B5.2-R4: AUTOMATA THEORY AND COMPILER DESIGN

«QP_SRLNO»

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) What do you mean by a language of a grammar G? Give an example.
b) When a grammar is said to be ambiguous?
c) What are regular languages? Define a regular expression.
d) What is meant by Empty Production removal in PDA?
e) What is an incremental compiler? Enlist the basic features of incremental compiler.
f) What are the benefits of using machine-independent intermediate form?
g) What are the limitations of using static allocation?

(7x4)

2.

a) Show that R is an equivalence relation in the following questions.
   - i) R is the relation on the set of ordered pairs of positive integers such that $(a,b), (c,d)$ Ɛ R whenever $ad = bc$.
   - ii) R is the relation on the set of positive integers such that (a,b) Ɛ R if and only if ab is a perfect square.

b) Use mathematical induction to prove that $n^3 + (n + 1)^3 + (n + 2)^3$ is divisible by 9, for $n \geq 0$.
c) Solve the recurrence relation $a_n - 2a_{n-1} = 3^n$; $a_1 = 5$.

(6+6+6)

3.

a) Design a finite state automaton that accepts the string of natural numbers (0 through 9) which are divisible by 3.
b) Compare a Mealy machine with a Moore machine. Construct a Mealy machine which is equivalent to the Moore machine given in the table below.

	Present state	**Next state**		**Output**
		$a = 0$	$a = 1$	
→	q0	q3	q1	0
	q1	q1	q2	1
	q2	q2	q3	0
	q3	q3	q0	0

(9+9)

4.

a) Find a reduced grammar equivalent to the grammar G whose productions are:

$S \rightarrow AB \mid CA$, $B \rightarrow BC \mid AB$, $A \rightarrow a$, $C \rightarrow aB \mid b$

b) Design a PDA that accepts $L = \{w \mid n_a(w) = 2n_b(w)\}$ or all strings over (a,b) in which the number of a's is twice the number of b's.

(9+9)

5.

a) Construct a Turing machine with tape symbols 0, 1 and B that will replace all 0's in the bit string with 1's and will not change any of the 1's in the bit string.

b) If L_1 and L_2 are recursively enumerable languages over $\sum$, then prove that L_1 U L_2 and $L_1 \cap L_2$ are also recursively enumerable.

c) Define passes of a compiler. Which are the factors that decide number of passes for a compiler?

(9+6+3)

6.

a) Consider the following grammar for postfix expressions:

E → E E +
E → E E *
E → **num**

i) Eliminate left recursion in the grammar.
ii) Do left-factorization of the grammar produced in part i).
iii) Calculate *Nullable*, *FIRST* for every production and *FOLLOW* for every non-terminal in the grammar produced in part ii).
iv) Make a LL(1) parse-table for the grammar produced in part ii).

b) What is a marker non-terminal symbol? Write a translation scheme to convert an infix expression to postfix form such that all actions appear at the end of the right hand side of production.

(12+6)

7.

a) Explain why a left-recursive grammar cannot be parsed using the predictive top-down parsing algorithms?

b) What are the criteria that need to be considered while applying the code optimization? Give the criteria for achieving machine dependent and machine independent optimization.

c) Create symbol table as list for the following program:

```
int a, b, c;
int sum (int x, int y)
{     a=x+y; return (a)}
main()     {   int u;
            u=sum(5, 6);
       }
```

(6+6+6)