SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CONTINUOUS ASSESSMENT TEST - II

SLOT: B1 + TB1

WINTER SEMESTER 2023-2024

Programme Name & Branch : **B.Tech (CSE)**
Course Code : **BCSE304L**
Course Name : **Theory of Computation**

Duration: 90 min. **Key** **Max. Marks: 50**

1. (a)

Answer: Let $D_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $D_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs that recognize A and B, respectively. Here, we shall construct a DFA $D = (Q, \Sigma, \delta, q, F)$ that recognizes the perfect shuffle of A and B.

The key idea is to design D to alternately switch from running D_A and running D_B after each character is read. Therefore, at any time, D needs to keep track of (i) the current states of D_A and D_B and (ii) whether the next character of the input string should be matched in D_A or in D_B. Then, when a character is read, depending on which DFA should match the character, D makes a move in the corresponding DFA accordingly. After the whole string is processed, if both DFAs are in the accept states, the input string is accepted; otherwise, the input string is rejected.

Formally, the DFA D can be defined as follows:

(a) $Q = Q_A \times Q_B \times \{A, B\}$, which keeps track of all possible current states of D_A and D_B, and which DFA to match.

(b) $q = (q_A, q_B, A)$, which states that D starts with D_A in q_A, D_B in q_B, and the next character read should be in D_A.

(c) $F = F_A \times F_B \times \{A\}$, which states that D accepts the string if both D_A and D_B are in accept states, and the next character read should be in D_A (i.e., last character was read in D_B).

(d) δ is as follows:

i. $\delta((x, y, A), a) = (\delta_A(x, a), y, B)$, which states that if current state of D_A is x, the current state of D_B is y, and the next character read is in D_A, then when a is read as the next character, we should change the current state of A to $\delta_A(x, a)$, while the current state of B is not changed, and the next character read will be in D_B.

ii. Similarly, $\delta((x, y, B), b) = (x, \delta_B(y, b), A)$.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CONTINUOUS ASSESSMENT TEST - II

SLOT: B1 + TB1

WINTER SEMESTER 2023-2024

---

1. (b)

The proof is by contradiction. Assume L_4 is regular. Then by the Pumping Lemma, there is an integer $p > 0$, such that for every $w \in L_4$, $|w| \geq p$, there exists a partition of $w = xyz$, such that $|y| > 0$, $|xy| \leq p$, and for all i $xy^iz \in L_4$.

Let $w = 0^{p+1}1^p$ and let x, y, z strings satisfying the conditions of the pumping lemma. Since $|xy| \leq 0$ then $y = 0^k$ for some $k \geq 1$. By the Pumping Lemma $w_1 = xy^iz = x(0^k)^iz$ belongs to L_4 for any value of i. But setting $i = 0$ makes the number of 0's in w_1 always equal or less than the number of 1s and $w_1 \notin L_4$. Thus when $i = 0$, w_1 does not belong to L_4, contradicting the pumping lemma.

2. (a)

$S \rightarrow S_1 S_2$ (S_1 is used for ww^R and S_2 is used for zz^R)

$S_2 \rightarrow 0S_2'0 \mid 1S_2'1 \mid 00 \mid 11$

$S_2' \rightarrow 0S_20 \mid 1S_21$

$S_1 \rightarrow 0S_10 \mid 1S_11 \mid 011X110$

$X \rightarrow 0X0 \mid 1X1 \mid \in$

(b)

Given grammar G

$S \rightarrow aXbX$

$X \rightarrow aY \mid bX \mid \epsilon$

$Y \rightarrow X \mid c$

After eliminating the null productions

$S \rightarrow aXbX \mid abX \mid aXb \mid ab$

$X \rightarrow aY \mid bX \mid b$

$Y \rightarrow X \mid c$

After eliminating the unit productions.

$S \rightarrow aXbX \mid abX \mid aXb \mid ab$

$X \rightarrow aY \mid bX \mid b \quad Y \rightarrow aY \mid bX \mid b \mid c$

The grammar in CNF

$S \rightarrow T_1 T_4 \mid T_2 X \mid T_3 T_b \mid T_a T_b$

$X \rightarrow T_a Y \mid T_b X \mid b$

$Y \rightarrow T_a X \mid T_b X \mid b \mid c$

$T_a \rightarrow a$

$T_b \rightarrow b$

$T_1 \rightarrow T_a X$

$T_2 \rightarrow T_a T_b$

$T_3 \rightarrow T_a X$

$T_4 \rightarrow T_b X$

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CONTINUOUS ASSESSMENT TEST - II

SLOT: B1 + TB1

WINTER SEMESTER 2023-2024

3.(a)

Solution Assume that L is context-free and let n be the constant for L prescribed by the stronger version of the pumping lemma. Consider $\alpha := a^n ca^n ca^n \in L$. The pumping lemma gives us the decomposition $\alpha = \beta_1\beta_2\beta_3\beta_4\beta_5$ with $|\beta_2\beta_4| \geqslant 1$ and $|\beta_2\beta_3\beta_4| \leqslant n$. Since $\alpha' := \beta_1\beta_3\beta_5 \in L$, $\beta_2\beta_4$ must not contain the symbol c, i.e., $\beta_2\beta_4$ consists only of a's. The condition $|\beta_2\beta_3\beta_4| \leqslant n$ implies that $\beta_2\beta_4$ can not stretch over all the three runs of a's in α. Therefore, α' lacks the defining property of the strings of L. This contradiction shows that L is not context-free. ∎

3.(b)

3 b

Define

$L_1 = \{\beta \# \beta^r \# \beta \mid \alpha, \beta \in \{a,b\}^*\}$,

$L_2 = \{\alpha \# \beta^r \# \beta \mid \alpha, \beta \in \{a,b\}^*\}$.

Clearly $L = L_1 \cap L_2$. A CFG can be designed which generates L_1.

$G_1 = (\Sigma, \{S, U, V\}, S, R)$

$S \to UV$

$U \to \# \mid aUa \mid bUb$

$V \to \# \mid Va \mid Vb$.

A similar grammar can be given for L_2.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CONTINUOUS ASSESSMENT TEST - II

WINTER SEMESTER 2023-2024

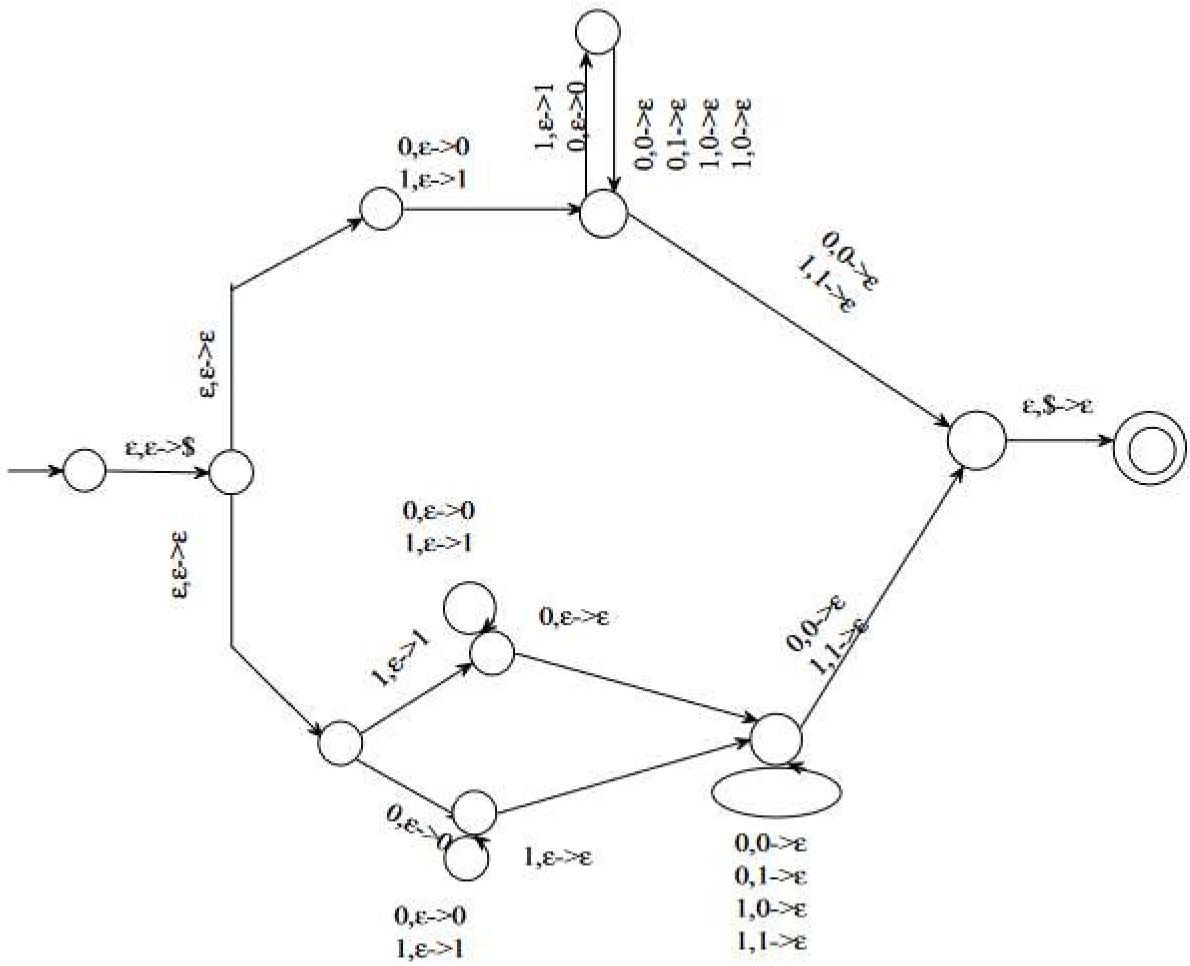
SLOT: B1 + TB1

---

4. Explanation should be given for the CYK algorithm with the necessary steps

A,B,C				
A,B,C	A,B,C			
B,C	A,B	A,B,C		
B,C	ϕ	A,B	A,C	
A,C	B	B	A,C	A,C
a	**b**	**b**	**a**	**a**

5.



The PDA recognizes the language by guessing the length of the input at the beginning. For the guess that the length is even it follows the upper branch, while for the guess that length is odd it follows the lower branch.

1) In the upper branch, the PDA accepts even length strings by pushing and popping alternatively, starting with the second symbol read and ending before the last symbol.
2) In the lower branch, the PDA first pushes the symbols onto the stack and guesses the middle symbol. If all the symbols except the first one and the start symbol $ can be popped out when the PDA finishes reading the symbol except the last one, then the guess is right. The lower branch remembers the first symbol by developing two branches, Each branch accepts when the guessed middle symbol is different from the first one.
3) Before entering the accept state, the PDA only pops out the first read symbol out of the stack when the last one read is the same. Thus it only accepts the string starting and ending with the same symbol.