



Fifth Semester B.E./B.Tech. Degree Examination, June / July 2026 Automata Theory and Compiler Design

Time: 3 hrs.

Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

- 1 a. Define the following terms :
 - (i) Alphabet
 - (ii) String
 - (iii) Length of String
 - (iv) Power of an alphabet
 - (v) Language

(05 Marks)
- b. Design DFA for the following languages over $\Sigma = \{a, b\}$
 - i) $L = \{w : |w| \bmod 3 > |w| \bmod 2\}$
 - ii) The set of all strings containing the substring 'ab' at the end.

(08 Marks)
- c. Convert the following NFA to its equivalent DFA.

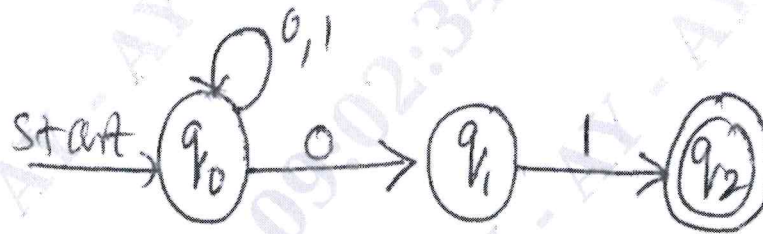


Fig. Q. 1 (c)

(07 Marks)

OR

- 2 a. Construct a minimized DFA for the following:

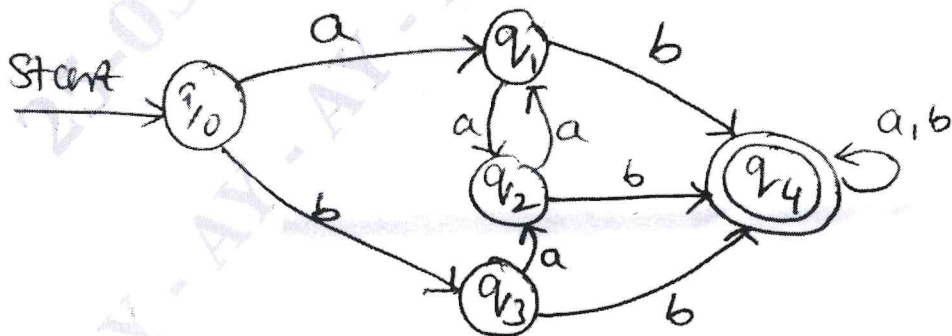


Fig. Q. 2 (a)

(08 Marks)

- b. With the help of a diagram, explain various phases of compiler. Also show the steps in translation of an assignment statement.
Position = initial + rate * 60

(12 Marks)

Module-2

- 3 a. Define Regular expression. Construct the regular expressions for the following languages:

- (i) Strings of a's and b's whose 3rd symbol from right is a
(ii) $L = \{ a^{2n} b^{2m+1} : m \geq 0, n \geq 0 \}$

(06 Marks)

- b. Construct an equivalent regular expression for the following DFA using state elimination method.

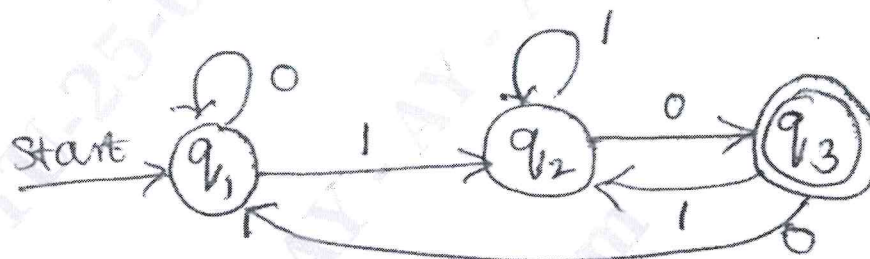


Fig. Q. 3 (b)

(08 Marks)

- c. Convert the regular expression $(0 + 1)^* 1 (0 + 1)$ to a ϵ -NFA.

(06 Marks)

OR

- 4 a. With a neat diagram explain input buffering schemes of lexical analyses.

(10 Marks)

- b. Construct the transition diagrams to recognize the following tokens.
i) relopertors ii) id and keywords iii) Unsigned numbers

(10 Marks)

Module-3

- 5 a. Define Context free Grammar. Obtain the CFG for the following languages:

i) $L = \{ a^m b^n : m \geq 1, n \geq 1 \}$

ii) $L = \{ 0^n 1^n : n \geq 0 \}$

(06 Marks)

- b. Consider the following CFG with productions.

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid \text{id},$$

Write the LMD, RMD and Parse tree for the string $\text{id} * (\text{id} + \text{id})$.

(09 Marks)

- c. Consider the CFG given in Q. 5 (b), eliminate the left recursion.

(05 Marks)

OR

- 6 a. With a net diagram explain the rule of a Parser. (05 Marks)
- b. Consider the grammar,
 $S \rightarrow iEtSS' \mid a$
 $S' \rightarrow eS \mid \epsilon$
 $E \rightarrow b$
 Construct Predictive Parsing table. (10 Marks)
- c. Briefly explain error recovery strategies of Predictive Parser. (05 Marks)

Module-4

- 7 a. Define PDA. Explain the working of PDA with a neat diagram. (05 Marks)
- b. Design a PDA to accept the language
 $L = \{ WW^R \mid W \in \{a, b\}^* \}$ by a final state. (05 Marks)
- c. What is Shift- Reduce Parsing? List the possible actions. Show the configurations of a Shift – Reduce Parser on the input $id_1 * id_2$. (10 Marks)

OR

- 8 a. Construct the SLR Parsing table for the grammar given in Q. 5 (b). (12 Marks)
- b. Construct LR(1) Automation for the grammar
 $S \rightarrow CC$, $C \rightarrow cC \mid d$ (08 Marks)

Module-5

- 9 a. With a neat diagram, explain the working of basic Turing machine. (05 Marks)
- b. Design a Turing machine to accept the language
 $L = \{ 0^n 1^n \mid n \geq 1 \}$ (10 Marks)
- c. Write a short note on multitape Turing machine. (05 Marks)

OR

- 10 a. Give SDD for a simple desk calculator. (05 Marks)
- b. Construct DAG for the expression
 $a + a * (b - c) + (b - c) * d$ (05 Marks)
- c. List and briefly explain the issues in the design of a code generator. (10 Marks)
