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.

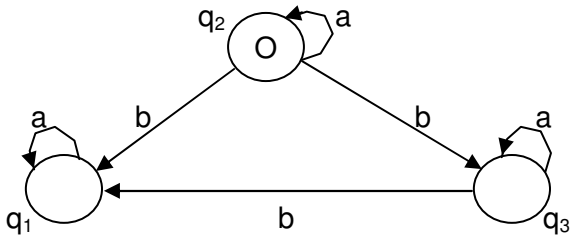
a) Construct an NFA for the regular expression 01* + 1.

b) What language is represented by (c*(aU(bc*))*)? Justify.

c) Design a Mealy machine to get a sequence of 1's, ignoring all 0's.

d) Describe a right shifting turing machine that shifts the string on the tape one position right.

e) Construct a regular expression for the language accepted by the given DFA:



f) Give one method of resolving collision in symbol tables.

g) Consider the DFA represented by the following transition table:

State	Input Symbol	
	a	b
A	B	C
A	B	D
C	B	C
D	B	E
E	B	C

Obtain an equivalent DFA that minimizes the number of states.

(7x4)

2.

a) Give transition table for Push Down Automata (PDA) recognizing the following language:
L = { w ε {a,b}*: w has the same number of a's and b's}.

b) Design CFG for the language $\{0^n 1^n \mid n >= 1\}$

c) Write a short-note on Church-Turing Thesis.

(8+6+4)

3.

a) Draw an ε-NFA (NFA allowing ε transitions) that accepts decimal numbers consisting of:
   i) An optional + or – sign
   ii) A string of digits
   iii) A decimal point

b) Show that the language L = { w ε $a^n b^n c^{2n}$ | w has an equal number of a's and b's and c's} is not context-free.

(9+9)

4.

a) Show that the complement of a regular language is also regular.

b) Show that the grammar $S \rightarrow 0A2$, $A \rightarrow 1A1$, $A \rightarrow 1$ is not an LR(0).

c) Show that the union of two context free languages is context free. However, the intersection of two context free languages may not be context free**.**

(6+6+6)

5. Consider the following grammar:

S -> e S -> SS

S -> (S)

a) Describe informally the language that this grammar generates.

b) Show that this grammar is ambiguous.

c) Define an equivalent unambiguous grammar.

(3+7+8)

6.

a) What role does semantic analysis play in compiler design? Give example of a semantic error that cannot be detected at compile stage.

b) Write a LEX program that reads an input file and counts the number of times a newline character appears in the file and outputs the final count.

c) Remove left recursion from the grammar:

E → E + T | T

T. → T * F | F

F → id

(6+6+6)

7.

a) Is the grammar given below an LR grammar? Justify.

E → E + E

E → E * E

E → id

b) Design a type checking scheme for the following grammar:

E → E + T

T → id | number

c) Describe the division of tasks between caller and callee subprograms, giving the contents of activation record.

d) Describe the terms common sub-expression elimination and dead code elimination in the context of code optimization:

(4+4+4+6)