AUTOMATA THEORY AND COMPILER DESIGN
(MCA2141)

Time Allotted : 2½ hrs **Full Marks : 60**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 4 (four) from Group B to E, taking one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A

1. Answer any twelve: **12 × 1 = 12**

Choose the correct alternative for the following

(i) Context sensitive grammar is accepted by
(a) Turing Machine (b) Linearly bounded automaton
(c) Pushdown automaton (d) Finite automaton

(ii) Given an arbitrary non-deterministic finite automaton (NFA) with n states, the maximum number of states in an equivalent minimized DFA is at least
(a) n^2 (b) 2n (c) 2^n (d) n!

(iii) Consider a DFA with states {q0, q1, q2}, where q0 is the initial state, and q2 is the only final state. The alphabet is {0, 1}, and the transition function is defined as follows:
- δ(q0, 0) = q0
- δ(q0, 1) = q1
- δ(q1, 0) = q2
- δ(q1, 1) = q0
- δ(q2, 0) = q1
- δ(q2, 1) = q2

Which of the following strings is ACCEPTED by this DFA?
(a) 011 (b) 101 (c) 110 (d) 100.

(iv) Which of the following grammars is in Chomsky Normal Form (CNF)?
(a) S → AB | a (b) S → ε
(c) S → aB | BC (d) S → a

(v) What is the primary role of a lexical analyzer in a compiler?
(a) To translate source code into machine code
(b) To parse the syntax of the source code
(c) To break down the source code into tokens
(d) To optimize the execution of code.

(vi) Shift-reduce parser uses
(a) pointer (b) stack (c) array (d) queue

(vii) Which derivation is generated by the bottom-up parser?
(a) Right-most derivation in reverse (b) Left-most derivation in reverse
(c) Right-most derivation (d) Left-most derivation.

(viii) Which graph describes the basic block and successor relationship?
(a) Hamiltonian graph (b) DAG
(c) Flow graph (d) Dependency graph.

(ix) Which of the following is a primary purpose of intermediate code generation in a compiler?
(a) To convert source code directly to machine code
(b) To optimize the source code before it is compiled
(c) To provide a machine-independent representation of the source code
(d) To perform lexical analysis of the source code.

(x) Which of the following techniques is commonly used to improve the performance of a loop in a program?
(a) Loop Unrolling (b) Code Refactoring
(c) Exception Handling (d) Algorithmic Complexity Analysis.

Fill in the blanks with the correct word

(xi) A finite automaton consists of a finite set of states, a finite set of input symbols, a transition function, an initial state, and a set of _____ states.

(xii) In a deterministic finite automaton (DFA), for each state and input symbol, there is exactly _____ transition to another state.

(xiii) In lexical analysis, a _____ is a sequence of characters that forms a basic unit of meaning, such as a keyword, identifier, or operator.

(xiv) A common method used in top-down parsing to handle recursive rules and ensure that the parsing process does not enter an infinite loop is _____ elimination.

(xv) In intermediate code generation, the use of _____ enables the compiler to perform optimizations and transformations on the code that are independent of the target architecture.

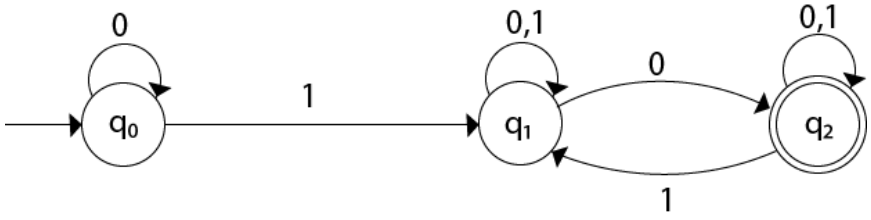
Group - B

2. Write Regular Expressions for the following (any four):
   i. All strings of letters that contain a valid e-mail address.
   ii. Identifiers of C language.
   iii. All strings of letters that contain the five vowels in order.
   iv. All strings of digits that contain a valid IP address.
   v. All strings of binary digits that contain at least three characters in which the second last character is always zero.

vi. Finding a Date in MM/DD/YYYY Format. *[(CO1)(Apply/IOCQ)]*

(3 × 4) = 12

3. (a) Convert the following NFA to DFA and minimize it (if possible).



[(CO1)(Apply/IOCQ)]

(b) Describe a simple FSM that recognizes strings over {a,b} where the number of 'a's is even. Provide a brief diagram or description of the FSM. *[(CO2)(Apply/IOCQ)]*

7 + 5 = 12

Group - C

4. (a) Explain the synthesis phase in the context of Two Phase Compilation. *[(CO2)(Understand/LOCQ)]*

(b) Convert the following CFG to GNF: S -> AB|BC, A -> AB|a, B -> AA|CB|b, C -> a|b. *[(CO3)(Apply/IOCQ)]*

(c) Prove that L = $\{a^i b^i \mid i \geq 0\}$ is not regular using pumping lemma. *[(CO3)(Apply/IOCQ)]*

4 + 4 + 4 = 12

5. (a) Define a context-free grammar (CFG) and explain its components. *[(CO3)(Remember/LOCQ)]*

(b) Given the CFG below, describe the language it generates and determine whether the string "aabb" belongs to the language: S → aSb | ε. *[(CO3)(Apply/IOCQ)]*

(c) Write a LEX program that will mimic the wc command of Unix. *[(CO2)(Apply/IOCQ)]*

4 + 4 + 4 = 12

Group - D

6. (a) Consider the grammar:
i→ i + d | i – d | i * d | d
d→ 0 | 1 | 2 | 3 | … | 9
Show the parse tree for the expression: 5 + 6 * 7 – 2. *[(CO4)(Apply/IOCQ)]*

(b) Consider the following grammar for arithmetic expressions:

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

Determine if the grammar is ambiguous. If it is, provide an example of a string that can be derived in more than one way using this grammar. *[(CO4)(Remember/LOCQ)]*

6 + 6 = 12

7. (a) Create the SLR parsing table for the following grammar.
S → CC, C → cC, C → d *[(CO4)(Apply/IOCQ)]*

(b) Explain Shift-Reduce conflict in SLR parsing with suitable example. *[(CO4)(Remember/LOCQ)]*

8 + 4 = 12

Group - E

8. (a) Explain the concept of three-address code in intermediate code generation. Discuss its advantages in the context of compiler design. *[(CO5)(Understand/LOCQ)]*

   (b) Convert the following expression into three-address code: A = (B + C) ∗ (D − E) / F
   Show each step clearly, indicating how temporary variables are introduced.
   [(CO5)(Apply/IOCQ)]

   6 + 6 = 12

9. (a) Consider the following C code snippet:

```
for (int i = 0; i < n; i++)
{
sum = sum + (a * b);
x = y + z;
}
```

   Identify and explain two loop optimization techniques that can be applied to the given loop. *[(CO6)(Apply/IOCQ)]*

   (b) Apply these optimizations to rewrite the code, showing how the optimizations reduce redundant computations and improve efficiency. *[(CO6)(Apply/IOCQ)]*

   (c) Given the following sequence of three-address code instructions, identify the basic blocks:
   1. a = b + c
   2. if a > 10 goto L1
   3. d = a * 2
   4. L1: e = d + 1.

   [(CO6)(Apply/IOCQ)]

   4 + 4 + 4 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	25	75	0