

COMPILER DESIGN QUESTION BANK

UNIT-1

Short Answer Questions

1. Define Compiler and what are the phases of compiler.
2. What are the functions performed in synthesis phase.
3. Differentiate analysis and synthesis phase.
4. Differentiate token, pattern, lexeme
5. Define the following
 - i) Preprocessor
 - ii) Assembler
 - iii) Loader and Linker

Long Answer Questions

1. What are the different phases of compiler? Explain the phases in detail. Write down the phases in detail. Write down the output of each phase for the expression $a:=b+c*50$.
2. Write a short notes on compiler construction tools
3. Explain in detail about input buffering.
4. Give the minimized DFA for the following expression $(a/b)^*abb$
5. Elaborate specification and recognition of tokens.
6. Discuss the various phases of compiler and trace the program segment $c=a+b=4$ for all phases.
7. Discuss about the input buffering scheme in lexical analyzer.

UNIT-2

Short Answer Questions

1. What is phrase level error recovery
2. Eliminate the left recursion for the following grammar

$E \rightarrow E+T/T$

$T \rightarrow T*F/F$

$F \rightarrow (E)/id$

3. Write down the necessary algorithms for FIRST and FOLLOW
4. Explain the error recovery in predictive parsing
5. Define operator precedence grammar.

Long Answer Questions

1. Construct SLR(1) for the following grammar

$E \rightarrow E+T/T$

$T \rightarrow TF/F$

$F \rightarrow F*/a/b$. Show the acceptance for the string $w=a+b*a$

2. Check whether the grammar is LALR(1) but not SLR(1)

$S \rightarrow Aa/bAc/dc/bda$

$A \rightarrow d$

3. Construct predictive parser for the following grammar

$S \rightarrow (L)/a$

$L \rightarrow L,S/S$ and parse any input string.

4. Construct a canonical parsing table for the grammar given below.

$E \rightarrow E+T$

$F \rightarrow (E)$

$E \rightarrow T$

$F \rightarrow id$

$T \rightarrow T*F$

$T \rightarrow F$ and parse any string derived from the grammar

5. Construct an operator precedence table for the given grammar

$S \rightarrow L = R / R$

$L \rightarrow *R / id$

$R \rightarrow L$ and construct operator precedence function table from the parsing table.

6. Consider the following grammar.

$S \rightarrow AS|b, A \rightarrow SA|a$

Construct the SLR parse table for the grammar. Show the actions of the parser for the input string abab

7. Find LALR for the given grammar and parse the sentence $(a+b)^*$

$E \rightarrow E+T|T, T \rightarrow T*F|F, F \rightarrow (E)|id$

UNIT-3

Short Answer Questions

1. Define Syntax-directed translation

2. Define an attribute. Give the types of an attribute

3. Define the following

I) Annotated parse tree

II) Dependency graph

III) Syntax tree

4. Define DAG. Give an example

5. Define backpatching.

Long Answer Questions

1. Explain with an example to generate the intermediate code for the flow of control Statements.

2. Explain three address codes and its types. How would you implement the three Address statements? Explain with suitable examples.

- 3.A) Write a note on the specification on simple type checker
 - B) List the various ways of calling the procedures? Explain in detail.
4. Describe the method of generating syntax directed definition for control Statements.
5. Explain and compare in detail the various implementation forms of three address code.
6. How would you convert the following into intermediate code.
 - (i) Assignments statements.
 - (ii) Case statements.
7. Write three codes for $x := A[y, z]$ with detail explanation.

UNIT-4

Short Answer Questions

1. Define triples, indirect triples and quadruples.
2. Define activation record and list out the fields in it.
3. Translate the arithmetic expression $a * -(b + c)$ into syntax tree and postfix notation.
4. What are the types of three address statements.
5. Define loop unrolling with an example.

Long Answer Questions

1. Explain the various issues in the design of code generation
2. Write a short notes on basic blocks and flow graphs.
3. Explain the peephole optimization in detail.
4. Explain the sequence of stack allocation process for a function call.
5. Define a Directed Acyclic graph. Construct a DAG and write the sequence of Instructions for the expression $a + a * (b - c) + (b - c) * d$.
6. Discuss the various issues in code generation with examples.
7. What is an activation record? Explain how it is related with run time storage organization

UNIT-5

Short Answer Questions

1. Define Peephole Optimization
2. Give the applications of DAG
3. What is machine dependent code optimization
4. Define flowgraph.
5. What is stack allocation in compiler design.

Long Answer Questions

1. Explain principle sources of optimization in details.
2. Explain the machine dependent and independent techniques
3. What is the purpose of code optimization? Explain in detail about loop Optimization with example.
4. What is an activation record? Explain how it is related with run time storage organization.
5. Discuss in detail the process of optimization of basic blocks.
Give an example.
6. Write global common subexpression elimination algorithm with an example.
7. Discuss the various peephole optimization techniques in detail