Shivaji University , Kolhapur
Question Bank For Mar 2022 (Summer) Examination
Subject Code :81546 Subject Name : Compiler Construction

QN	QUESTION TEXT	OPTION1	OPTION2	OPTION3	OPTION4	ANS
1	Which of the following is not a tool used in compiler?	Parser Generator	Syntax Directed Translation Engine	Scanner Generator	Static Checker	D
2	Optimizing Compiler _____.	Optimized to occupy less space	Optimize the code	Take less time to execute	None of these	B
3	Silicon compilers are used in circuit design.	TRUE	FALSE			A
4	_____or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.	Lexical analysis	Diversion	Modeling	None of these	A
5	The lexical analyzer takes_____as input and produces a stream of_____as output.	Source program,tokens	All of these	Token,source program	None of these	A
6	YACC builds up _____.	SLR parsing table	Canonical LR parsing table	LALR parsing table	None of these	C
7	The linker _____.	is similar to interpreter	uses source code as its input	is required to create a load module	none of the above	C
8	For specifying token _____ is used.	Regular language	Finite Automata	Regular Expression	PDA	C
9	Which of the following is not a token of C program?	1.02E+02	#define	MAX	123.33	B
10	For reading input character by character, buffering mechanism is used.	TRUE	FALSE			A
11	Compiler is a program that _____.	places programs into memory and prepares them for execution	automates the translation of assembly language into machine language	accepts a program written in a high level language and produces an object program	appears to execute a source program as if it were machine language	C
12	For predictive parsing the grammar A->AA I (A) I ε is not suitable because	The grammar is right recursive	The grammar is left recursive	The grammar is ambiguous	The grammar is an operator grammar	B
13	In a compiler, the data structure responsible for the management of information about variables and their attributes is _____.	Semantic stack	Parser table	Symbol table	Abstract syntax-tree	C
14	Shift reduce parsers are _____.	Top down parser	Bottom up parser	May be top down or bottom up parser	None of the above	B
15	A grammar that produces more than one parse tree for leftmost or rightmost derivation of some sentence is called _____.	Ambiguous	Unambiguous	Regular	None of these	A
16	In a compiler _____ checks every character of the source text.	The lexical analyzer	syntax analyzer	code generator	code optimizer	A
17	Advantage of panic mode of error recovery is that _____.	it is simple to implement	it never gets into an infinite loop	both (a) and (b)	none of these	C
18	Synthesized attribute can be easily simulated by a _____.	LL grammar	Ambiguous grammar	LR grammar	None of the above	C

QUESTION BANK

Unit 1: Introduction to Compiling

1. Define Compilers. Explain phases of a compiler
2. Explain different following tools for which compiler technology is used to create:
   a. Structure editors
   b. Pretty Printers
   c. Static Checkers
   d. Interpreters
   e. Silicon Compilers
3. Explain different compiler construction tools.
4. Explain following cousins of compiler:
   a. Linker
   b. Loader
   c. Assembler
5. What are different types of compiler? Explain.
6. What is pass in Compilers? What is meant by analysis & synthesis phase of a compiler?
7. Explain translation of a statement using 6 phases of compiler.
8. What is difference between pass 1 & pass 2 compilers?

Unit 2: Lexical Analysis

1. Explain structure of Lexical Analyzer.
2. How error recovery is handled in lexical analysis?
3. What are tokens? Explain specification & recognition of tokens.
4. Create a lexical analyzer with LEX as tool.
5. Explain Lex specification.
6. Design a lexical analyzer generator.
7. What is input buffering? Explain with example.
8. Construct Transition Diagram for relational operator (relop)
9. Generate string for the following languages.
   i) aa(a/b)*bb
   ii) aa*(a/b)*bb*
   iii) (aa)*abb(bb)*
10. Construct Finite Automata for following Language

Unit 4: Syntax Directed Translation and Intermediate Code Generation

1. What is SDD?
2. Differentiate between synthesized & inherited attributes.
3. What is S attributed definition & L attributed definition? Explain with examples.
4. Construct SDD syntax tree for expressions.
5. What is SDT? Explain top down translation for L attributed definition.
6. What is intermediate code generation?
7. Explain following:
   a. Syntax tree
   b. Postfix notation
   c. Three address code
8. Differentiate between parse tree & syntax tree.
9. What is backpatching?
10. What is Annotated Parse Tree? Construct Annotated Parse Tree for 3*5+4 using S-Attributed Definition.
11. Write Syntax Directed Translation Scheme for Assignment Statements.
12. Write Syntax Directed Translation Scheme for Boolean Expression.

Unit 5: Code Optimization

1. What are sources of optimization?
2. What is peephole optimization?
3. What are basic blocks?
4. What are loops in flow graphs?

Unit 6: Code Generation

1. What is code generation? Explain basic blocks & flow graphs?
2. Explain three address codes with example
3. Explain quadruples & triples with examples.
4. What are issues in design of a code generator?
5. Explain Run time storage management & next use information.
6. Explain issues in register allocation.
7. Explain code generation form DAGs.
8. Construct DAG(Directed Acyclic Graph) for following Expression
   ((a*b)+(a*b))+((c*d)+(c*d))

i) (a/b)*abb
ii) aa*/bb*

Unit 3: Syntax Analysis

1. What is syntax analysis (parsing)? Explain structure of a parser.
2. Explain syntax errors & their recovery with the help of compilers.
3. Explain following error recovery strategies:
   a. Panic Mode b. Phrase Level c. Error Productions d. Global Corrections
4. What is ambiguity in grammar? Explain with example.
5. What is precedence & associativity of operators? Explain with example.
6. What is top down parsing? Explain with example.
7. What are recursive descent parsers? Explain with example.
8. What is backtracking? Explain with example.
9. How to design non backtracking recursive descent parsers?
10. Explain non recursive predictive parsing algorithm.
11. Explain LL (1) parsing algorithm.
12. What is operator precedence parsing? Explain with example.
13. Explain SLR parsing.
14. Explain LALR parsing.
15. How to calculate FIRST & FOLLOW sets?
16. What is Left Recursion? Perform Left recursion on following grammar
    E→E+T/T
    T→T*F/F
    F→(E)/id
17. Check the below grammar is in LL(1) or not
    S→AB
    A→a
    B→b
18. What is Shift-Reduce Parsing Technique? Explain with Example.
19. Check the below grammar is in LR(0) or not
    E→T+E
    E→T
    T→i
20. What are the conditions to check SR and RR Conflict in SLR (1) Parsing Technique?

19	Which one of the following statement is false for the SLR (1) and LALR (1) parsing tables for a context free grammar?	The reduce entries in both the tables may be different	The error entries in both the tables may be different	The go to part of both tables may be different	The shift entries in both the tables may be identical	C
20	A bottom up parser generates _____.	Right most derivations	Right most derivations in reverse	Leftmost derivations	Leftmost derivations in reverse	B
21	Which table is a permanent data bases that has an entry for each terminal symbol?	Terminal table	Literal table	Identifier table	Reductions	A
22	_____ is a top-down parser	Operator precedence parser	An LALR (k) parser	An LR (k) parser	Recursive descent parser	D
23	In a compiler, when is the keywoards of a language are recognized?	During the lexical analysis of a program	During parsing of the program	During the code generation	During the data flow analysis	A
24	Which of the following is not an intermediate code form?	Postfix notation	Syntax trees	Three address codes	Quadruples	D
25	To convert an arbitrary CFG to an LL (1) grammar	Left Factor the grammar alone	Remove left recursion alone	Remove left recursion and left factor the grammar	None of these	C
26	Common subexpression elimination is not the code optimization technique.	TRUE	FALSE			B
27	Which of the following system software resides in main memory?	text editor	assembler	linker	all of these	D
28	_____ replaces an expensive operation by a cheaper one, such as a multiplication by an addition.	Code motion	Compaction	Reduction in strength	Induction	C

29	Peephole optimization is a form of _____.	loop optimization	local optimization	constant folding	none of these	B
30	Task of the lexical analysis phase is _____.	to parse the source program into basic elements or tokens of the language	to build a literal able and an identifier table	to build a uniform symbol table	all of these	D
31	In flow graphs, nodes represent computations.	TRUE	FALSE			A
32	In flow graphs, edges represent the flow of control.	TRUE	FALSE			A
33	The output of a code generator is a _____ .	syntax tree	target program	parse tree	source program	B
34	Comments in the source program to a computer are deleted by _____.	the lexical analysis	the semantic analysis	the code generation	the code optimizer	A
35	The quality of generated code is determined by its _____ .	behavior & size	behavior & speed	speed & size	behavior only	C