
USN

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

Internal Assessment Test 1 – November 2025

Sub:	Theory of Computation				Sub Code:	BCS503	Branch:	AIML/CSE AIML
Date:	29/09/2025	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) Alphabet ii) Language iii) Power of alphabet.	3	CO1	L1

i) Alphabet

Definition:
An **alphabet** is a finite set of symbols or characters from which strings (words) can be formed in a formal language or system.

Example:

- The English alphabet: {a,b,c,…,z}\{a, b, c, \ldots, z\}{a,b,c,…,z}
- A binary alphabet: {0,1}\{0, 1\}{0,1}

ii) Language

Definition:
A **language** over an alphabet is a set of strings (finite sequences) composed of symbols from that alphabet. A language can be finite or infinite.

Example:

- Over the alphabet {a,b}-a language could be the set of all strings with even length: {ϵ,ab,ba,aabb,bbaa,…},{ϵ,ab,ba,aabb,bbaa,…}

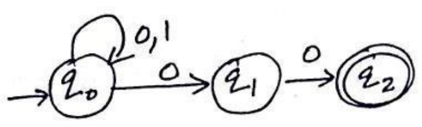
iii) Power of Alphabet

Definition:
The **power of an alphabet** (also called the cardinality of the alphabet) is the number of distinct symbols in the alphabet.

Example:

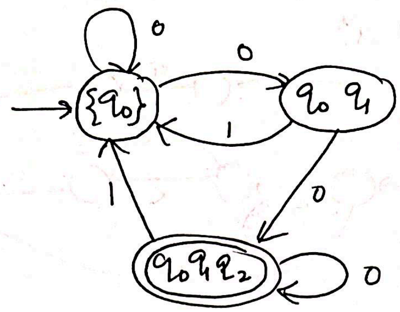
- The English alphabet has a power of 26 because it contains 26 letters.
- The binary alphabet {0,1}\{0,1\}{0,1} has a power of 2.

1.(b)	Design a DFA from the given NFA using a subset construction method.	7	CO1	L3

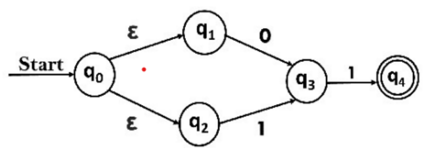


Sol. 1(b)

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$



2	Design DFA for the following epsilon NFA	10	CO1	L3



Sol. 2 $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1\}$

$\epsilon\text{-closure}(q_2) = \{q_2\}$

$\epsilon\text{-closure}(q_3) = \{q_3\}$

$\epsilon\text{-closure}(q_4) = \{q_4\}$

Start state of DFA = $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$ – A.

$\delta(A, 0) = \epsilon\text{-closure}(\delta(A, 0))$

$= \epsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon\text{-closure}(q_3)$

$= \{q_3\}$ – B

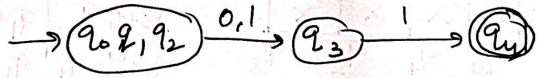
$\delta(A, 1) = \epsilon\text{-closure}(\delta(A, 1))$

$= \epsilon\text{-closure}(\{q_3\})$

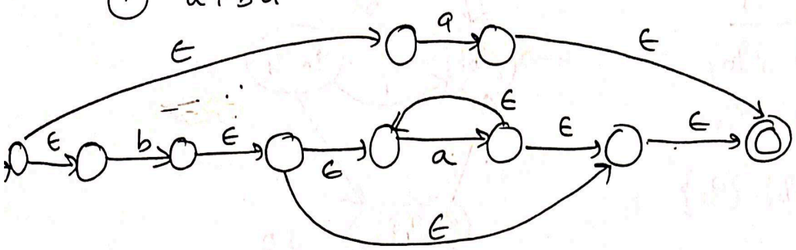
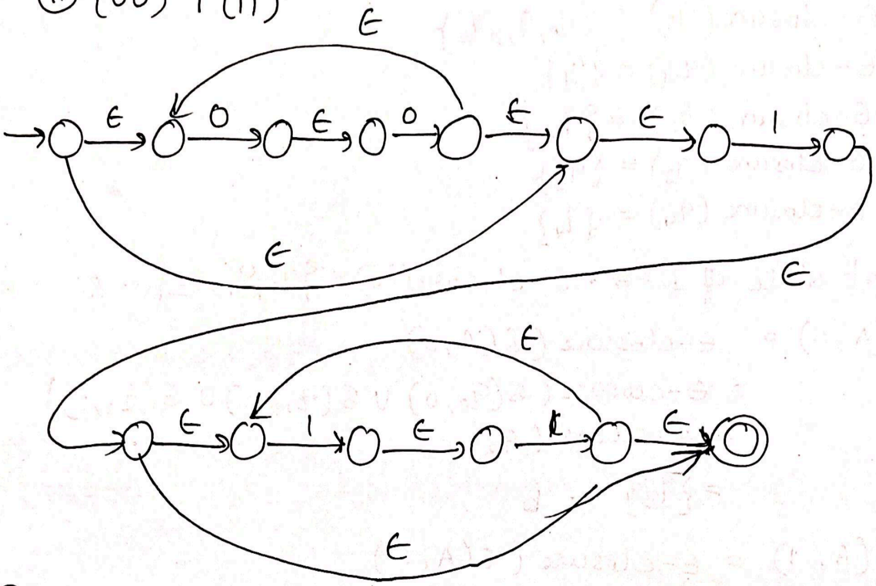
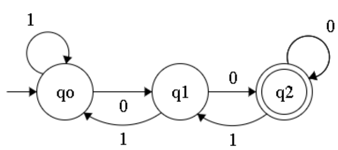
$= \{q_3\}$.

$\delta(B, 0) = \phi \qquad \delta(B, 1) = \{q_4\}$

	0	1
$\rightarrow \{q_0, q_1, q_2\}$	q_3	q_3
q_3	ϕ	q_4
q_4	ϕ	ϕ



<table>
<tr><td>3(a)</td><td>Construct epsilon NFA for the given Regular expression
(i) a+ba* (iii) (00)* 1 (11)*

3(a) ε-NFA.
(i) a+ba*

(ii) (00)* 1 (11)*
</td><td>3</td><td>CO2</td><td>L2</td></tr>
<tr><td>3.(b)</td><td>Find the regular expression for the following DFA using state elimination method.
</td><td>7</td><td>CO2</td><td>L2</td></tr>
</table>

<table>
<tr><td>4.(a)</td><td>State and prove that pumping lemma for Regular language. Prove that the language L= {$0^n 1^n$ /n>=1}is not regular

Statement:

If a language L is regular, then there exists a constant p (called the pumping length) such that every string $s \in L$ with $|s| \geq p$ can be divided into three parts,
$s = xyz$
such that:

1. $y \neq \varepsilon$ (i.e., y is not empty)
2. $|xy| \leq p$
3. $\forall i \geq 0,\ xy^iz \in L$
(i.e., the string remains in the language when we pump y zero or more times)</td><td></td><td>CO2</td><td>L1</td></tr>
</table>

Proof that the language L= {$0^n 1^n$ /n>=1}is not regular
Proof by contradiction :
Let's assume that L is regular.

1. Given n>=1.
2. Let choose w = $0^n 1^n$ such that |w|= 2n>=n.
3. Using pumping lemma, given partition
   w= xyz
   Such that |xy| <= n and
   |y| > 0
   Consider the string

   $\underbrace{000000\ldots\ldots0}_{Xy}\ \underbrace{1111\ldots.1}_{z}$

   Where x= 0^m and y=0^l
   The string x and y consists of only 0's. |xy|<= n
4. Let select k=0
   xy^0z = xz
   = 0^(n-l) 1^n = 0^m 1^n
   Where m = n - |y|
   = n-l != n.

Therefore, xy^0z not belongs to L
=> L is not regular.

Q. No.	Question	Marks	CO	Level
4.(b)	Define Regular expression. Write the regular expression for the following languages:	5	CO2	L3

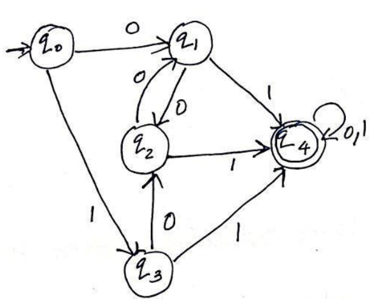
a. Strings of a's and b's starting with a and ending with b.
b. L = { $a^n b^m$ \| (n + m) is even} }.
Solution:

A **regular expression (RE)** is a symbolic notation used to describe a **regular language**. It defines a set of strings over a given **alphabet** using operators like:

- **Concatenation**: ababab means "a followed by b"
- **Union (or)**: a+ba + ba+b or a∣ba \mid ba∣b means "a or b"
- **Kleene star**: a*a^*a* means "zero or more occurrences of a"
- **Parentheses** for grouping: (a+b)*(a + b)^*(a+b)*

a. a(a+b)*b
b. (aa)*(bb)* + a(aa)*b(bb)*

Q. No.	Question	Marks	CO	Level
5	Minimize the following DFA using table filling method.	10	CO1	L2



5 (c)

q_1	2			
q_2	2			
q_3	2			
$*q_4$	×	×	×	×
	q_0	q_1	q_2	q_3

Pass II
On state q_0

	0	1
$q_0 q_3$	$q_1 q_2$	$q_3 q_4$
$q_0 q_2$	$q_1 q_1$	$q_3 q_4$
$q_0 q_1$	$q_1 q_2$	$q_3 q_4$

on state q_1

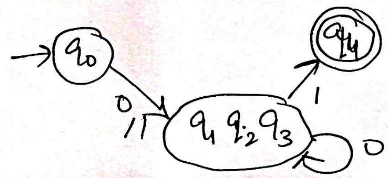
	0	1
$q_1 q_3$	$q_2 q_2$	$q_4 q_4$
$q_1 q_2$	$q_2 q_1$	$q_4 q_4$

One state q_2

	0	1
$q_2 q_3$	$q_1 q_2$	$q_4 q_4$

States $\to \{q_0, \{q_1, q_2, q_3\}, q_4\}$

DFA

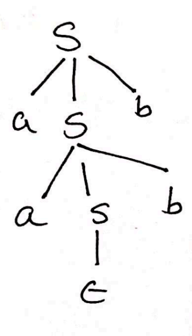


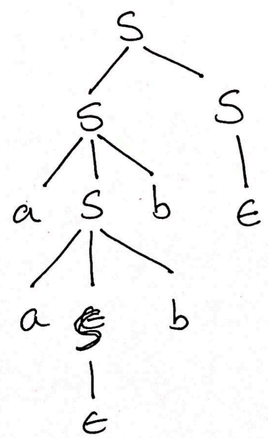
6	Define CFG with tuple components? Derive the derivation tree for the following CFG S □ aSb S □ € S □ SS	10	CO3	L2

6. Context free Grammars (CFG) are defined by a set of four components: $G = (V, T, P, S)$
Nonterminal symbols (V)
Terminal symbols (T)
Production Rules $A \to w$
Start Symbol (S)

$S \to aSb \mid \epsilon$
$S \to SS$

$w = aabb$





	ii) Define Ambiguous grammar with one example?			
	ii) ~~Ab~~. Ambiguous grammar: refers to a set of productio rules that can produce more than one parse tree for the same i/p string $S \to aSb \mid \epsilon$ $S \to SS$			

Faculty Signature CCI Signature HOD Signature