CS450 - Design of Language Processor

Unit-I & II-Question bank

1. What is the difference between a compiler and interpreter?
2. Explain the advantages of compiler over an interpreter.
3. Draw and Explain the structure of compiler.
4. Explain applications of compiler technology.
5. Explain the role of Lexical analyzer.
6. Explain LEX: lexical analyzer generator.
7. Explain recognition of reserved words and identifiers.
8. Define: alphabet, string and language and also define terms for partsof string.
9. Explain operations on languages and regular definitions.
10. Explain Lexical analysis versus parsing and define token, patterns and lexemes.
11. Explain the structure of LEX program
12. Explain phases of compiler with example.
13. Define cross compiler, token and handle, yacc.
14. Write a short note on input buffering method.
15. Difference between syntax tree and parse tree.
16. Explain the roles of Linker and Loader.
17. Explain types of compiler.
18. Compare one pass and two pass compilers.
19. Explain role of finite automata in the compiler.
20. Discuss error handling methods in the syntax analysis phase.
21. What is the difference between parse tree and syntax tree.
22. Discuss various error recovery strategies of compilers.
23. Draw transition diagram for relational operators.
24. What is symbol table? For what purpose compiler uses symbol table?

CS450 - Design of Language Processor

Unit-III Question bank

1. Find FOLLOW for given grammar

   $S \rightarrow ACB / CbB / Ba$, $A \rightarrow da / BC$, $B \rightarrow g / \epsilon$, $C \rightarrow h / \epsilon$
2. Determine whether following grammar if of type LL(1) or not? If it is of type LL(1) then prepare parsing table.

   $E \rightarrow TE'$, $E' \rightarrow +TE' / \epsilon$, $T \rightarrow FT'$, $T' \rightarrow *FT' / \epsilon$, $F \rightarrow id / (E)$
3. Prepare LALR(1) parsing table for given grammar

   $S \rightarrow AaAb / BaBa$, $A \rightarrow \epsilon$, $B \rightarrow \epsilon$
4. Prepare CLR(1) parsing table for given grammar

   $S \rightarrow Aa / bAc / Bc / bBa$, $A \rightarrow d$, $B \rightarrow d$
5. Prepare SLR(1) parsing table for given grammar

   $S \rightarrow dA / aB$, $A \rightarrow bA / c$, $B \rightarrow bB / c$
6. Prepare SLR(1) parsing table for given grammar

   $S \rightarrow AA$, $A \rightarrow aA / b$
7. Explain LR (0) parsing technique
8. Explain SLR (1) parsing technique
9. Explain CLR (1) parsing technique
10. Explain LALR (1) parsing technique
11. Explain error recovery strategies used by parser.
12. Define handle and handle pruning. Explain the stack implementation of shift reduce parser with the help of example.
13. Check given grammer is LL(1) but not SLR(1).

    $S \rightarrow AaAb \mid BbBa$

    $A \rightarrow \epsilon$

    $B \rightarrow \epsilon$
14. Construct CLR parsing table for the following grammer.

S → CC

C → cC | d

15. Find out FIRST and FOLLOW for the following grammer.

S →1AB | ϵ

A → 1AC | 0C

B → 0S

C → 1

16. Find out FIRST and FOLLOW set for all the Nonterminals.

S →AcB | cbB | Ba

A → da | BC

B → g | ϵ

C → h | ϵ

17. For the following grammer

D → TL;

L → L,id | id

T → int | float

(1) Remove left recursion(if required)

(2) Find first and follow for each non terminal for Resultatnt grammer

(3) Construct LL(1) parsing table

(4) Parse the following string (show stack actions clearly) and draw parse tree for the input: int id,id;

18. Test whether grammer is LL(1) or not & construct parsing table.

S → AaAb | BbBa

A → ϵ

B → ϵ

19. Test whether the following grammer is LL(1) or not. Construct predictive parsing table for it.

S → 1AB | ϵ

A → 1AC | 0C

B → 0S

C → 1

20. Draw parsing table for table driven parser for the given grammer is the grammer LL(1)?

    A → AaB | x

    B → BCb | Cy

    C → Cc | ϵ

21. Construct an LALR(1) parsing table for the following grammer:

    S → Aa | bAc | dc | bda

    A → d

22. Write SLR parsing table for: S → T, T → CC, C → cC, C → d.

23. Construct LL(1) parsing table for the following grammer.Also show moves made by input string : abba.

    S → aBa

    B → bB | ϵ

24. What is left factoring and left recursion? Explain it with suitable example.

25. Construct CLR parsing table for following grammer.

    S → aSA | ϵ

    A → bS | c

26. Define Ambiguous grammer,handle pruning.

CS450 - Design of Language Processor

Unit-IV Question bank

1. Give the translation scheme that converts infix to postfix notation. Generate the annotated parse tree for input string 3-5+4.
2. Define syntax tree. What is S-attributed definition? Explain construction of syntax tree for the expression a-4+c using SDD.
3. Translate the arithmetic expression a*-(b + c) into
    1) Syntax tree
    2) Postfix notation
    3) Three address code
4. Write syntax directed definition to produce three address code for the expression containing the operators := , + , - (unary minus), () and id.
5. What is importance of intermediate code? Discuss various representations of three address code using the given expression.
   A = b* - c + b* - c.
6. Define following : DAG, Basic Blocks,Flow graph.
7. What is inherited attribute? Write syntax directed definition with inherited attributes for type declaration for list of identifiers.
8. Explain quadruples and Triples form of three address code with example.
9. Draw a DAG for expression: a + a*(b - c)+(b - c)*d.
10. Write syntax directed definition for simple desk calculator. Using this definition, draw annotated parse tree for 3*5+4 n.

CS450 - Design of Language Processor

Unit-V Question bank

1. Explain peephole optimization method.
2. Discuss the issues in the design of code generator.
3. Explain Dynamic storage allocation technique.
4. Discuss any three methods for code optimization.
5. Explain Activation record and Activation tree in brief.
6. Explain the calling sequence with an example.
7. list and elaborate on the issues of the code generator.
8. list code optimization techniques and explain any 3.