**UNIT 1**

1.Convert the RE (a|b)\*abb into NFA E and find the equivalent minimum state DFA.

2.What is the role of transition diagrams in the construction of lexical analyzer?

3.Convert the RE (a|b)\*abb into NFA E and find the equivalent minimum state DFA.

4.What is the role of transition diagrams in the construction of lexical analyzer?

5.Show how an input a=b+c\*60 get processed in compiler. show the output at each stage of compiler.

6.Find the transition diagram for the Regular expression and the Regular definition

i.a(a|b)\*a

ii.((E|a)b\*)\*

iii.All the string of digit with at most one repeated digit

7.Explain different phases of a compiler with a neat diagram.

8.Construct minimum DFA for RE (0+1)\*(0+1)01.

**UNIT 2**

1.Check whether the following grammar is SLR(1) or not. Explain your answer the reason. S→ L=R ,S→R, L→ \*R ,L→ id, R→ L.

2.Generate the SLR parsing table for the following grammar

E→E+T,E→T, T→T\*F, T→F, F→ (E), F→id.

3.i. Consider the grammar: E→ E+E, E→ E\*E, E→id Perform shift reduce parsing of the input string “id1+id2+id3”

ii.Write the differences between SLR, CLR, LALR parsers.

4.Write an algorithm for constructing CLR parsing table Following Grammar S→CC, C→aC, C→d

5.Calculate FIRST and FOLLOW for the following grammar

E→E+T/T, T→T\*F/F, F→ (E)/id..

6.i. Write the differences between SLR, CLR, LALR parsers.

ii. List out of the types of Parsers available in compiler Design.

7.Construct the LALR parser for the following Grammar? S →CC , C →aC,C→d.

8.Generate the SLR parsing table for the following grammar

E→E+T,E→T, T→T\*F, T→F, F→ (E), F →id.

**UNIT 3**

1.i. Define S-attribute. Construct parse tree, syntax tree, annotated parse tree for the input string is 5\*6+7;.

ii. Define inherited attribute. Give one example

2.What is type checker? Explain the specification of a simple type checker.

3.i. Differentiate between L attribute and S attribute.

ii. Describe the evaluation order of SDT with an example

4.Write the translation scheme to generate intermediate code for assignment statements with array references

5.i. Define syntax tree. Give one example.

ii. Explain the Translation scheme of SDD.Give one example.

6.i. Define intermediate code? Translate the expression (a\*b)+(c+d)-(a+b) into quadruples, triples and indirect triples

ii. Explain in detail about Flow of control statements. Give one example.

7.Define type checker? Explain the specification of a simple type checker

8. i. Describe the evaluation order of SDT with suitable example.

ii. Explain in detail about dependency graphs with suitable example.

**UNIT 4**

1.i. Explain in brief about peephole optimization techniques.

ii. Explain different methods for register allocation and assignment.

2.Explain in detailed about description on DAG.Give one example.

3.Explain in brief about Heap Storage allocation strategy.

4.i. Differentiate between Static and Dynamic Storage allocation Strategies.

5. Explain in brief about simple code generator.

6.Define Activation Record? Explain its usage in stack allocation strategy. How it is different from heap allocation?

7.Define reference counting. What is the role of reference counting in garbage collection?

8.What is Activation Record? Explain its usage in stack allocation strategy. How it is different from heap allocation?

9.Define Symbol table? Explain about the data structures used for Symbol table.

**UNIT 5**

1.Explain in detail about:

i. Common sub expression

ii. Dead code elimination.

2.Explain in detail about:

i. Copy propagation.

ii.Constant folding.

3. i.Explain in detail about loop optimization technique with example.

ii. Explain in detail about Copy propagation and constant folding.

4. What is a Flow Graph? Explain how a given program can be converted in to a Flow graph?

5. What are loop invariant Computations? Explain how they affect the efficiency of a program.

6.Explain the following machine independent optimization techniques.

7.Explain in brief about different Principal sources of optimization techniques with suitable examples.

8.i. Explain in detail about redundancy elimination techniques.

ii. Explain in detail about constant propagation with example