B5.2-R4: AUTOMATA THEORY & 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 differences between contest free grammar and context sensitive grammar.
b) Design a DFA which accepts all strings which are ending with 0101 over an Alphabet {0, 1}.
c) What is Syntax directed definition?
d) Explain the role of the Parser.
e) What do you mean by ℓ-values and r-values of an identifier? How are they used in translation of expressions?
f) Suppose G is the grammar with the following production rules:

list→list+list
list→list-list
list→digit
digit→0|1|2|…..|9

Show that G is ambiguous and convert it into unambiguous one.
g) Illustrate with an example how a DAG can be used for code optimization.

(7x4)

2.

a) Design a DFA that recognizes the language of all string over the alphabet {0, 1} where the decimal equivalent of the strings in the language is divisible by 3. Check the acceptance of the string 1001.
b) Convert the following grammar into CNF.

S->aAD
A->aB|bAB
B->b
D->d.

(9+9)

3. Consider the following grammar:

D->TL
T-> int |float
L -> L, id |id

a) Write the Syntax Directed Definitions to add the type of each identifier to its entry in the symbol table during semantic analysis.
b) Draw an annotated parse tree for the declaration: float id1, id2, id3;

(9+9)

4.

a) Draw NFA for recognizing the language generated by (0+1)(01)*(011)*. Convert it into DFA. Check the acceptance of 1011 by both FAs.
b) Even though context free grammars are capable for specifying the tokens of a programming language, regular expressions and FAs are used for the lexical analysis phase. Justify this with proper reasons.
c) Explain the reasons for separating the lexical analysis from syntax analysis.

(9+6+3)

5.

a) Construct the SLR(1) parse table for the following grammar: S->0S0|1S1| 10.
b) Describe the Stack and Heap in runtime allocation.
c) What the different phases of a compiler? How can you categorize them into front?

(10+4+4)

6.

a) What is flow-graph? Explain how given program can be converted into flow-graph?
b) Explain the loop optimization?

(12+6)

7. Write Short Notes on **any three** of the following:

a) Non- Deterministic Turing machine
b) Kleen's theorem
c) Bacos Naur Form
d) Code improving transformation

(3x6)