


SIDDHARTH GROUP OF INSTITUTIONS :: PUTTUR

Siddharth Nagar, Narayanavanam Road – 517583

QUESTION BANK (DESCRIPTIVE)
Subject with Code : Compiler Design (16CS524)

Course & Branch : B. Tech - CSE

Year & Sem : III B.Tech& II-Sem

Regulation : R16

UNIT-1

1. How many phases are there in a compiler? Explain each phase in detail? [L2, 10M]
- 2.a) Describe the role of lexical analysis in compiler design? [L1, 5M]
 - b) Explain Input Buffering with simple examples? [L2, 5M]
3. Explain about Language Processor in compiler Design? [L2, 10M]
4. Explain the following terms
 - a) Specification of Tokens [L2, 4+6M]
 - b) Recognition of Tokens
- 5.a) Explain the Structure of Compiler? [L2, 5M]
 - b) What is the need for separating lexical analysis and syntax analysis? [L2, 5M]
6. Explain LEX Tool with an Lex Programme? [L1, 10M]
7. Write short notes [L6, 5+5M]
 - a) pass and phases of a compiler
 - b) Bootstrapping
8. How to design the compiler by using the source program position: $\text{initial} + \text{rate} * 60$. [L2, 10M]
9. Write short notes [L6, 5+5M]
 - a) Application of compiler technology
 - b) Specification of Tokens
- 10
 - a) List the various phases of a compiler. [L1, 2M]
 - b) Differentiate tokens, patterns, and lexeme. [L4, 2M]
 - c) Differences between compiler and Interpreter. [L1, 2M]
 - d) Define Regular Expressions and Regular Grammar. [L1, 2M]
 - e) List the various error recovery strategies for a lexical analysis. [L1, 2M]

UNIT-2

- 1.a) Construct the recursive decent parser for the following grammar? [L4, 5M]
 $E \rightarrow E+T/T$
 $T \rightarrow T^*F/F$
 $F \rightarrow (E)/id$
- b) Explain about Left factoring and Left Recursion with an examples? [L2, 5M]
2. Define augmented grammar? Construct the LR(0) items for the following Grammar? [L1, 10M]
 $S \rightarrow L=R$
 $S \rightarrow R$
 $L \rightarrow *R$
 $L \rightarrow id$
 $R \rightarrow L$
3. Calculate FIRST and FOLLOW for the following grammar? [L3, 5M]
 a) $E \rightarrow E+T/T$
 $T \rightarrow T^*F/F$
 $F \rightarrow (E)/id$
 b) $S \rightarrow xABC$ [L3, 5M]
 $A \rightarrow a|bbD$
 $B \rightarrow a|\epsilon$
 $C \rightarrow b|\epsilon$
 $D \rightarrow c|\epsilon$
4. Construct Predictive Parse Table for the grammar $E \rightarrow E+T/T, T \rightarrow T^*F/F, F \rightarrow (E)|id$ and parse the string $id+id*id$. [L3, 5M]
5. Perform Shift Reduce Parsing for the following
 i) $S \rightarrow (L)|a$
 $L \rightarrow L,S|S$ input string: $(a,(a,a))$
 ii) $E \rightarrow E+E / E^*E / (E) / id$ input string $(id*id+id)$ [L3, 5M]
6. Construct CLR Parsing table for the given grammar [L3, 10M]
 $S \rightarrow CC$
 $C \rightarrow aC/d$
7. Consider the grammar [L3, 10M]
 $S \rightarrow AB|ABad$
 $A \rightarrow d$
 $E \rightarrow b$
 $D \rightarrow b|\epsilon$
 $B \rightarrow c$

Construct the predictive parsing table. Show that the given grammar is LL(1) or not

8. Consider the grammar $S \rightarrow xABC$
 $A \rightarrow a|bbD$
 $B \rightarrow a|\epsilon$
 $C \rightarrow b|\epsilon$
 $D \rightarrow c|\epsilon$
 Construct predictive parsing table for the given grammar. [L3, 10M]
9. Perform Shift Reduce Parsing for the input string using the grammar. [L4, 5+5M]
 $S \rightarrow (L)a$
 $L \rightarrow L,S|S$
 a) (a,(a,a))
 b) (a,a)
- 10 a) Define LL(1)? [L1, 2M]
 b) Differences between SLR,CLR, LALR parsers? [L1, 2M]
 c) Problems in Top Down Parsing? [L1, 2M]
 d) Define Handle prunig?. [L1, 2M]
 e) Define Ambiguous grammar? [L2, 2M]

UNIT-3

1. Explain syntax directed definition with simple examples? [L2, 10M]
2. Describe the evaluation order of SDT with an example. [L5, 10M]
3. Explain the Type Checking with suitable examples? [L2, 10M]
4. Explain the Translation scheme of SDD. [L2, 10M]
5. Describe the representation of 3-address code with an examples. [L5, 10M]
6. Explain in detail about Backpatching Technique?. [L2, 10M]
7. Explain the applications of Syntax Directed Definition. [L2, 10M]
8. Write down the translation procedure for control statement? [L6, 10M]
9. Explain different types of intermediate code representations? [L2, 10M]
10. a) Define a syntax-directed translation. [L1, 2M]
 b) Define annotated parse tree. [L1, 2M]
 c) What are the three functions of backpatching? [L1, 2M]
 d) Write the Syntax of case statement?. [L6, 2M]
 e) Differentiate between L attribute and S attribute. [L4, 2M]

UNIT-4

1. Draw the format of Activation Record in stack allocation and explain each field in it. [L4, 10M]
2. Explain about Global data flow analysis. [L2, 10M]
3. Explain the Storage Organization with simple examples. [L2, 10M]
4. Define Symbol table. Explain different types of Data structure for symbol table [L1, 10M]
5. Distinguish between static scope and dynamic scope. Briefly explain access to non-local Names in static scope. [L4, 10M]

- | | |
|--|-----------|
| 6. Explain the Non Block Structured Languages? | [L2, 10M] |
| 7. Explain Storage allocation strategies with suitable examples? | [L2, 10M] |
| 8. Explain heap management mechanism. | [L2, 10M] |
| 9. Explain about block structured language. | [L6, 10M] |
| 10 a) Define Activation Record. | [L6, 2M] |
| b) Name any four procedural optimization techniques | [L6, 2M] |
| c) Define scope and life time of variable. | [L1, 2M] |
| d) Define symbol table. | [L1, 2M] |
| e) Define data flow equation?. | [L1, 2M] |

UNIT-5

- | | |
|--|-----------|
| 1. Write about all issues in code generation. Describe it. | [L6, 10M] |
| 2. Explain the target machine architecture? | [L2, 10M] |
| 3. Explain optimization techniques on Basic Blocks with simple examples? | [L2, 10M] |
| 4. Describe the various strategies in register allocation. | [L5, 10M] |
| 5. Explain the peephole optimization Technique?. | [L2, 10M] |
| 6. Construct the DAG for following statement. $a+b*c+d+b*c$ | [L3, 10M] |
| 7. Construct the DAG for the following basic blocks | [L3, 10M] |

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. $i:=t7$
10. if $i \leq 20$ goto 1

- | | |
|---|------------|
| 8. Explain the simple code generator and generate target code sequence for the following statement $d:=(a-b)+(a-c)+(a-c)$ | [L2, 10M] |
| 9. Write short notes on i) Simple code generator
ii) Register allocation | [L6, 5+5M] |
| 10. Explain the following terms | |
| a) Role of peephole optimization in compilation process | [L1, 2M] |
| b) Issues in the design of a code generator.(any 4) | [L6, 2M] |
| c) Give the different forms in target program | [L1, 2M] |
| d) Give the applications of DAG. | [L1, 2M] |
| e) Define Dead-code elimination with example. | [L1, 2M] |