B5.2-R4: AUTOMATA THEORY AND COMPILER DESIGN

NOTE:

1. **Answer question 1 and any FOUR from questions 2 to 7.**
2. **Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) Explain the role of Parser with an example.

b) Construct a Mealy machine which is equivalent to Moore machine given by the following table:

Present state	Next state		Output
	a=0	a=1	
→ q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0

c) Construct a grammar generating L= $\{wcw^T | w \in \{a, b\}^*\}$.

d) How can you find a Regular Expression determined by a transition system using Arden's theorem. Write down the assumptions you have made.

e) If G is a grammar S produces S → SBS|a. Show that G is ambiguous.

f) What do you mean by ℓ-values and r-values of an identifier? How are they used in translation of expressions?

g) How can you define technically a directed acyclic graph (DAG) for a basic block? Write down the applications of DAG.

(7x4)

2.

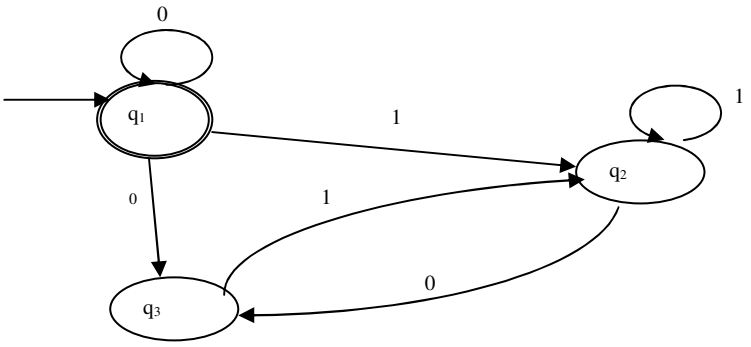
a) Construct a DFA accepting all strings over a, b ending with a b. Minimize the above DFA.

b) Design a Turing machine to recognize all strings consisting of an even number of 1's.

(9+9)

3.

a) Find out the Regular Expression for the language accepted by the following NFA.



b) Construct a grammar generating $\{a^n b^n c^n | n \geq 1\}$.

c) Construct a PDA accepting the set of all strings over {a,b} with equal no. of a's & b's.

(7+4+7)

4.

a) What are the different phases of a compiler? How can you categorize them into front end and back end?

b) When is a grammar said to be an ambiguous? Give an example. Convert this ambiguous grammar into an unambiguous one.

c) What do you mean by syntax directed translation? Write a syntax directed definition for a mathematical expression with + and – symbols for infix to prefix translation. Draw the parse tree with attribute values at nodes of the expression.

(4+4+10)

5.

a) Describe the use of Stack & Heap in runtime allocation.

b) To improve the target code we generally use copy propagation, code motion and reduction in strength. Explain and give examples in each case.

c) While generating codes from DAG show how can you get the optimal ordering of DAG to get a better code?

(6+6+6)

6.

a) Suppose you want to parse the string id +id*id. Show the operator precedence relation of id, + and *. Give the procedure for finding handle using the above precedence relation.

b) What are the different "type expressions" used in a language? Write a syntax directed definition with inherited or synthesized attribute for a simple desk calculator having +, - and ( ).

(10+8)

7.

a) When a grammar is said to be left recursive? Is there any problem of using left recursive grammar? Justify your answer.

b) Mention any lexical analyzer generator and write down the functionality of its different components.

c) Write down the goals of an error handler in a parser. What are the different error strategies used by a parser to recover from a syntactic error?

(5+5+8)