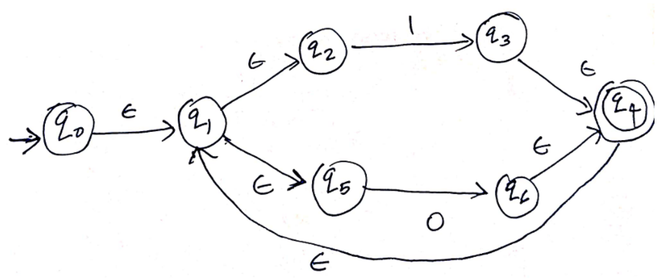
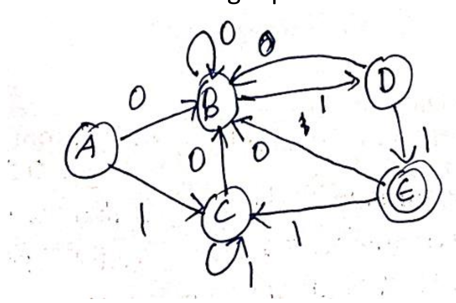
USN

Internal Assessment Test 1 – November 2024

Sub:	**Theory of Computation**				Sub Code:	BCS503	Branch:	AIML/CSE AIML
Date:	**8/11/2024**	Duration:	90 min's	Max Marks: 50	Sem/Sec:	V /A,B CSEAIML(A)		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.(a)	Define the following with examples: i) Language iii) Power of alphabet.	2	CO1	L1
1.(b)	Design a DFA for L= {w | w∈{a,b}*; w do not contains the substring abb}. Write the definition. Show computation for w = bab and w=abb and state whether it is an accepting or rejecting configuration.	8	CO1	L2
2.(a)	Define extended transition function of DFA	4	CO1	L1
2.(b)	Convert the ϵ-NFA to DFA 	6	CO1	L3
3(a)	Write regular expression for the following language: (i) All strings containing exactly 2 a's over Σ = {a,b}. (ii) {w∈{0,1}*: w has third character from the right end is 0.} (iii) {w∈{a,b}*: w has begins or ends with either aa or bb}	3	CO2	L3
3.(b)	Consider the DFA given below and compute, I. Distinguishable and equivalent states II. Minimize the DFA using equivalence method 	7	CO1	L3
4.(a)	State and prove that pumping lemma for Regular language. Prove that the language L={a^n |n is a prime number}not regular	8	CO2	L3
4.(b)	Construct the FSM for the following regular expression $(0+1)^*01$	2	CO2	L3
5	Prove that the regular language is closed under intersection and complement.	10	CO2	L3
6	What is ambiguous grammar. Shows that the following grammar is ambiguous. E→E+E | E*E | (E) | id	10	CO3	L3

Faculty Signature CCI Signature HOD Signature

USN

Internal Assessment Test 1 – November 2024

Sub:	THEORY OF COMPUTATION				Sub Code:	BCS503	Branch:	AIML/ CSE AIML
Date:	8.11.2024	Duration:	90 mins	Max Marks: 50	Sem/Sec:	V/ A,BC		OBE

Answer any FIVE FULL Questions | MARKS | CO | RBT

1 (a) Define the following with example [2] CO1 L1

i) Language

A language is a set of strings from some alphabet (finite or infinite). In other words, any subset L of E* is a language

ii) Power of alphabet

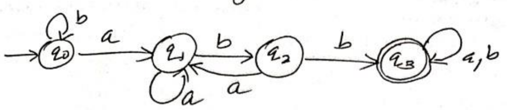
If Σ is an alphabet, the set of all strings can be expressed as a certain length from that alphabet by using exponential notation. The power of an alphabet is denoted by Σk and is the set of strings of length k.

For example,

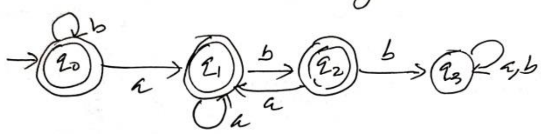
- $\Sigma = \{0,1\}$
- $\Sigma^1 = \{0,1\}$ ($2^1=2$)
- $\Sigma^2 = \{00,01,10,11\}$ ($2^2=4$)
- $\Sigma^3 = \{000,001,010,011,100,101,110,111\}$ ($2^3= 8$)

(b) Design a DFA for L= {w | w∈{a,b}*; w do not contains the substring **abb**}. Write the definition. Show computation for **w = bab** and **w=abb** and state whether it is an accepting or rejecting configuration. [8] CO1 L2

w contains substring abb



doesn't contain substring abb



$\hat{\delta}(q_0, bab) = \delta(\hat{\delta}(q_0, ba), b)$

$= \delta(\hat{\delta}(\delta(q_0,b),a)b)$

$= \delta(\delta(q_0,a)b)$

$= \delta(q_1,b)$

$= q_2$ is accepting state

∴ bab is accepted

$\hat{\delta}(q_0, abb) = \delta(\hat{\delta}(q_0,ab)b)$

$= \delta(\delta(\delta(q_0,a)b)b)$

$= \delta(\delta(q_1,b)b)$

$= \delta(q_2,b)$

$= q_3$ is non accepting state

∴ abb is not accepted

Q.No.	Question	Marks	CO	Level
2 (a)	Define extended transition function of DFA.	[4]	CO1	L1

The extended transition function can be recursively defined as follows:

- **Base case** (empty string): If the input string w is empty (i.e., $w = \epsilon$), the machine stays in the same state.

$$\delta^*(q, \epsilon) = q \quad \text{for all } q \in Q$$

- **Recursive case** (non-empty string): If $w = a_1 a_2 \ldots a_n$, where $a_1, a_2, \ldots, a_n \in \Sigma$, the extended transition function is defined as:

$$\delta^*(q, a_1 a_2 \ldots a_n) = \delta(\delta^*(q, a_1 a_2 \ldots a_{n-1}), a_n)$$

  This means that the machine first processes the string $a_1 a_2 \ldots a_{n-1}$ and reaches some state, and then it processes the last symbol a_n from that state.

The **extended transition function** is a generalization of the transition function that applies to an entire string of symbols, not just a single symbol. It describes the state of the automaton after processing a string of symbols, starting from a given initial state.

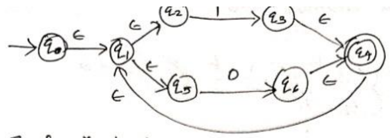
Formally, the extended transition function is often denoted as:

$$\delta^*(q, w) : Q \times \Sigma^* \to Q$$

Where:

- q is the current state.
- w is the input string (where $w \in \Sigma^*$, the set of all strings over the alphabet Σ).
- $\delta^*(q, w)$ gives the state that the automaton ends in after processing the entire string w starting from state q.

Q.No.	Question	Marks	CO	Level
(b)	Convert the ϵ-NFA to DFA	[6]	CO1	L3



1) Find ϵ^* of starting state

$\epsilon^*(q_0) = \{q_0, q_1, q_2, q_5\}$

Assume $\epsilon^*(q_0)$ as A

$\delta(A, 0) = \epsilon^*(\delta(\epsilon^*(A, 0)))$

$= \epsilon^*(\delta(q_0 q_1, q_2 q_5))$

$= \epsilon^*(q_6)$

$= \{q_6, q_4, q_1, q_2, q_5\}$ — B

$\delta(A, 1) = \epsilon^*(\delta(\epsilon^*(A, 1)))$

$= \epsilon^*(\delta((q_0, q_1, q_2 q_5), 1))$

$= \epsilon^*(q_3)$

$= \{q_3\ q_4, q_1 q_2, q_5\}$ — C

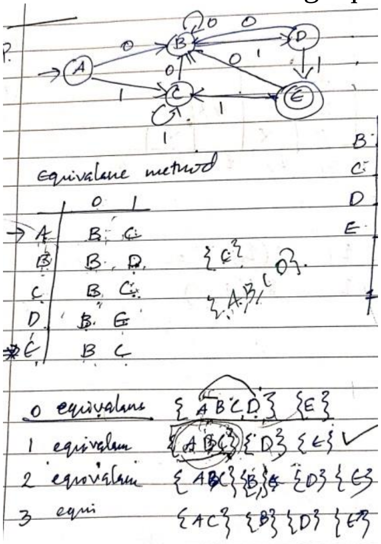
$(B, 0) = \epsilon^*(\delta(\epsilon^*(B, 0)))$

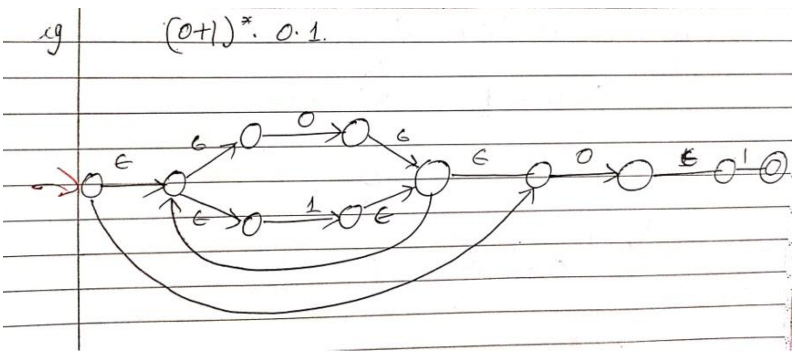
$= B$

$\delta(B, 1) = C$

$\delta(C, 0) = B$

$\delta(C, 1) = C$

3 (a)	Write regular expression for the following language: (i) All strings containing exactly 2 a's over $\Sigma = \{a,b\}$. $(b)^*a(b)^*a(b)^*$ (ii) $\{w \in \{0,1\}^*$: w has third character from the right end is 0.$\}$ $(0+1)^*0(0+1)(0+1)$ (iii) $\{w \in \{a,b\}^*$: w has begins or ends with either aa or bb$\}$ $(aa+bb)(a+b)^*+(a+b)^*(aa+bb)$	[3]	CO2	L3
(b)	Consider the DFA given below and compute, I. Distinguishable and equivalent states If X and Y are two states in a DFA, we can combine these two states into {X, Y} if they are not distinguishable. Two states are distinguishable, if there is at least one string S, such that one of δ (X, S) and δ (Y, S) is accepting and another is not accepting. Hence, a DFA is minimal if and only if all the states are distinguishable. II. Minimize the DFA using equivalence method 	[7]	CO1	L3
4 (a)	State and prove that pumping lemma for Regular language. Prove that the language $L=\{a^n \mid n$ is a prime number$\}$not regular. **Statement of the Pumping Lemma:** Let LLL be a regular language. Then, there exists a constant ppp (called the **pumping length**) such that for any string w∈Lw \in Lw∈L with |w|≥p|w| \geq p|w|≥p, www can be decomposed into three parts w=xyzw = xyzw=xyz, such that the following conditions hold: 1. **Length condition**: |xy|≤p|xy| \leq p|xy|≤p. 2. **Non-empty condition**: |y|≥1|y| \geq 1|y|≥1. 3. **Pumping condition**: For all i≥0i \geq 0i≥0, the string xyiz∈Lxy^i z \in Lxyiz∈L. That is, the string w=xyzw = xyzw=xyz can be "pumped" by repeating the middle part yyy any number of times, and the resulting string will still belong to the	[8]	CO2	L3

	language LLL P.T $L = \{a^p \mid p \text{ is a prime}\}$ is not regular. $L = \{aa, aaa, aaaaa.$ $n = 3$ $w = aaa$ $\lvert w\rvert \geq n$ $w = xyz$ $x = a$ $3 \geq 3$ $y = a$ $z = a$ $\lvert xy\rvert \leq n$ $k=0$ $aa \in L$ $2 \leq 3$ $k=1$ $aaa \in L$ $k=2$ $aaaa \notin L$			
(b)	Construct the FSM for the following regular expression. (0+1)*01 	[2]	CO2	L3
5	Prove that the regular language is closed under intersection and complement. **1. Closure under Intersection:**	[10]	CO2	L3

Proof:

Since regular languages are recognized by deterministic finite automata (DFAs), let us assume that L1L_1L1 and L2L_2L2 are recognized by DFAs.

- Let A1=(Q1,Σ,δ1,q1,0,F1)A_1 = (Q_1, \Sigma, \delta_1, q_{1,0}, F_1)A1 =(Q1,Σ,δ1,q1,0,F1) be a DFA for L1L_1L1, where:
  - Q1Q_1Q1 is the set of states,
  - Σ\SigmaΣ is the input alphabet,
  - δ1\delta_1δ1 is the transition function,
  - q1,0q_{1,0}q1,0 is the start state,
  - F1F_1F1 is the set of accepting states.
- Let A2=(Q2,Σ,δ2,q2,0,F2)A_2 = (Q_2, \Sigma, \delta_2, q_{2,0}, F_2)A2 =(Q2,Σ,δ2,q2,0,F2) be a DFA for L2L_2L2, where the components are similar for A2A_2A2.

To recognize the intersection L1∩L2L_1 \cap L_2L1∩L2, we construct a new DFA A=(Q,Σ,δ,q0,F)A = (Q, \Sigma, \delta, q_0, F)A=(Q,Σ,δ,q0,F) as follows:

1. **States**: The set of states QQQ is the Cartesian product of the state sets of A1A_1A1 and A2A_2A2:

   Q=Q1×Q2Q = Q_1 \times Q_2Q=Q1×Q2

   Each state in QQQ is a pair (q1,q2)(q_1, q_2)(q1,q2), where q1q_1q1 is a state from A1A_1A1 and q2q_2q2 is a state from A2A_2A2.

2. **Start State**: The start state q0q_0q0 is the pair (q1,0,q2,0)(q_{1,0}, q_{2,0})(q1,0,q2,0), where q1,0q_{1,0}q1,0 is the start state of A1A_1A1 and q2,0q_{2,0}q2,0 is the start state of A2A_2A2.
3. **Transition Function**: The transition function δ\deltaδ is defined as:

   δ((q1,q2),a)=(δ1(q1,a),δ2(q2,a))\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))δ((q1,q2),a)=(δ1(q1,a),δ2(q2,a))

   for each a∈Σa \in \Sigmaa∈Σ, where δ1\delta_1δ1 and δ2\delta_2δ2 are the transition functions of A1A_1A1 and A2A_2A2, respectively.

4. **Accepting States**: The set of accepting states FFF consists of all pairs (q1,q2)(q_1, q_2)(q1,q2) such that q1∈F1q_1 \in F_1q1∈F1 and q2∈F2q_2 \in F_2q2∈F2, i.e., both automata are in an accepting state:

   F=F1×F2F = F_1 \times F_2F=F1×F2

This construction ensures that AAA accepts a string if and only if both A1A_1A1 and A2A_2A2 accept the string. Thus, AAA recognizes L1∩L2L_1 \cap L_2L1 ∩L2.

Since AAA is a DFA, L1∩L2L_1 \cap L_2L1∩L2 is a regular language. Therefore, regular languages are closed under intersection.

2. Closure under Complement:

Proof:

Let LLL be a regular language. By definition, a regular language is recognized by a DFA. Suppose that LLL is recognized by a DFA A=(Q,Σ,δ,q0,F)A = (Q, \Sigma, \delta, q_0, F)A=(Q,Σ,δ,q0,F), where:

- QQQ is the set of states,
- Σ\SigmaΣ is the input alphabet,
- δ\deltaδ is the transition function,
- q0q_0q0 is the start state,
- FFF is the set of accepting states.

To recognize the complement L¯\overline{L}L, we construct a new DFA A′=(Q,Σ,δ,q0,F′)A' = (Q, \Sigma, \delta, q_0, F')A′=(Q,Σ,δ,q0,F′), where F′=Q∖FF' = Q \setminus FF′=Q∖F is the set of non-accepting states (i.e., the complement of FFF).

- **States**: The set of states remains the same, i.e., QQQ.
- **Start State**: The start state remains the same, i.e., q0q_0q0.
- **Transition Function**: The transition function remains the same, i.e., δ′=δ\delta' = \deltaδ′=δ.
- **Accepting States**: The accepting states in A′A'A′ are the states in QQQ that are **not** in FFF, i.e., F′=Q∖FF' = Q \setminus FF′=Q∖F.

Since the only change in the construction is that we swap the accepting and non-accepting states, A′A'A′ is also a DFA. Therefore, A′A'A′ recognizes L¯\overline{L}L, and thus L¯\overline{L}L is regular.

Hence, regular languages are closed under complement.

Conclusion:

Since we have shown that regular languages are closed under both intersection and complement, the class of regular languages is closed under both operations.

6	What is ambiguous grammar? Shows that the following grammar is ambiguous. E→E+E | E*E | (E) | id	[10]	CO3	L3

A **context-free grammar (CFG)** is said to be **ambiguous** if there exists at least one string in the language generated by the grammar that can be derived in more than one way, i.e., the string has **two or more distinct leftmost derivations or two or more distinct rightmost derivations** or **two or more distinct parse trees**.

Parse Tree Representation

Now that we have two different derivations, we can show that these correspond to different parse trees. The two parse trees for the string id+id∗id\text{id} + \text{id} * \text{id}id+id∗id are as follows:

```
      E
     /   \
   E       E
  /         \
 id          E * E
            /   \
          id    id

      E
     /   \
   E * E    E
  /   \     /  \
 id     id  id   id
```

the parse trees have different structures, corresponding to different interpretations of the string. In the first tree, the addition happens before multiplication, and in the second tree, the multiplication happens before addition.

CI CCI HOD

USN | | | | | | | | | | |

CMRIT — CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – November 2024

Sub:	THEORY OF COMPUTATION			Sub Code:	BCS503	Branch:	AIML/ CSE AIML
Date:	8.11.2024	Duration:	90 mins	Max Marks: 50	Sem/Sec:	V/ A,BC	OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1 (a)	Define the following with example i) Language ii) Power of alphabet **Definition and Example-------------------------(0.5+0.5)*3 M**	[2]	CO1	L1
(b)	Design a DFA for L= {w | w∈{a,b}*; w do not contains the substring **abb**}. Write the definition. Show computation for **w = bab** and **w=abb** and state whether it is an accepting or rejecting configuration. **Design-------------------4M** **Definition--------------1M** **Extended transition function for both input string-----------------(1.5*2)=3M**	[8]	CO1	L2
2 (a)	Define extended transition function of DFA. **Definition------------3M** **Eg-----------1M**	[4]	CO1	L1
(b)	Convert the ϵ-NFA to DFA **Design-------------------4.5M** **Diagram--------------1.5M**	[6]	CO1	L3
3 (a)	Write regular expression for the following language: (i) All strings containing exactly 2 a's over Σ = {a,b}. $(b)^*a(b)^*a(b)^*$ (ii) {w∈{0,1}*: w has third character from the right end is 0.} $(0+1)^*0(0+1)*(0+1)*$ (iii) {w∈{a,b}*: w has begins or ends with either aa or bb} (aa+bb)(a+b)*+(a+b)*(aa+bb) **1 mark each**	[3]	CO2	L3
(b)	Consider the DFA given below and compute, I. Distinguishable and equivalent states **1.5 marks each** II. Minimize the DFA using equivalence method **4 marks**	[7]	CO1	L3
4 (a)	State and prove that pumping lemma for Regular language. Prove that the language L={a^n |n is a prime number}not regular. State and prove that pumping lemma- **4 marks** L={a^n |n is a prime number}not regular - **4 marks**	[8]	CO2	L3
(b)	Construct the FSM for the following regular expression. **Diagram-2 marks**	[2]	CO2	L3
5	Prove that the regular language is closed under intersection and complement. closed under intersection-5 marks closed under complement- 5 marks	[10]	CO2	L3

6	What is ambiguous grammar? Shows that the following grammar is ambiguous. $E \rightarrow E+E \mid E*E \mid (E) \mid id$ Definition – 3 marks Solution- 7 marks	[10]	CO3	L3

CI CCI HOD