



USN

--	--	--	--	--	--	--	--	--	--

21CS51

Fifth Semester B.E. Degree Examination, June/July 2024
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 terminologies with appropriate examples and notations:
 i) Kleene star ii) Alphabet iii) Language iv) Power of an alphabet (08 Marks)
 b. Design a DFA to accept the language
 $L = \{ w \mid w \text{ is of even length and begins with } 01 \}$ (07 Marks)
 c. Explain briefly phases of a compiler. (05 Marks)

OR

- 2 a. Define with terminologies different ways of representing Automata considering an example. (08 Marks)
 b. Consider the following ϵ -NFA Fig.Q2(b).

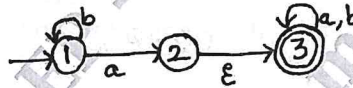


Fig.Q2(b)

- i) Compute the ϵ -Closure of each state ii) Convert the automation to a DFA (07 Marks)
 c. Explain in brief commonly used compiler construction tools. (05 Marks)

Module-2

- 3 a. Write regular expression (RE) for the following Languages.
 i) The set of all strings such that the number of 0's is ODD. $\Sigma = \{0, 1\}$
 ii) Every ODD length string begins with 11. $\Sigma = \{0, 1\}$ (08 Marks)
 b. Convert the following FSM into RE using state elimination technique. Refer Table Q3(b).

δ	0	1
$\rightarrow q_1$	q_2	q_1
q_2	q_2	q_4
$*q_3$	q_4	q_2
$*q_4$	q_4	q_1

Table Q3(b)

- c. Describe the languages denoted by the following regular expressions :
 i) $a.(a+b)^*.b$ ii) $(a+b)^*.a.(a+b)(a+b)$ (05 Marks)

OR

- 4 a. Write Regular Expressions for the following languages:
 i) All strings of lowercase letters that contain the five vowels in order.
 ii) All the strings of a's and b's that contain the substring abb. (08 Marks)
 b. Convert the following DFA in Fig.Q4(b) to a Regular Expression using Kleene's theorem.

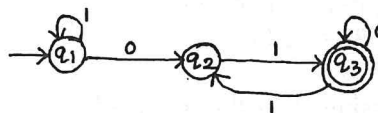


Fig.Q4(b)

- c. Explain with neat diagram interactions between the lexical analyzer and the parser. (05 Marks)

Module-3

- 5 a. Design Context-Free Grammars (CFG) for the following languages:
- $L = \{a^i b^j c^k \mid i = j = k\}$
 - The set of all strings of 0's and 1's where the number of 0's is equal to the number of 1's. (08 Marks)
- b. Given the Context-Free Grammar below:
- $$S \rightarrow AS \mid \epsilon$$
- $$A \rightarrow aa \mid ab \mid ba \mid bb$$
- Give leftmost and rightmost derivations and parse tree for the following strings:
- aaba
 - baabab
 - aaabbb
- (06 Marks)
- c. Construct the top-down parse tree for string $w = id + id * id$ by using grammar given below:
- $$E \rightarrow TE'$$
- $$E' \rightarrow +TE' \mid \epsilon$$
- $$T \rightarrow FT'$$
- $$T' \rightarrow *FT' \mid \epsilon$$
- $$F \rightarrow (E) \mid id$$
- (06 Marks)

OR

- 6 a. Remove ambiguity from the Grammar given below:
- $$S \rightarrow aSb$$
- $$S \rightarrow aaSb$$
- $$S \rightarrow \epsilon$$
- (08 Marks)
- b. Consider the Context – Free Grammar given below:
- $$S \rightarrow aB \mid bA$$
- $$A \rightarrow a \mid aS \mid bAA$$
- $$B \rightarrow b \mid bS \mid aBB$$
- The string $w = 'aaabbabbba'$ and find
- Left-most derivation
 - Right-most derivation
 - Parse-tree
- (06 Marks)
- c. Explain the role of Parser in the compiler model. (06 Marks)

Module-4

- 7 a. Design PDA to accept the language
- $$L = \{WcW^R \mid W \in \{a, b\}^*\}$$
- Write ID for $W = 'bacab'$ (10 Marks)
- b. Construct bottom-up parse tree for the following input strings by considering grammar given below:
- $$E \rightarrow E + T \mid T$$
- $$T \rightarrow T * F \mid F$$
- $$F \rightarrow (E) \mid id$$
- $W_1 = id * id$ $W_2 = id + id * id$ (10 Marks)

OR

- 8 a. Design a ND-PDA to accept the language
- $$L = \{a^m b^n \mid m \neq n, n, m > 0\}$$
- and write ID for $W = aaabb$ (10 Marks)
- b. Explain LR – Parsing algorithm in detail. (10 Marks)

Module-5

- 9 a. Design Turing Machine for the language
 $L = \{ a^i b^i \mid i > 0 \}$
Write ID for string $W = "aabb"$ (10 Marks)
- b. Write a short note on the following topics:
i) Recursive Languages ii) Universal Turing Machines (10 Marks)
- OR**
- 10 a. Construct Direct Acyclic Graph (DAG) and corresponding three address code for the following expressions:
i) $a + a * (b - c) + (b - c) * d$
ii) $((x + y) - ((x + y) * (x - y))) + ((x + y) * (x - y))$ (10 Marks)
- b. Write a short note on the following :
i) Multitape Turing Machine
ii) Non-Deterministic Turing Machine (10 Marks)

* * * * *

