Course Code: 20CS0516 **R20**

SIDDARTHA INSTITUTE OF SCIENCE AND TECHNOLOGY:: PUTTUR
Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (DESCRIPTIVE)

Subject with Code : Compiler Design (20CS0516) **Course & Branch :** B. Tech - CSE
Year & Sem : III B.Tech & I-Sem **Regulation :** R20

UNIT –I
INTRODUCTION AND LEXICAL ANALYSIS

1	**a**	What do you understand by language processor?	[L2][CO1]	**[2M]**
	b	Describe about different language processors used in compiler design	[L2][CO1]	**[4M]**
	c	Give the differences between compiler and interpreter.	[L4][CO1]	**[6M]**
2	**a**	Define compiler.	[L1][CO1]	**[2M]**
	b	Analyse the process of compilation while designing a compiler.	[L4][CO2]	**[10M]**
3	**a**	List all the phases of compiler	[L1][CO2]	**[2M]**
	b	Give the neat diagram of phase of a compiler	[L2][CO2]	**[4M]**
	c	Explain each phase of a compiler.	[L2][CO2]	**[6M]**
4		Design the compiler by using the source program position=intial+rate*60.	[L6][CO3]	**[12M]**
5	**a**	Analyze the reasons for separating the lexical analysis and syntax analysis.	[L4][CO2]	**[4M]**
	b	Illustrate the steps involved in designing the compiler by using the source program a=b+c*10.	[L3][CO3]	**[8M]**
6	**a**	Describe Bootstrapping	[L2][CO1]	**[8M]**
	b	Explain the different applications of compiler technology	[L2][CO1]	**[4M]**
7	**a**	Discuss the Compiler construction Tools	[L2][CO3]	**[6M]**
	a	Differentiate tokens, patterns, and lexeme.	[L4][CO1]	**[6M]**
8	**a**	Explain in detail about the role of lexical analyzer in Compiler Design.	[L2][CO1]	**[6M]**
	b	Write about input buffering?	[L3][CO1]	**[6M]**
9		Discriminate the following terms a) Specification of Tokens b) Recognition of Tokens	[L5][CO1]	**[12M]**

10	**a**	What is LEX	[L2][CO3]	**[2M]**
	b	Explain the working of a LEX Tool	[L2][CO3]	**[6M]**
	c	Give the structure of LEX program	[L2][CO3]	**[4M]**

Course Code: 20CS0516 **R20**

UNIT –II

SYNTAX ANALYSIS AND TOP DOWN PARSING

1	**a**	Explain the role of parser.	[L2][CO1]	**[4M]**
	b	Define Context Free Grammar with example.	[L1][CO1]	**[4M]**
	c	Compare left most and right most derivations with examples	[L4][CO1]	**[4M]**
2	**a**	Define parse tree.	[L1][CO2]	**[2M]**
	b	Construct Leftmost and Rightmost derivation and parse tree for the string 3*2+5 from the given grammar. Also check it's ambiguity for Set of alphabets $\sum = \{0,\ldots,9, +, *, (,)\}$ E → I E → E + E E → E * E E → (E) I → ε | 0 | 1 | … | 9	[L6][CO2]	**[10M]**
3	**a**	Define Ambiguity.	[L1][CO1]	**[2M]**
	b	Interpret how to eliminate ambiguity for the given Ambiguous Grammar.	[L3][CO1]	**[10M]**
4	**a**	What is left recursion? Describe the procedure of eliminating Left recursion.	[L5][CO1]	**[4M]**
	b	Eliminate left recursion for the following grammar E→E+T/T T→T*F/F F→(E)/id	[L1][CO1]	**[4M]**
	c	Show what you understand by Left factoring. Perform left factor for the grammar A→abB/aB/cdg/cdeB/cdfB	[L2][CO1]	**[4M]**
5	**a**	List the types of Parsers available	[L1][CO2]	**[4M]**
	b	Design the recursive decent parser for the following grammar E→ E+T/T T→T*F/F F→(E)/id	[L6][CO3]	**[8M]**
6	**a**	What is meant by Non-recursive predictive parsing	[L2][CO3]	**[2M]**
	b	Illustrate the rules to be followed in finding the FIRST and FOLLOW.	[L3][CO1]	**[6M]**
	c	Find FIRST and FOLLOW for the following grammar? E→ E+T/T T→T*F/F F→(E)/id	[L3][CO2]	**[4M]**

7		Consider the grammar S→AB | ABad A→d E →b D→b | ε B→c Construct the predictive parse table and check whether the given grammar is LL(1) or not.	[L6][CO3]	**[12M]**
8		Consider the grammar E→TE' E'→+TE' | -TE' | ε T→FT' T'→*FT' | / FT' | ε F→GG' G'→^F / ε G→(E) / id Calculate FIRST and FOLLOW for the above grammar and Construct LL(1) Table for the above grammar.	[L4][CO2]	**[12M]**
9		Consider the grammar E→E+T/T, T→T*F/F, F→(E)|id Design predictive parsing table and check the given grammar is LL(1) or not?	[L6][CO3]	**[12M]**
10	**a**	Discuss the types of errors.	[L2][CO2]	**[6M]**
	b	Explain Error recovery in predictive parsing with an Example.	[L2][CO2]	**[6M]**

Course Code: 20CS0516 **R20**

UNIT –III

BOTTOM UP PARSING AND SEMANTIC ANALYSIS

1	**a**	Explain about handle pruning	[L2][CO1]	**[6M]**
	b	Summarize about LR parsing	[L2][CO1]	**[6M]**
2	**a**	Describe bottom up parsing	[L1][CO2]	**[4M]**
	b	Differences between SLR, CLR, LALR parsers	[L4][CO2]	**[8M]**
3		Prepare Shift Reduce Parsing for the input string using the grammar S→(L)|a L→L,S|S a. (a,(a,a)) b. (a,a)	[L6][CO3]	**[12M]**
4	**a**	Define augmented grammar.	[L1][CO2]	**[2M]**
	b	Construct the LR(0) items for the following Grammar S→L=R / R L→*R / id R→L	[L6][CO3]	**[10M]**
5		Construct SLR Parser for the following grammar E→E+T / T T→TF / F F→F* / a / b	[L6][CO3]	**[12M]**
6		Construct CLR Parsing table for the given grammar S→CC C→aC/d	[L6][CO3]	**[12M]**
7		Design the LALR parser for the following Grammar S → AA A → aA A →b	[L6][CO3]	**[12M]**
8	**a**	What is YACC parser?	[L1][CO3]	**[2M]**
	b	Explain in detail the processing procedure of YACC Parser generator tool.	[L2][CO3]	**[6M]**
	c	How YACC will resolve the parsing action conflicts and the error recovery.	[L2][CO3]	**[4M]**
9	**a**	Explain syntax directed definition with example	[L2][CO2]	**[6M]**
	b	Define a syntax-directed translation and explain with example.	[L2][CO2]	**[6M]**
10	**a**	Give the evaluation order of SDD with an example.	[L5][CO2]	**[6M]**
	b	Discuss Type Checking with suitable examples.	[L2][CO4]	**[6M]**

Course Code: 20CS0516 **R20**

UNIT –IV

INTERMEDIATE CODE GENERATION AND RUN TIME ENVIRONMENT

1	**a**	What do you understand by Intermediate Code	[L2][CO5]	**[2M]**
	b	Analyse different types of Intermediate Code with an example.	[L4][CO5]	**[10M]**
2	**a**	List and define various representation of Three Address Codes	[L1][CO5]	**[4M]**
	b	Explain representation of Three Address Codes with suitable Examples	[L2][CO5]	**[8M]**
3		Produce quadruple, triples and indirect triples for following expression: (x + y) * (y + z) + (x + y + z)	[L6][CO5]	**[12M]**
4	**a**	Describe scope and life time of variable.	[L2][CO4]	**[2M]**
	b	Illustrate Control Flow Statements.	[L3][CO4]	**[10M]**
5	**a**	Justify the need for Storage Organization.	[L6][CO4]	**[4M]**
	b	Describe the Storage Organization with simple examples.	[L2][CO4]	**[8M]**
6	**a**	List out the properties of memory management	[L1][CO4]	**[4M]**
	b	Discuss Storage allocation strategies with suitable example	[L2][CO4]	**[8M]**
7		Evaluate the following terms i. Stack allocation ii. Static allocation iii. heap allocation	[L5][CO4]	**[12M]**
8	**a**	Define Activation Record.	[L1][CO5]	**[2M]**
	b	Sketch the format of Activation Record in stack allocation and explain each field in it.	[L3][CO5]	**[10M]**
9	**a**	Discuss about symbol table entries.	[L2][CO4]	**[6M]**
	b	Describe the various operations on symbol table.	[L2][CO4]	**[6M]**
10	**a**	Define Symbol table.	[L1][CO4]	**[2M]**
	b	Explain different types of Data structure used for symbol table.	[L2][CO4]	**[10M]**

UNIT –V

CODE OPTIMIZATION AND CODE GENERATION

1		Interpret the principles of optimization techniques to be considered during code generation.	[L3][CO5]	**[12M]**
2	**a**	Discuss about function preserving transformations.	[L2][CO6]	**[6M]**
	b	Describe about loop optimization technique.	[L2][CO5]	**[6M]**
3		Explain the following i) Basic blocks ii) Flow Graphs	[L3][CO6]	**[12M]**
4	**a**	List the optimization techniques of basic blocks	[L1][CO6]	**[4M]**
	b	Analyse different types of optimization techniques of basic blocks	[L4][CO6]	**[8M]**
5	**a**	Create the DAG for following statement. a+b*c+d+b*c	[L6][CO6]	**[6M]**
	b	Construct the DAG for the following basic blocks 1. t1:=4*i 2. t2:=a[t1] 3. t3:=4*i 4. t4:=b[t3] 5. t5:=t2*t4 6. t6:=prod+t5 7. prod:=t6 8. t7:=i+1 9. i:=t7 **if i<=20 goto 1**	[L6][CO6]	**[6M]**
6	**a**	List out the properties of global data flow analysis and explain it.	[L2][CO6]	**[6M]**
	b	Discuss about machine dependent optimization	[L2][CO5]	**[6M]**
7		Explain the peephole optimization Technique with examples.	[L2][CO5]	**[12M]**
8	**a**	List all the issues in the design of a code generator	[L2][CO6]	**[4M]**
	b	Explain the issues to be handled when code generator is designed.	[L2][CO6]	**[8M]**
9	**a**	Analyse the different forms in target program.	[L4][CO6]	**[6M]**
	b	Explain the target machine in code generator.	[L2][CO6]	**[6M]**
10	**a**	Analyze Simple code generator	[L4][CO6]	**[6M]**
	b	Evaluate Register allocation and register assignment techniques.	[L5][CO6]	**[6M]**

SIDDHARTH GROUP OF INSTITUTIONS :: PUTTUR
Siddharth Nagar, Narayanavanam Road – 517583
QUESTION BANK (OBJECTIVE)

Subject with Code : Compiler Design (20CS0516) **Course & Branch :** B. Tech - CSE
Year & Sem : III B.Tech & I-Sem **Regulation :** R20

UNIT-I

1.A language processor is a special type of computer software that has the capacity of translator for translating the ________ into machine codes. [ ]

A) Source code B) Program codes C) Both A & B D) 3 address code

2.Which of the following language processors? [ ]

A) Compiler B) Assembler C) Interpreter D) All the above

3.The language processor that reads the complete source program written in high-level language as a whole in one go and translates it into an equivalent program in machine language is called a __________ [ ]

A) Compiler B) Assembler C) Interpreter D) All the above

4.The __________ is used to translate the program written in Assembly language into machine code. [ ]

A) Compiler B) Assembler C) Interpreter D) All the above

5.The translation of a single statement of the source program into machine code is done by a language processor and executes immediately before moving on to the next line is called an ___________ [ ]

A) Compiler B) Assembler C) Interpreter D) All the above

6.There are _______ parts in Compiler [ ]

A) 2 B) 6 C) 4 D) 5

7. The _________ part of the compiler breaks up the source program into constituent pieces and imposes a grammatical structure on them. [ ]

A) logical B) analysis C) interface D) synthesis

8. The _________ part of the compiler constructs the desired target program from the intermediate representation and the information in the symbol table. [ ]

A) logical B) analysis C) interface D) synthesis

9. The first four phases is often called the _______ of the compiler and last two phases is called the _______ [ ]

A) front end, back end B) first pass, second pass

C) analysis part, synthesis part D) All the above

10. The Compiler has _____ Phases [ ]

A) 2 B) 6 C) 4 D) 5

11. The first phase of a compiler is called _________ [ ]

A) lexical analysis B) syntax analysis C) semantic analysis D) code generator

12. The lexical analyzer reads the stream of characters making up the source program and groups the characters into meaningful sequences called ________ [ ]

A) code B) syntax C) lexemes D) program

13. In a compiler, keywords of a language are recognized during (GATE 2011) [ ]

A) parsing of the program

B) the code generation

C) the lexical analysis of the program

D) dataflow analysis

14. The lexical analysis for a modern computer languages such as Java needs the power of which one of the following machine models is necessary and sufficient sense. (GATE 2011) [ ]

A) Finite state automata

B) Deterministic pushdown automata

C) Non-Deterministic pushdown automata

D) Turing Machine

15. The second phase of the compiler is syntax analysis is also called ______ [ ]

A) code checker B) tree generator C) scanner D) parsing

16. The parser uses the first components of the tokens produced by the lexical analyzer to create a _________ representation that depicts the grammatical structure of the token stream. [ ]

A) parse tree B)syntax tree C) tree-like intermediate D) All the above

17. The semantic analyzer uses the syntax tree and the information in the symbol table to check the source program for _______ with the language definition. [ ]

A) syntax B) semantic consistency

C) intermediate representation D) all the above

18. An important part of semantic analysis is_______, where the compiler checks that each operator has matching operands. [ ]

A) type checking B) tree generator C) scanner D) parsing

19. An intermediate form called _______, which consists of a sequence of assembly-like instructions with three operands per instruction. [ ]

A) three-variable code B) three-address code

C) three-operand code D) three-instruction code

20. A simple intermediate code generation algorithm followed by _______ is a reasonable way to generate good target code. [ ]

A) code optimization B) code generator C) code scanner D) code parsing

21. Match all items in Group 1 with correct options from those given in Group 2.

(GATE 2009) [ ]

Group 1	Group 2
P. Regular expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

A)P-4. Q-1, R-2, S-3 B) P-3, Q-1, R-4, S-2

C) P-3, Q-4, R-1, S-2 D) P-2, Q-1, R-4, S-3

22. The ________ takes as input an intermediate representation of the source program and maps it into the target language. [ ]

A) lexical analysis B) syntax analysis C) semantic analysis D) code generator

23. The ________ is a data structure containing a record for each variable name, with fields for the attributes of the name. [ ]

A) lexical table B) syntax table C) symbol table D) code table

24. In a two-pass assembler, symbol table is (GATE 2015) [ ]

A) Generated in first pass B) Generated in second pass

C) Not generated at all D) Generated and used only in second pass

25. __________ is a process in which simple language is used to translate more complicated program which in turn may handle for more complicated program. [ ]

A) Lexeme B) Data flow C) Input buffer D) Bootstrapping

26. Commonly used compiler-construction tools include [ ]

A) Parser generators B) Scanner generators

C) Data-flow analysis engines D) All the above

27. Compiler optimizations must meet the following design objectives: [ ]

a) The optimization must be correct, that is, preserve the meaning of the compiled program

b) The optimization must improve the performance of many programs

c) The compilation time must be kept reasonable

d) The engineering effort required must be manageable

A) Only a & b   B) Only c & d   C) a, c & d   D) All the above

28. The lexical analyzers are divided into two processes: [ ]

A) Scanning, Lexical analysis   B) Analysis part, Synthesis part

C) Front end, Back end   D) Scanner, Parser

29. Which of the following is False with regard with token: [ ]

A) A token is a pair consisting of a token name and an optional attribute value.

B) The token name is an abstract symbol representing a kind of lexical unit.

C) A token should have syntactic meaning represented.

D) Token is a group of characters with logical meaning.

30. The number of tokens in the following C statement is

printf("i = %d, &i = %x", i, &i); (GATE 2000) [ ]

A) 3   B) 26   C) 10   D) 21

31. The output of a lexical analyzer is_______ [ ]

A) A parse tree   B) intermediate code

C) Machine code   D) Stream of tokens

32. _______ is a rule that describe the characters that can be grouped into tokens. [ ]

A) Lexeme   B) Error   C) Token   D) Pattern

33. ________ is a sequence of characters in the source program that matches with the pattern for a token. [ ]

A) Lexeme   B) Error   C) Token   D) Pattern

34. A ______ error occurs when a sequence of characters does not match the pattern of any token. [ ]

A) syntax   B) Lexical   C) logical   D) Semantic

35. Two pointers to the input are maintained in Two Buffer Scheme are: [ ]

A) lexemeBegin, forward   B) lexBegin, forward

C) lexemeStrat, forward   D) lexStrat, forward

36. _________ are an important notation for specifying lexeme patterns. [ ]

A) Pattern expressions   B) Regular expressions

C) Lexical expressions   D) Semantic expressions

37. Operations on Languages are [ ]

A) Union, Concatenation B) Union, Concatenation, Kleen

C) Union, Concatenation, Positive D) Union, Concatenation, Kleen, Positive

38. Transition diagrams have a collection of ______ called states. [ ]

A) nodes B) edges C) arrows D) input

39. Lex program when compiled generates ______ program [ ]

A) Lex B) C C) a.out D) exec

40. Structure of Lex program has ______ sections [ ]

A) 3 B) 2 C) 1 D) 4

UNIT-II

1.The syntax of programming language constructs can be specified by _____ [ ]

A) syntax analyzer B) context-free grammars

C) parser D) scanner

2.The parsing methods used in compilers can be classified as _______ [ ]

A) top-down B) bottom-up C) both A & BD) Grammar

3._______ methods build parse trees from the top to the bottom [ ]

A) Top-down B) Bottom-up C) Scanner D) Lexical

4._______ methods start from the leaves and work their way up to the root [ ]

A) Top-down B) Bottom-up C) Scanner D) Lexical

5.Which of the following are Syntax Error Handling [ ]

A) panic-mode and phrase-level scanner

B) panic-mode and phrase-level syntax former

C) panic-mode and phrase-level recovery

D) panic-mode and phrase-level parser

6.________ include misspellings of identifiers, keywords, or operators. [ ]

A) Lexical errors B) Syntactic errors C) Semantic errors D) Logical errors

7.________ include misplaced semicolons or extra or missing braces. [ ]

A) Lexical errors B) Syntactic errors C) Semantic errors D) Logical errors

8._______ include type mismatches between operators and operands. [ ]

A) Lexical errors B) Syntactic errors C) Semantic errors D) Logical errors

9.______ can be anything from incorrect reasoning on the part of the programmer. [ ]

A) Lexical errors B) Syntactic errors C) Semantic errors D) Logical errors

10. Which type of error compiler cannot handle? [ ]

A) Lexical errors B) Syntactic errors C) Semantic errors D) Logical errors

11. Which of the following is not the goal of the error handler in a parser? [ ]

A) Report the presence of errors clearly and accurately.

B) Recover from each error quickly enough to detect subsequent errors.

C) Rectify the error and process the program.

D) Add minimal overhead to the processing of correct programs.

12. A context-free grammar consists of [ ]

A) terminals & non-terminals B) productions

C) start symbol D) All the above

13. Beginning with the start symbol, each rewriting step replaces a nonterminal by the body of one of its productions is called _______ [ ]

A) derivation B) substitution C) generation D) production

14. Beginning with the______ , each rewriting step replaces a nonterminal by the body of one of its productions is called derivation. [ ]

A) terminals B) productions C) start symbol D) non-terminals

15. If G is the grammar with productions S->SaS / aSb / bSa / SS / ∈ (GATE 2017)

Where S is the start variable, then which one of the following is not generated by G? [ ]

A) abab B) aaab C) abbaa D) babba

16. A parse tree is a graphical representation of a derivation that filters out the order in which productions are applied to replace _______ [ ]

A) terminals B) productions C) start symbol D) non-terminals

17. A ______ is a graphical representation of a derivation that filters out the order in which productions are applied to replace non-terminals [ ]

A) parse tree B) lexeme tree C) semantic tree D) terminals tree

18. Each interior node of a parse tree represents the application of a_______. [ ]

A) terminals B) productionsC) start symbol D) non-terminals

19. A grammar that produces more than one parse tree for some sentence is said to be [ ]

A) ambiguous B) duplicate C) effective D) error

20. A CFG is ambiguous if [ ]

A) It has more than one rightmost derivation

B) It has more than one leftmost derivation

C) No parse tree can generated for the CFG

D) A or B

21. A grammar is _______ if it has a nonterminal A such that there is a derivation $A \overset{+}{\Rightarrow} A\alpha$ for some string a. [ ]

A) left factoring B) left recursive C) recursive-descent D) top-down parser

22. _______ is a grammar transformation that is useful for producing a grammar suitable for predictive, or top-down parsing. [ ]

A) left factoring B) left recursive C) recursive-descent D) top-down parser

23. A _______ program consists of a set of procedures, one for each nonterminal. [ ]

A) left factoring B) left recursive C) recursive-descent D) top-down parser

24. A recursive-descent parsing program consists of a set of procedures, one for each_______. [ ]

A) terminal B) productions C) start symbol D) non-terminal

25. _______ is a set of terminal symbols that begin in strings derived from α. [ ]

A) First(α) B) Follow(α) C) Non-Terminal(α) D) Prodution(α)

26. ________ is a set of terminal symbols that appear immediately to the right of α. [ ]

A) First(α) B) Follow(α) C) Non-Terminal(α) D) Prodution(α)

27. Algorithm for calculating Follow set includes [ ]

a) if α is a start symbol, then FOLLOW() = $

b) if α is a non-terminal and has a production α → AB, then FIRST(B) is in FOLLOW(A) except Ɛ.

c) if α is a non-terminal and has a production α → AB, where B Ɛ, then FOLLOW(A) is in FOLLOW(α).

A) a and b B) a and c C) b and c D) a, b and c

28. Calculate the first function for S for the given grammar-

S → AaAb / BbBa, A → ∈, B → ∈ [ ]

A) First(S) = { a , b } B) First(S) = { ∈ } C) First(S) = { a } D) First(S) = { b }

29. Consider the grammar P→xQRS Q→yz / z R→w / ∈ S→y

What is the FOLLOW (Q)? (GATE 2017) [ ]

A) {R} B) {w} C) {w,y} D) {w,$}

30. Calculate the follow function A for the given grammar-

S → AaAb / BbBa, A → ∈, B → ∈ [ ]

A) Follow(A) = { a , b } B) Follow(A) = { ∈ }

C) Follow(A) = { a } D) Follow(A) = {b}

31. Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order (GATE 2000). [ ]

A) Leftmost derivation

B) Leftmost derivation traced out in reverse

C) Rightmost derivation

D) Rightmost derivation traced out in reverse

32. Which one of the following is a top-down parser? (GATE 2007) [ ]

A) Recursive descent parser B) Operator precedence parser

C) LR(k) parser D) LALR(k) parser

33. Predictive parsers is also called as [ ]

A) Recursive-descent parsers B) Backtracking parsers

C) Recursive parsers D) Look-ahead parser

34. Predictive parsers, that is, recursive-descent parsers needing no backtracking, can be constructed for a class of grammars called _______ [ ]

A) LL(0) B) LL(1) C) LR(0) D) LR(1)

35. The first "L" in LL(1) stands for

A) scanning the input from left to right

B) producing a leftmost derivation

C) lookahead at each step to make parsing action decisions

D) All the above

36. The second "L" in LL(1) stands for [ ]

A) scanning the input from left to right

B) producing a leftmost derivation

C) lookahead at each step to make parsing action decisions

D) All the above

37. Which of the following be sufficient to convert a CFG to an LL(1) grammar? (GATE 2003). [ ]

A) Removing left recursion alone

B) Factoring the grammar alone

C) Removing left recursion and factoring

D) None of these

38. While parsing the input string the end of file is represented by the special symbol is ____ [ ]

A) $ B) υ C) μ D) π

39. The input buffer contains the string to be parsed, followed by the end-marker ____ [ ]

A) $ B) υ C) μ D) π

40. _________ error recovery is based on the idea of skipping symbols on the input until a token in a selected set of synchronizing tokens appears [ ]

A) phrase-level recovery B) Panic-mode C) Logical D) None above

UNIT-III

1.The process of reducing a string w to the start symbol of the grammar is known as_____ [ ]

A)Parsing B)Top Down Parsing

C)Bottom up parsing D) both B and C

2. In Which parsing techniques the construction of parse from leaves to root? [ ]

A)Top Down Parsing B)Syntax tree

C)Three address code D)Bottom up parsing

3._________is a sub string that matches the body of the production, reduction represents in one-step along the reverse of rightmost derivation. [ ]

A)Handle Pruning B) Pruning C)Handle D)Error Handling

4.Shift Reducing is the form of which parsing technique? [ ]

A)Top Down Parsing B)Bottom up parsing

C)Handle Pruning D) Precedence parsing

5.How many data structures can be used in shift reduce parsing? [ ]

A)2 B)3 C)4 D)1

6.Which of the following data structures are used in shift reduce parsing? [ ]

A) Stack, Queue B) Stack , Input Buffer

C) Shift , Reduce D) stack only

7. How many possible operations are considered for the shift reduce parsing? [ ]

A) 4 B) 2 C) 3 D)1

8. Which of the following are the possible actions in shift reduce parsing? [ ]

A)Shift, reduce, accept, error B) shift, reduce only

C) accept, error only D)shift, reduce and accept only

9. How many types of LR parsing Techniques are available in Bottom Up Parsing? [ ]

A) 2 B) 4 C) 3 D) 5

10. Which of the following is not a LR Parsing Techniques? [ ]

A) SLR parsing B)CLR parsing C) Operator grammar D) LALR parsing

11. In LR(k) parsing the R stands for ________ [ ]

A) Constructing a left most derivation in reverse

B) Constructing a right most derivation

C) Constructing a right most derivation in reverse

D) both A & C

12. In which item a grammar G is a production of G with a dot at some position of the body. [ ]

A)LR(0) B) LR C) LR(2) D) all of the above

13. Which of the following parts are represented for the construction of LR parsing table ? [ ]

A)ACTION , ACTION 1 B)GOTO, GOTO1

C)Stack , Input buffer D)ACTION, GOTO

14. Which of the following are look ahead LR Parser? [ ]

A)SLR B)SLR(1) C) LALR D) CLR

15. What is the similarity between SLR, CLR and LALR parser? [ ]

A) Use same algorithm, but different parsing table

B) Same parsing table, but different algorithm

C) Their Parsing tables and algorithm are similar but uses top down approach

D) Both Parsing tables and algorithm are different

16. Which of the following one is more powerful parser in LR Parsing? [ ]

A) CLR B) SLR C) LALR D)All 3 parsers

17. The construction of the canonical collection of the sets of LR (1) items are similar to the construction of the canonical collection of the sets of LR (0) items. Which is an exception? [ ]

A) Closure and goto operations work a little bit different

B) Closure and goto operations work similarly

C) Closure and additive operations work a little bit different

D) Closure and associatively operations work a little bit different

18. YACC stands for _______ [ ]

A) Yet Another Compiler B) Yet Another Compiler Compiler

C) Yet any compiler comiler D) Yes another compiler compiler

19. Which of the following one is a parser generator tool? [ ]

A) LEX B) YACC

C) A & B D) Data Flow Engine

20. Yacc is available as a command on which operating system, and has been used to help implement many production compilers? [ ]

A) UNIX B) LINUX C) UBUNTU D) both A & B

21. The Syntax Directed Definition is the combination of __________ [ ]

A) CFG + Association Rules B) CFG + Semantic Rules

C) CFG+ Production rules D) CFG+CSG

22. How many attributes are used in defining syntax directed definition? [ ]

A) 3 B) 4 C) 2 D) 1

23. Which of the following attributes are used in syntax directed definition? [ ]

A) Synthesized and inherited B) Synthesized only

C) Inherited attribute only D) S and D attributes

24. In which attribute, at node N is defined only in terms of attribute values at the children of N and at N itself. [ ]

A) Synthesized attribute. B) Inheritted attribute

C) both A & B D) B only

25. In synthesized attribute node value is calculated from ____ [ ]

A) Leaves to root B)From top to bottom

C)Both A & B D)None

26. ____________is an attribute whose value at a node in a parse tree is defined in terms of attribute at the parent and/or sibling of that node. [ ]

A) L-attribute B)S-attribute C) Synthesized D) Inherited

27. An attribute grammar in which all attributes are____ then it is called S attributed grammar.[ ]

A) Parsed B)Inherited C) A-attributed D) synthesized

28. An attribute grammar in which all the attributes are synthesized is called____Attributed grammar. [ ]

A) P B)Q C) R D) S

29. A parse tree, showing the value(s) of its attribute(s) is called_____________ [ ]

A) Annotated Parse tree B) parse tree

C) Syntax tree D) derivation tree

30. In which attribute to evaluate attributes in any bottom-up order, such as that of a postorder traversal of the parse tree. [ ]

A) synthesized attribute B)Inherite attribute

C) S-attibute D) L-attibute

31. Which are useful tool for determining an evaluation order for the attribute instances in a given parse tree? [ ]

A) Dependency Graph B) syntax tree

C) parser generator D) Scanner Generator

32. which one depicts the flow of information among the attribute instances in a particular parse tree? [ ]

A)parse tree B)derivation tree

C) Annonated parse tree D) Dependency Graph

33. A Type name is______ [ ]

A)Type expression B)Type Checking C) Backpatching D) None

34. Makelist(i) is a function of______ [ ]

A)Type expression B)Type Checking C) Backpatching D) Backtraking

35. Which of the following one is a context free grammar with program fragments embedded within production bodies [ ]

A)Syntax directed translation scheme B) SDD

C) Syntax tree D) Type checking

36. Syntax directed translation scheme is desirable because ___ [ ]

A)It is based on the syntax

B)It is easy to modify

C)Its description is independent of any implementation

D)All of these

37. If Conversion from one type to another type is done automatically by the complier then, it is called__ [ ]

A) Implicit conversion B) Coercions

C) Both A & B D) None of these

38. Which of the following one builds up the type of an expression from the types of its subexpressions? [ ]

A) Type expressions B) Type synthesis C) Type checking D) Type Analysis

39. Which one is used to determines the type of a language construct from the way it is used [ ]

A) Type inference B) Type synthesis C) Type checking D) Type Analysis

40. Type checking is normally done during [ ]

A) Lexical Analyser B) Syntax directed translation

C) Syntax Analysis D) Code Optimization

UNIT-4

1.Which of the following one is the interface between front end and back end in a compiler? [ ]

A) Intermediate Code B)Syntax trees

C) Semantic Analyser D) Lexical Analyzer

2. In which of the following one is not a intermediate code representation? [ ]

A)Postfix notation B)Syntax trees

C)Three address code D)Allocation of graph

3. Intermediate code generation phase gets input from which phase in compiler design? [ ]

A)Lexical analyser B) Syntax analyser

C)Semantic analyser D)Error Handling

4.General Form of a three-address statement is ________________ [ ]

A)a:=b (op) c B) a:=b c C) a:=b D) B only

5.Which of the following is an intermediate code form? [ ]

A) Three address code B) syntax tree

C) parser D) derivation tree

6. In quadruple notation maximum how many fieldsare used to represent operands. [ ]

A) 1 B)2 C) 3 D) 4

7. Generation of intermediate code based on a abstract machine model is useful in compilers because [ ]

A)it makes implementation of lexical analysis and syntax analysis easier

B)syntax directed translation can be written for intermediate code generation.

C)It enhances the portability of the front end of the compiler

D)it is not possible to generate code for real machines directly from high level language programs

8. Which of the following are the low level representation of machine dependant taks. [ ]

A) Trees B)LR parsing C) Ambiguous D) Syntax Trees

9. In which of the following one is not considered as a machine dependant tasks. [ ]

A) Register allocation B)Instruction Selection C) Both A & B D) target program

10. DAG stands for___________________ [ ]

A) Directed Acyclic graph B)Distributed Acyclic Graph

C) Directed Associated Graph D) Direct Acyclic Graph

11. Which one is used to eliminate the common sub expression elimination in the source

code? [ ]

A) Syntax tree B) Parse Trees C) DAG D) both A & B

13. In how many ways the 3-address code can be implemented in compiler design? [ ]

A) 3 B)2 C) 4 D) 5

14. Which of the following one is not considered a 3-address code implementation techniques? [ ]

A) Quadraples B)Triples C) Indirect triples D) Target program

15. Which technique is used to implement the control flow statements in one pass? [ ]

A) Register allocation B)Instruction Selection

C) Backpatching D) B only

16. To manipulate the list of jumps in control flow statements how many functions are used? [ ]

A) 3 B)4 C) 2 D) 1

17. Which of the following function is used to concatenates the lists pointed to by p1 and p2? [ ]

A)makelist(p1,p2) B) mergelist(p1,p2)

C) backpatch(p,i) D) map(p1p2)

18. which function is used to inserts i as target lable for each of the instructions on the list pointed to by p [ ]

A)makelist(p1,p2) B) mergelist(p1,p2)

C) backpatch(p,i) D) map(p1p2)

19. In which one a function call is unravelled into the evaluation of parameters in preparation for a call followed by the call itself. [ ]

A)Intermediate code B) 3-address code

C) syntax tree D) Target program instruction

20. which of the following one is not deal with Runtime Environment? [ ]

A)layout and allocation of storage B) Access to variable and data

C) Linkage between procedures D) parsing techniques

21. Space left unused due to alignment considerations is referred as__________ [ ]

A)Alignment B) Padding

C) storage allocation D) mapping

22. To check whether a variable is exactly defined once or not is a ________ check. [ ]

A) Uniqueness check B) Flow of Control Check

C) name check D) Above all

23. The storage strategy in which activation record is maintained even after the execution of a procedure is completed. [ ]

A) Stack Allocation B) Heap Allocation

C) Static Allocation D) Dynamic Allocation

24. How many storage allocation strategies are available in runtime environment? [ ]

A) 4 B) 2 C) 5 D) 3

25. Which of the following one is not storage allocation strategy? [ ]

A) Stack Allocation B) Heap Allocation

C) Static Allocation D) Dynamic Allocation

26. In which technique the operating system is maps the logical addresses into physical addresses, which are usually spread throughout memory? [ ]

A) Storage Organization B) Heap Allocation

C) Static Allocation D) Dynamic Allocation

27. ___ is a Data Structure, which is used by compiler to keep track of information [ ]

A) Lexical analyser B) Symbol Table

C) Semantic Table D) Semantic Analyzer

28.A symbol is said to be ____if it has different meaning depending on its context or use. [ ]

A) Override B)Overloaded

C) Overwrite D) overlapping

29. Which of the following can be visualized as a set of records in data structure? [ ]

A)Symbol Table B) Variable

C) both A & B D) none

30. A hash function should produce the same hash value for two different keys then it is called ______ [ ]

A) collisions B) Heap allocation

C) Stack allocation D) static allocation

31. Set of information constitute a record in dynamic allocation is called _____ [ ]

A)Activation Record (AR) B)frame.

C) both A & B D) Record

32. Procedure calls and return are usually managed by a runtime stack is called___ [ ]

A) Control stack B) Access link

C) Actual parameters D) Local Data

33. Which of the following one is used to stores the information of data which is outside the

local space? [ ]

A) Control stack B) Access link

C) Actual parameters D) Local Data

34. How many attributes are considered for the implementation of Activation Record? [ ]

A) 7 B) 4 C) 5 D) 6

35. Properties of memory managers are____________ [ ]

A) Space efficiency B) Program efficiency

C) low overhead D) all of the above

36. which of the following are operations performed on symbol table? [ ]

A) allocate ,insert, lookup , set attribute B) get attribute

C) allocate and free only D) both A & B

37. which of the following is not implementation techniques for symbol table? [ ]

A) Order linear list B) Binary Search Tree

C) Hash Table D) Static allocation

38. Which one of the following are the attributes of symbol table? [ ]

A) scope B)Data types

C) names D)all above

39. One of the purposes of using intermediate code in compilers is to [ ]

A) make parsing and semantic analysis simpler.

B) improve error recovery and error reporting

C) increase the chances of reusing the machine-independent code optimizer in other compliers.

D) improve the register allocation

40. Pick the machine independent phase of the compiler [ ]

A)Syntax analysis B) Lexical analysis

C) Intermediate code generation D) all of the above

UNIT-5

1.______is the process of transformation of code to an efficient code [ ]

A) Compiler B) Generator

C) Optimization D) parse tree

2. Determining Common sub expression can be done using____ [ ]

A) Compiler B) Interpreter

C) DAG D) parse tree

3. In DAG the interior nodes are labeled by ______ symbol [ ]

A) Operands B) operator

C) both A & B D) none of these.

4. Graphical representations are [ ]

A) triples B) DAG

C) postfix notations D) quadruples

5. Local and loop optimization in turn provide motivation for_____ [ ]

A)Data flow analysis B)Constant folding

C)Peephole optimization D)DFA and Constant folding

6. The process of Move the code from inner side to outer sideis known as_____ [ ]

A) Code motion B) Constant folding

C) Copy propagation D) none

7. Output of code generator phase in compiler design is ____ [ ]

A) Source code B)Intermediate code

C)Assembly code D)None of these

8. Which of the following is a simple, systematic technique for allocating registers and managing register spills. [ ]

A) DAG B) Graph coloring

C) A & B D) none

9. One approach to ______is to assign specific values in the target program to certain registers. [ ]

A) register allocation B) register assignment

C) register allocation and assignment D) none

10. The transformations that are applied across the basic blocks is called as____ [ ]

A) Global Optimization B) Local Optimization

C) Block Optimization D) none

11. Acronym for DAG ____ [ ]

A) Directed Acyclic Graph B)Direct Cyclic Graph

C)Derived Acyclic Graph D)Deviated Acyclic Graph

12. The first statement in basic block is________ [ ]

A) Main Statement B)Follow

C)Header D)Leader

13. Any statement that is immediately follows a conditional or unconditional statement is a leader statement. [ ]

A) Leader B)Follow

C)Header D)Statement

14. __________ give a pictorial representation of how values are computed at one statement. [ ]

A) Compiler B) DAG

C) Interpreter D) parse tree

15. The optimization which avoids test at every iteration is____ [ ]

A)Loop Unrolling B)Loop jamming

C)Constant folding D)None of these

16. The optimization technique which is typically applied on loops is__ [ ]

A) Peephole optimization B) Removal of invariant computation

C)Constant folding D)All of these

17. Which of the following is optimization technique used to optimize the code? [ ]

A) Dead code elimination B)Common Subprograms

C) Copy intermediate loop D) Loop Declaration

18. Live variables are used in_____ elimination [ ]

A)Common sub Expression B)Copy Propagation

C)Code Motion D)Dead code

19. DAG is constructed from ______ [ ]

A)3 address code B)program

C)blocks D)none

20. A Symbol table is _____ [ ]

A)Data structure B)Variable

C)Data Type D)None

21. Input to the code generator is _____ [ ]

A)Source code B)Intermediate code

C)Target code D)All of the above

22. __ is the portion of the program which will not be executed in any path of the program. [ ]

A) Live code B) Dead Code

C) reachable Code D) none of these

23. Which one is an estimate of how frequently a variable used in basic block? [ ]

A) Usage count B)Reference count

C) Program count D)Process count

24. A flow graph is a directed graph in which the flow of control information is added to_____ [ ]

A)blocks B)graph

C)tree D)basic blocks

25. Basic block is Sequence of ______ [ ]

A)Statements B)Loops

C)Values D)None

26. Graph coloring is strategies of _______ [ ]

A)Register allocation B)Heap allocation

C) Stack allocation D)None

27. DAG representation of a basic block allows [ ]

A) Automatic detection of local common sub expressions

B) Automatic detection of induction variables

C) Automatic detection of loop variant

D) None of the above

28. Code generation phase converts the __into a sequence of machine instruction [ ]

A) Intermediate optimized code B) assembly code

C) target code D) none of these

29. If the value of the variable is changed every time then that variable is called as [ ]

A)invarient variable B)Dead variable

C)Live variable D)Induction variable

30. The strength reduction is related to _______ [ ]

A)variables B)Loops

C)operators D)All

31. DAG representation of a basic block allows [ ]

A) Automatic detection of local common sub expressions

B) Automatic detection of induction variables

C) Automatic detection of loop variant

D) None of the above

32. Some code optimizations are carried out on the intermediate code because (GATE 2008) [ ]

A) They enhance the portability of the compiler to other target processors

B) Program analysis is more accurate on intermediate code than on machine code

C) The information from dataflow analysis cannot otherwise be used for optimization

D) The information from the front end cannot otherwise be used for optimization

33. Consider the following statements. GATE 2020 [ ]

I. Symbol table is accessed only during lexical analysis and syntax analysis.

II. Compilers for programming languages that support recursion necessarily need heap storage for memory allocation in the run-time environment.

III. Errors violating the condition ‘any variable must be declared before its use’ are detected during syntax analysis.

Which of the above statements is/are TRUE?

A) I and III only B) II only C) I only D) None of the above

34. Which languages necessarily need heap allocation in the runtime environment? GATE 2000[ ]

A) Those that support recursion

B) Those that use dynamic scoping

C) Those that allow dynamic data structure

D) Those that use global variables

35. Optimization techniques can be applied to ______ [ ]

A) intermediate code B) final target code C)Both A and B D)None

36. A graphical representation of three address code is called ________ [ ]

A)blocks B)flow graph C)tree D)basic blocks

37. Any statement of conditional or unconditional statement is _______ [ ]

A) Leader B)Follow C)Header D)Statement

38. A transformation of a program is called ______ If it is applied with in the basic block.

[ ]

A) Live B)Follow C)Local D)Global

39. How many types of transformations in Principle source of optimization. [ ]

A) 2 B)4 C)3 D)5

40. ______transformations that are performed without changing the function it computes. [ ]

A) Function Preserving transformations

B) Structure- preserving transformations

C) Algebraic- transformations

D) All the above

Prepared by: R. Priyadarshini/Assoc. Prof/SIST /CSE