\delta(q0,c,B)=(q0,a)$ (delta(q_0, vepsilon, B) = (q_0, a)\delta(q0,c,C)=(q0,a)2Define CFG and CNF. Convert the given CFG to CNF $\delta(q0,c,C)=(q0,a)$ (delta(q_0, vepsilon, C) = (q_0, a)\delta(q0,c,C)=(q0,a)2Define CFG and CNF. Convert the given CFG to CNF $\delta(q0,c,C)=(q0,a)$ (delta(q_0, vepsilon, C) = (q_0, a)\delta(q0,c,C)=(q0,a) 2^{-} Define CFG and CNF. Convert the given CFG to CNF $\delta(q0,c,C)=(q0,a)$ (delta(q_0, vepsilon, C) = (q_0, a)\delta(q0,c,C)=(q0,a) 2^{-} Define CFG and CNF. Convert the given CFG to CNF					
Thus, the final ID shows that the PDA has accepted the string.Image: Construction of the same in the following CFG to PDA and give the procedure for the same.4CO3L3S \Rightarrow aABB aAAA \Rightarrow aBB aB \Rightarrow bBB AC \Rightarrow aImage: Construction of the same.4CO3L3B \Rightarrow bBB AC \Rightarrow aImage: Construction of the same.4CO3L3C \Rightarrow aImage: Construction of the same.4CO3L3B \Rightarrow bBB AC \Rightarrow aImage: Construction of the same.4CO3L3C \Rightarrow aImage: Construction of the same.4CO3L3B \Rightarrow bBB AC \Rightarrow aImage: Construction of the same.4CO3L3C \Rightarrow aImage: Construction of the same.4CO3L3B \Rightarrow bBB AC \Rightarrow aImage: Construction of the same.4CO3L3B \Rightarrow bBBB AC \Rightarrow aImage: Construction of the same.4CO3L3B $(q0, c, C)=(q0, S) (delta(q_0, lepsilon, Z_0) = (q_0, S) (q0, c, C)=(q0, aABB) (q0, c, S)=(q0, aABB) (delta(q_0, lepsilon, S) = (q_0, aAA) (q0, c, S)=(q0, aABB) (delta(q_0, lepsilon, A) = (q_0, a) (q0, c, A)=(q0, aBB) (delta(q_0, lepsilon, A) = (q_0, a) (q0, c, A)=(q0, a) (q0, c, B)=(q0, A) (q0, c, B)=(q0, A) (q0, c, B)=(q0, A) (q0, c, B)=(q0, A) (q0, c, C)=(q0, a) (q0, c, C)=(q0, a) (q0, c, C)=(q0, a) (delta(q_0, lepsilon, C) = (q_0, a) (q0, c, C)=(q0, a) (q0, c, C)=(q0, a) (q0, c, C)=(q0, a) (delta(q_0, lepsilon, C) = (q_0, a) (q0, c, C)=(q0, a) (delta(q_0, lepsilon, C) = (q_0, a) (q0, c, $					
1.(b)Convert the following CFG to PDA and give the procedure for the same.4CO3L3 $S \rightarrow aABB aAA$ $A \rightarrow aBB a$ $B \rightarrow bBB A$ $C \rightarrow a$ 1Initialization: Start with the initial stack symbol Z0Z_0Z0. $\delta(q0,\epsilon,Z0)=(q0,S)(delta(q_0, \epsilon, Z_0) = (q_0, S)\delta(q0,\epsilon,Z0)=(q0,S)$ \Box For $S \rightarrow aABBS$ to $aABBS \rightarrow aABB$: $\delta(q0,\epsilon,S)=(q0,aABB)(delta(q_0, \epsilon, S) = (q_0, aABB)\delta(q0,\epsilon,S)=(q0,aABB)$ \Box For $S \rightarrow aABC$ to $aBBA \rightarrow aAB$: $\delta(q0,\epsilon,S)=(q0,aAA)(delta(q_0, \epsilon, S) = (q_0, aABB)\delta(q0,\epsilon,S)=(q0,aAA)$ \Box For $A \rightarrow aBA$ to $aBA \rightarrow aaBB$: $\delta(q0,\epsilon,A)=(q0,aBB)(delta(q_0, \epsilon, A) = (q_0, aBB)\delta(q0,\epsilon,A)=(q0,aBB)$ \Box For $A \rightarrow aBA$ to $aA \rightarrow a$: $\delta(q0,\epsilon,A)=(q0,a)(delta(q_0, \epsilon, A) = (q_0, a)BB(q0,\epsilon,A)=(q0,aBB)$ \Box For $B \rightarrow bBB (to BBB \rightarrow bBB)$: $\delta(q0,\epsilon,B)=(q0,bBB)(delta(q_0, \epsilon, B) = (q_0, bBB)\delta(q0,\epsilon,B)=(q0,bBB)$ \Box For $B \rightarrow aB$ to $AB \rightarrow A$: $\delta(q0,\epsilon,C)=(q0,a)(delta(q_0, \epsilon, B) = (q_0, a)\delta(q0,\epsilon,B)=(q0,bBB)$ \Box For $C \rightarrow aC$ to $aC \rightarrow a$: $\delta(q0,\epsilon,C)=(q0,a)(delta(q_0, \epsilon, C) = (q_0, a)\delta(q0,\epsilon,C)=(q0,a)$ 10CO3L1. L32Define CFG and CNF. Convert the given CFG to CNF $A \rightarrow A BaC aaaB \rightarrow bBb a DC \rightarrow CA AC10CO3L1.L3$					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	1(h)		1	CO3	13
$\begin{array}{ c c c c c c } A \rightarrow aBB _{a}^{a} \\ B \rightarrow bBB _{A} \\ C \rightarrow a \\ \hline Initialization: Start with the initial stack symbol Z0Z_0Z0. \\ & & & & & & & & & & & & & & & & & & $	1.(0)		-	005	LJ
$ \begin{array}{ c c c c c } & B \rightarrow bBB A \\ C \rightarrow a & & \\ \hline & Initialization: Start with the initial stack symbol Z0Z_0Z0. \\ & \delta(q0,\epsilon,Z0)=(q0,S) \langle delta(q_0, \langle epsilon, Z_0)=(q_0,S)\delta(q0,\epsilon,Z0)=(q0,S) \\ \hline & For S \rightarrow aABBS \langle to aABBS \rightarrow aABB: \\ & \delta(q0,\epsilon,S)=(q0,aAB) \langle delta(q_0, \langle epsilon, S)=(q_0, aABB)\delta(q0,\epsilon,S)=(q0,aABB) \\ \hline & For S \rightarrow aAAS \langle to aAAS \rightarrow aAA: \\ & \delta(q0,\epsilon,S)=(q0,aBB) \langle delta(q_0, \langle epsilon, S)=(q_0, aAA)\delta(q0,\epsilon,S)=(q0,aAA) \\ \hline & For A \rightarrow aBBA \langle to aBBA \rightarrow aBB: \\ & \delta(q0,\epsilon,A)=(q0,aBB) \langle delta(q_0, \langle epsilon, A)=(q_0, aBB)\delta(q0,\epsilon,A)=(q0,aBB) \\ \hline & For A \rightarrow aA \langle to aA \rightarrow a: \\ & \delta(q0,\epsilon,A)=(q0,a) \langle delta(q_0, \langle epsilon, A)=(q_0, a)\delta(q0,\epsilon,A)=(q0,a) \\ \hline & For B \rightarrow bBBB \langle to bBBB \rightarrow bBB: \\ & \delta(q0,\epsilon,B)=(q0,bBB) \langle delta(q_0, \langle epsilon, B)=(q_0,bBB)\delta(q0,\epsilon,B)=(q0,bBB) \\ \hline & For C \rightarrow aC \langle to aC \rightarrow a: \\ & \delta(q0,\epsilon,C)=(q0,a) \langle delta(q_0, \langle epsilon, C)=(q_0,a)\delta(q0,\epsilon,C)=(q0,a) \\ \end{array} $					
$\begin{array}{ c c c c c } \hline C \rightarrow a & & & & \\ \hline Initialization: Start with the initial stack symbol Z0Z_0Z0. \\ & & \\ & $					
$ \begin{array}{ c c c c c } \hline \textbf{Initialization: Start with the initial stack symbol Z0Z_0Z0.} \\ & \delta(q0,\epsilon,Z0)=(q0,S) \langle delta(q_0, \langle epsilon, Z_0) = (q_0, S)\delta(q0,\epsilon,Z0)=(q0,S) \\ & & \textbf{For S} \rightarrow \textbf{aABBS} \backslash \textbf{to aABBS} \rightarrow \textbf{aABB:} \\ & \delta(q0,\epsilon,S)=(q0,aABB) \langle delta(q_0, \langle epsilon, S) = (q_0, aABB)\delta(q0,\epsilon,S)=(q0,aABB) \\ & & \textbf{For S} \rightarrow \textbf{aAAS} \backslash \textbf{to aAAS} \rightarrow \textbf{aAA:} \\ & \delta(q0,\epsilon,S)=(q0,aAA) \langle delta(q_0, \langle epsilon, S) = (q_0, aAA)\delta(q0,\epsilon,S)=(q0,aAA) \\ & & \textbf{For A} \rightarrow \textbf{aBBA} \backslash \textbf{to aBBA} \rightarrow \textbf{aBB:} \\ & \delta(q0,\epsilon,A)=(q0,aBB) \langle delta(q_0, \langle epsilon, A) = (q_0, aBB)\delta(q0,\epsilon,A)=(q0,aBB) \\ & & \textbf{For A} \rightarrow \textbf{aA} \backslash \textbf{to aA} \rightarrow \textbf{a:} \\ & \delta(q0,\epsilon,A)=(q0,a)BB \backslash \textbf{to aBB} \rightarrow \textbf{bBB:} \\ & \delta(q0,\epsilon,A)=(q0,a)BB \backslash \textbf{to aB} \rightarrow \textbf{bBB:} \\ & \delta(q0,\epsilon,A)=(q0,bBB) \backslash \textbf{delta}(q_0, \langle epsilon, B) = (q_0, a)\delta(q0,\epsilon,A)=(q0,a) \\ & & \textbf{For B} \rightarrow \textbf{AB} \backslash \textbf{to AB} \rightarrow \textbf{A:} \\ & \delta(q0,\epsilon,B)=(q0,bBB) \backslash \textbf{delta}(q_0, \langle epsilon, B) = (q_0, a)\delta(q0,\epsilon,B)=(q0,bBB) \\ & & \textbf{For C} \rightarrow \textbf{aC} \backslash \textbf{to aC} \rightarrow \textbf{a:} \\ & \delta(q0,\epsilon,C)=(q0,a) \backslash \textbf{delta}(q_0, \langle epsilon, C) = (q_0, a)\delta(q0,\epsilon,C)=(q0,a) \\ \end{array} \right. $					
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$		C ≠a			
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$					
$ \begin{bmatrix} For S \rightarrow aABBS \ to \ aABBS \rightarrow aABB: \\ \delta(q0,\epsilon,S)=(q0,aABB)\ delta(q_0, \ epsilon, S) = (q_0, aABB)\delta(q0,\epsilon,S)=(q0,aABB) \\ Bror S \rightarrow aAAS \ to \ aAAS \rightarrow aAA: \\ \delta(q0,\epsilon,S)=(q0,aAA)\ delta(q_0, \ epsilon, S) = (q_0, aAA)\delta(q0,\epsilon,S)=(q0,aAA) \\ Bror A \rightarrow aBBA \ to \ aBBA \rightarrow aBB: \\ \delta(q0,\epsilon,A)=(q0,aBB)\ delta(q_0, \ epsilon, A) = (q_0, aBB)\delta(q0,\epsilon,A)=(q0,aBB) \\ Bror A \rightarrow aA \ to \ aA \rightarrow a: \\ \delta(q0,\epsilon,A)=(q0,a)\ delta(q_0, \ epsilon, A) = (q_0, a)\delta(q0,\epsilon,A)=(q0,aBB) \\ Bror B \rightarrow bBBB\ to \ bBBB \rightarrow bBB: \\ \delta(q0,\epsilon,B)=(q0,bBB)\ delta(q_0, \ epsilon, B) = (q_0, bBB)\delta(q0,\epsilon,B)=(q0,bBB) \\ Bror B \rightarrow AB\ to \ AB \rightarrow A: \\ \delta(q0,\epsilon,B)=(q0,A)\ delta(q_0, \ epsilon, B) = (q_0, A)\delta(q0,\epsilon,B)=(q0,A) \\ Bror C \rightarrow aC\ to \ aC \rightarrow a: \\ \delta(q0,\epsilon,C)=(q0,a)\ delta(q_0, \ epsilon, C) = (q_0, a)\delta(q0,\epsilon,C)=(q0,a) \\ \hline 2 Define\ CFG\ and\ CNF.\ Convert\ the\ given\ CFG\ to\ CNF \\ S \rightarrow ABC BaB \\ A \rightarrow aA BaC aaa \\ B \rightarrow bBb a D \\ C \rightarrow CA AC \\ \hline \end{tabular}$		•			
$ \begin{array}{ c c c c c } \delta(q0,\epsilon,S) = (q0,aABB) \ (delta(q_0, \ (epsilon, S) = (q_0, aABB) \delta(q0,\epsilon,S) = (q0,aABB) \\ \hline & For S \rightarrow aAAS \ to aAAS \rightarrow aAA: \\ \delta(q0,\epsilon,S) = (q0,aAA) \ (delta(q_0, \ (epsilon, S) = (q_0, aAA) \delta(q0,\epsilon,S) = (q0,aAA) \\ \hline & For A \rightarrow aBBA \ to aBBA \rightarrow aBB: \\ \delta(q0,\epsilon,A) = (q0,aBB) \ (delta(q_0, \ (epsilon, A) = (q_0, aBB) \delta(q0,\epsilon,A) = (q0,aBB) \\ \hline & For A \rightarrow aA \ to aA \rightarrow a: \\ \delta(q0,\epsilon,A) = (q0,a) \ (delta(q_0, \ (epsilon, A) = (q_0, a) \delta(q0,\epsilon,A) = (q0,a) \\ \hline & For B \rightarrow bBBB \ to bBB \rightarrow bBB: \\ \delta(q0,\epsilon,B) = (q0,bBB) \ (delta(q_0, \ (epsilon, B) = (q_0, bBB) \delta(q0,\epsilon,B) = (q0,bBB) \\ \hline & For B \rightarrow AB \ to \ AB \rightarrow A: \\ \delta(q0,\epsilon,B) = (q0,A) \ (delta(q_0, \ (epsilon, B) = (q_0, A) \delta(q0,\epsilon,B) = (q0,A) \\ \hline & For C \rightarrow aC \ (to \ aC \rightarrow a: \\ \delta(q0,\epsilon,C) = (q0,a) \ (delta(q_0, \ (epsilon, C) = (q_0, a) \delta(q0,\epsilon,C) = (q0,a) \\ \end{array} $		$\delta(q0,\epsilon,Z0) = (q0,S) \setminus delta(q_0, \forall epsilon, Z_0) = (q_0, S) \delta(q0,\epsilon,Z0) = (q0,S)$			
$ \begin{bmatrix} \mathbf{For } \mathbf{S} \rightarrow \mathbf{aAAS } \text{ to } \mathbf{aAAS} \rightarrow \mathbf{aAA} : \\ \delta(q0, \epsilon, S) = (q0, aAA) \land (q0, \epsilon, S) = (q0, aAA) \delta(q0, \epsilon, S) = (q0, aAA) \\ \hline \mathbf{For } \mathbf{A} \rightarrow \mathbf{aBBA } \text{ to } \mathbf{aBBA} \rightarrow \mathbf{aBB} : \\ \delta(q0, \epsilon, A) = (q0, aBB) \land (q0, \epsilon, A) = (q0, aBB) \delta(q0, \epsilon, A) = (q0, aBB) \\ \hline \mathbf{For } \mathbf{A} \rightarrow \mathbf{aA} \text{ to } \mathbf{aA} \rightarrow \mathbf{a} : \\ \delta(q0, \epsilon, A) = (q0, a) \land (q0, \epsilon, A) = (q0, a) \delta(q0, \epsilon, A) = (q0, a) \\ \hline \mathbf{For } \mathbf{B} \rightarrow \mathbf{bBBB} \text{ to } \mathbf{bBBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, bBB) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, bBB) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, bBB) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, A) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, A) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, A) \land \mathbf{bBB} \rightarrow \mathbf{bBB} : \\ \delta(q0, \epsilon, B) = (q0, A) \land \mathbf{bBB} = (q0, A) \land (q0, \epsilon, B) = (q0, A) \\ \hline \mathbf{For } \mathbf{C} \rightarrow \mathbf{aC} \text{ to } \mathbf{aC} \rightarrow \mathbf{a} : \\ \delta(q0, \epsilon, C) = (q0, a) \land (q0, \epsilon, C) = (q0, a) \\ \hline \mathbf{C} \rightarrow \mathbf{cA} \mathbf{baBB} \mathbf{aBBB} a$					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $		$\delta(q0,\epsilon,S) = (q0,aABB) \setminus delta(q_0, \forall epsilon, S) = (q_0, aABB) \delta(q0,\epsilon,S) = (q0,aABB)$			
$ \begin{bmatrix} For A \rightarrow aBBA \ to \ aBBA \rightarrow aBB: \\ \delta(q0, \epsilon, A) = (q0, aBB) \ delta(q_0, \ epsilon, A) = (q_0, aBB) \\ \delta(q0, \epsilon, A) = (q0, a)B) \ delta(q_0, \ epsilon, A) = (q_0, a)BB \ delta(q0, a) \\ \hline For A \rightarrow aA \ to \ aA \rightarrow a: \\ \delta(q0, \epsilon, A) = (q0, a) \ delta(q_0, \ epsilon, A) = (q_0, a) \\ \delta(q0, \epsilon, B) = (q0, bBB) \ delta(q_0, \ epsilon, B) = (q_0, bBB) \\ \delta(q0, \epsilon, B) = (q0, A) \ delta(q_0, \ epsilon, B) = (q_0, A) \\ \hline For C \rightarrow aC \ to \ aC \rightarrow a: \\ \delta(q0, \epsilon, C) = (q0, a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \delta(q0, \epsilon, C) = (q0, a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \hline 2 Define \ CFG \ and \ CNF. \ Convert \ the \ given \ CFG \ to \ CNF \\ S \rightarrow ABC BaB \\ A \rightarrow aA BaC aaa \\ B \rightarrow bBb a D \\ C \rightarrow CA AC \\ \hline \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$		$\Box \text{ For } S \rightarrow aAAS \setminus to \ aAAS \rightarrow aAA:$			
$ \begin{bmatrix} For A \rightarrow aBBA \ to \ aBBA \rightarrow aBB: \\ \delta(q0, \epsilon, A) = (q0, aBB) \ delta(q_0, \ epsilon, A) = (q_0, aBB) \\ \delta(q0, \epsilon, A) = (q0, a)B) \ delta(q_0, \ epsilon, A) = (q_0, a)BB \ delta(q0, a) \\ \hline For A \rightarrow aA \ to \ aA \rightarrow a: \\ \delta(q0, \epsilon, A) = (q0, a) \ delta(q_0, \ epsilon, A) = (q_0, a) \\ \delta(q0, \epsilon, B) = (q0, bBB) \ delta(q_0, \ epsilon, B) = (q_0, bBB) \\ \delta(q0, \epsilon, B) = (q0, A) \ delta(q_0, \ epsilon, B) = (q_0, A) \\ \hline For C \rightarrow aC \ to \ aC \rightarrow a: \\ \delta(q0, \epsilon, C) = (q0, a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \delta(q0, \epsilon, C) = (q0, a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \hline 2 Define \ CFG \ and \ CNF. \ Convert \ the \ given \ CFG \ to \ CNF \\ S \rightarrow ABC BaB \\ A \rightarrow aA BaC aaa \\ B \rightarrow bBb a D \\ C \rightarrow CA AC \\ \hline \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$		$\delta(q0,\epsilon,S) = (q0,aAA) \setminus delta(q_0, epsilon, S) = (q 0, aAA) \delta(q0,\epsilon,S) = (q0,aAA)$			
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $		\Box For A \rightarrow aBBA \to aBBA \rightarrow aBB:			
$ \begin{bmatrix} For A \rightarrow aA \ to aA \rightarrow a: \\ \delta(q0,\epsilon,A)=(q0,a)\ delta(q_0, \ epsilon, A) = (q_0, a)\delta(q0,\epsilon,A)=(q0,a) \\ \\ \hline For B \rightarrow bBBB\ to bBBB \rightarrow bBB: \\ \delta(q0,\epsilon,B)=(q0,bBB)\ delta(q_0, \ epsilon, B) = (q_0, bBB)\delta(q0,\epsilon,B)=(q0,bBB) \\ \\ \hline For B \rightarrow AB\ to AB \rightarrow A: \\ \delta(q0,\epsilon,B)=(q0,A)\ delta(q_0, \ epsilon, B) = (q_0, A)\delta(q0,\epsilon,B)=(q0,A) \\ \\ \hline For C \rightarrow aC\ to aC \rightarrow a: \\ \delta(q0,\epsilon,C)=(q0,a)\ delta(q_0, \ epsilon, C) = (q_0, a)\delta(q0,\epsilon,C)=(q0,a) \\ \end{bmatrix} $					
$\begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{bmatrix} For B \rightarrow bBBB \ to \ bBBB \rightarrow bBB: \\ \delta(q0,\epsilon,B) = (q0,bBB) \ delta(q_0, \ epsilon, B) = (q_0, bBB) \\ \delta(q0,\epsilon,B) = (q0,A) \ delta(q_0, \ epsilon, B) = (q_0, A) \\ \delta(q0,\epsilon,B) = (q0,A) \ delta(q_0, \ epsilon, B) = (q_0, A) \\ \delta(q0,\epsilon,C) = (q0,a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \delta(q0,\epsilon,C) = (q0,a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \delta(q0,\epsilon,C) = (q0,a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ 10 CO3 L1, \\ L3 L3 L3 L3 L3 L3 L3 \\ A \rightarrow aA BaC aaa \\ B \rightarrow bBb a D \\ C \rightarrow CA AC C C C C C C C C C $		•			
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
$ \begin{bmatrix} \mathbf{For B} \rightarrow \mathbf{AB} \setminus \mathbf{to AB} \rightarrow \mathbf{A}: \\ \delta(q0,\epsilon,B) = (q0,A) \setminus \det(q_0, epsilon, B) = (q_0, A) \delta(q0,\epsilon,B) = (q0,A) \\ \Box \mathbf{For C} \rightarrow \mathbf{aC} \setminus \mathbf{to aC} \rightarrow \mathbf{a}: \\ \delta(q0,\epsilon,C) = (q0,a) \setminus \det(q_0, epsilon, C) = (q_0, a) \delta(q0,\epsilon,C) = (q0,a) \\ 2 & \text{Define CFG and CNF. Convert the given CFG to CNF} \\ S \rightarrow ABC \mid BaB \\ A \rightarrow aA \mid BaC \mid aaa \\ B \rightarrow bBb \mid a\mid D \\ C \rightarrow CA \mid AC \\ \end{bmatrix} $					
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $					
□For C→aC \to aC→a: $\delta(q0,\epsilon,C)=(q0,a) \ delta(q_0, \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $					
$ \delta(q0,\epsilon,C) = (q0,a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ \delta(q0,\epsilon,C) = (q0,a) \ delta(q_0, \ epsilon, C) = (q_0, a) \\ 0 \ CO3 \ L1, \\ S \rightarrow ABC BaB \\ A \rightarrow aA BaC aaa \\ B \rightarrow bBb a D \\ C \rightarrow CA AC \ delta(C) = (q0,a) \\ 0 \ C = (q0,a) \\ 0$					
2Define CFG and CNF. Convert the given CFG to CNF10CO3L1, $S \rightarrow ABC BaB$ $A \rightarrow aA BaC aaa$ $B \rightarrow bBb a D$ $C \rightarrow CA AC$ L1L3					
$S \rightarrow ABC BaB$ $A \rightarrow aA BaC aaa$ $B \rightarrow bBb a D$ $C \rightarrow CA AC$		$\delta(qU,\epsilon,C) = (qU,a) \det(q_U, epsilon, C) = (q_0, a)\delta(qU,\epsilon,C) = (qU,a)$			
$S \rightarrow ABC BaB$ $A \rightarrow aA BaC aaa$ $B \rightarrow bBb a D$ $C \rightarrow CA AC$				<i>a</i>	
$A \rightarrow aA BaC aaaB \rightarrow bBb a DC \rightarrow CA AC$	2	•	10	CO3	· · ·
B→bBb a D C→CA AC		S→ABC BaB			L3
B→bBb a D C→CA AC		A→aA BaC aaa			
C→CA AC					

 Context-Free Grammar (CFG) A Context-Free Grammar (CFG) is a formal grammar where every production rule is of the form: A→γA \to \gammaA→γ where AAA is a non-terminal symbol and γ\gammaγ is a string of terminals and/or non-terminals. In other words, the left-hand side of every production consists of a single non-terminal, and the right-hand side can be a string of terminals, non-terminals, or an empty string (denoted ε\epsilonε). A CFG is used to define the syntax of programming languages, natural languages, and other formal languages. Chomsky Normal Form (CNF) A Context-Free Grammar (CFG) is in Chomsky Normal Form (CNF) if all of its production rules satisfy one of the following conditions: A → BC, where AAA, BBB, and CCC are non-terminal symbols, and BBB and CCC are not the start symbol. A → a, where AAA is a non-terminal symbol and the production is allowed only if the language generated by the grammar includes the empty string. In CNF, all productions must either have two non-terminals on the right-hand side or a single terminal symbol. Converting CFG to CNF To convert a CFG to CNF, the following steps are typically followed: Eliminate <i>x</i>-productions (productions of the form A→€A \to \epsilonA→ε, except for the start symbol (productions of the form A→€A \to \epsilonA→ε, except for the start symbol (productions of the form A→EA \to BA→B, where both AAA and BBB are non-terminals). Eliminate unit productions into binary form (i.e., ensure that every production is either of the form A→BCA \to BCA→BC or A→aA \to aA→a, where aaa is a terminal). 			
 a) State and prove pumping Lemma for context free languages. The Pumping Lemma for Context-Free Languages is a property that all context-free languages must satisfy. It is used primarily to prove that certain languages are not context-free by showing they do not meet this property. Statement of the Pumping Lemma for CFLs: If a language LLL is context-free, then there exists a pumping length ppp such that 	5	CO3	L1, L2

		grammar GGG.			
	2.	Suppose GGG has nnn non-terminal symbols.			
	3.	Let $p=2np = 2^np=2n$. This is the pumping length.			
	4.	Consider any string $w \in Lw \setminus in Lw \in L$ with $ w \ge p w \setminus geq p w \ge p$. The parse			
		tree for www has a height of at least n+1n+1n+1 due to the pigeonhole			
		principle, since there are more symbols in the string than non-terminals in the grammar.			
	5.	By the pigeonhole principle, some non-terminal symbol must appear more than once along a path from the root to a leaf in the parse tree.			
	6.	Let this repeated non-terminal be AAA. Consider the substring generated by AAA on the first and second occurrences.			
	7.	Decompose w=uvxyzw = uvxyzw=uvxyz, where:			
		 vvv and yyy are the substrings derived from repeated occurrences of AAA. 			
		\circ xxx is the part between these substrings.			
	8.	Since the grammar is context-free, replacing the repeated occurrences of			
		AAA by producing more or fewer copies still generates valid strings in			
		LLL.			
	9.	Thus, for any integer kkk, the string $u(vk)x(yk)z\in Lu(v^k)x(y^k)z \setminus in$			
		$Lu(vk)x(yk)z\in L.$			
2 (1)		roves the Pumping Lemma for CFLs.		COA	x 0
3.(b)	Prove	that language L= $\{a^nb^nc^n n \ge 1\}$ is not context free.	5	CO3	L3
	1	Assume LLL is context-free.			
		Let the pumping length be ppp.			
		Consider the string w=apbpcpw = $a^p b^p c^pw=apbpcp$.			
		Decompose w=uvxyzw = uvxyzw=uvxyz, where $ vxy \le p vxy \le p vxy \le p$			
		and $ vy \ge 1 vy \ge 1$.			
	5.	The substring vxyvxyvxy can contain symbols from at most two of the three			
		blocks apa^pap, bpb^pbp, and cpc^pcp, because its length is at most ppp.			
	6.	Pumping vvv and yyy will unbalance the counts of at least two symbols,			
		causing the string to fall out of the language.			
	This c	ontradicts the Pumping Lemma, so LLL is not context-free			
4	Consti	ruct Turing Machine to accept the language $L = \{a^{2n}b^n n \ge 1\}$. Give the	10	CO4	L2,L
		ion table as well as Transition diagram of TM obtained.	10	007	3
	li ansit	ion tuble as well as fransition diagram of five obtained.			

	$R = \frac{1}{2}$			
5	 Write a short notes on: i) Turing Machine and its working Turing Machine and Its Working A Turing Machine (TM) is a theoretical computational model that defines an abstract machine capable of simulating any computer algorithm. It was introduced by Alan Turing in 1936 and serves as a foundation for the theory of computation. Components of a Turing Machine: 1. Tape: An infinite strip of cells, each capable of holding a symbol from a finite alphabet. 2. Head: A read/write head that can move left or right along the tape. 3. States: A finite set of states, including a start state and one or more 	10	CO4	L2
	 accepting or rejecting states. 4. Alphabet: A set of symbols, including a special blank symbol (usually denoted by ''). 5. Transition Function: A set of rules that define how the machine transitions between states based on the current symbol and state. 			
	 Working of a Turing Machine: Initialization: The machine starts in the initial state with the tape containing the input string, and the head positioned at the first symbol. Reading and Writing: The head reads the current symbol on the tape. Based on the symbol and the current state, the machine consults its transition function. Transition: The machine may: Write a new symbol on the current tape cell. Move the head left or right by one cell. Change to a new state. Halting: The machine halts when it reaches an accepting or rejecting state or 			
	 The machine halts when it reaches an accepting or rejecting state or if no transition is defined for the current configuration. 			





ey Properti	ies:
• •	sively Enumerable:
0	The Universal Language is recursively enumerable (RE) but not recursive.
0	A Turing Machine can simulate MMM on www; if MMM accepts www, the universal Turing Machine also accepts the encoded pair.
2. Non-I	Decidability:
0	The Universal Language is not decidable because determining whether an arbitrary Turing Machine accepts an input is equivalent to solving the Halting Problem , which is undecidable.
3. Closu	re Properties:
0	It is closed under union and intersection but not under complementation.

Faculty Signature

CCI Signature

HOD Signature