B.TECH/IT/5TH SEM/INFO 3132(BACKLOG)/2021

COMPILER DESIGN
(INFO 3132)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) A system program that combines separately compiled modules of a program into a form suitable for execution is
(a) Assembler (b) Linking Loader
(c) Cross Compiler (d) None of the mentioned

(ii) The graph that shows basic blocks and their successor relationship is called
(a) Dag (b) Flow Graph
(c) Control Graph (d) Hamilton Graph

(iii) Which of the following derivations does a top-down parser use while parsing an input string?
(a) Leftmost derivation (b) Leftmost derivation in reverse
(c) Rightmost derivation (d) Rightmost derivation in reverse

(iv) $(a + b)^*$=?
(a) $(a^* + b^*)^*$ (b) a* + b*
(c) (a*b*)* (d) both (a) and (c)

(v) 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

(vi) Consider the grammar G whose SLR parser has n1 states and LALR parser has n2 states. What is the relation between n1 and n2?
(a) n1 = n2 (b) n1 < n2
(c) n1 > n2 (d) None of the above

(vii) For predictive parsing the grammar A→AA I (A) I ε is not suitable because
(a) The grammar is right recursive
(b) The grammar is left recursive
(c) The grammar is ambiguous
(d) The grammar is an operator grammar

(viii) Which of the following is the most powerful parser?
(a) SLR (b) LALR
(c) Canonical LR (d) Operator precedence

(ix) In a compiler, the data structure responsible for the management of information about variables and their attributes is
(a) Semantic stack (b) Parser table
(c) Symbol table (d) Abstract syntax-tree

(x) The languages that need heap allocation in the runtime environment are
(a) Those that use global variables
(b) Those that use dynamic scoping
(c) Those that support recursion
(d) Those that allow dynamic data structure

Group- B

2. (a) What is the role of Lexical Analyzer in compilation process? What are lexemes and tokens? [(CO1) (Remember/LOCQ)]
(b) Using Thompson's construction rule convert regular expression ((a|b)*b*)ab to NFA and then convert the NFA of ((a|b)*b*)ab to DFA. [(CO2) (Apply/IOCQ)]

(1 + 1 + 1) + (4 + 5) = 12

3. (a) Write the regular expression over alphabet {0, 1} for the set of strings with odd number of 0's followed by even number of 1's. Convert this regular expression into an equivalent NFA using Thompson's Construction rules. [(CO2) (Apply/IOCQ)]
(b) Explain the use of sentinels to recognize tokens. [(CO1) (Understand/LOCQ)]

(3 + 5) + 4 = 12

Group - C

4. (a) What are Left Recursion and Left Factoring? Why should a grammar be free from these? [(CO2) (Understand/LOCQ)]
(b) Explain Handle Pruning with an example. [(CO3) (Apply/IOCQ)]
(c) Check whether the following Grammar is LL(1) Grammar or not
S→xEySS'|a
S'→bS| Є
E→c. [(CO6) (Evaluate/HOCQ)]

(1 + 1 + 2) + 3 + 5 = 12

5. (a) Define augmented grammar with example. Draw parse tree for the following code fragment:

```
if(basic ≤ 15,000)
        salary =basic*2.8;
else
        salary= basic*3.1 + PF;
```

[(CO3) (Evaluate/HOCQ)]

(b) What is Syntax-Directed Definition? Explain the form of a Syntax-Directed Definition. [(CO7) (Understand/LOCQ)]

(3 + 5) + (1 + 3) = 12

Group - D

6. (a) Explain the different fields of activation records at run-time. [(CO4) (Apply/IOCQ)]

(b) Translate the expression x= - (a-b)*(c+d)+(a-b+c) into

(i) Quadruples
(ii) Triples
(iii) Indirect triples. [(CO4) (Apply/IOCQ)]

6 + (3 × 2) = 12

7. (a) Explain code motion with an example. [(CO1) (Understand/LOCQ)]

(b) Consider the following statements to construct 3-address code with quadruples:

```
x:= y+z+k;
if  x ≤ 10 then
        x:=y*z*k;
else
        x:=y+z-k;
```

[(CO4) (Apply/IOCQ)]

6 + 6 = 12

Group - E

8. Consider the following c code to find the sum of digits of a number

```
int n,rem,sum=0;
Read n;
while(n!=0)
{
        rem=n%10;
         sum=sum+rem;
        n=n/10;
}
```

(i) Translate the above program into three-address statements
(ii) Construct a flow graph from the three-address statements
(iii) Show the dominator tree for the flow graph. [(CO4) (Evaluate/HOCQ)]

(3 × 4) = 12

9. (a) Explain different structure-preserving transformation technique with an example. [(CO4) (Apply/IOCQ)]
   (b) Explain PEEPHOLE optimization technique. [(CO1)(Remember/LOCQ)]

6 + 6 = 12

Cognition Level	LOCQ	IOCQ	HOCQ
Percentage distribution	28.12%	45.83%	26.05%

Course Outcome (CO):

After the completion of the course students will be able to

1 Describe the theory and practice of compilation, in particular the lexical analysis, syntax, and semantic analysis, code generation and optimization phases of compilation.
2 Create lexical rules and grammars for a programming language.
3 Use Flex or similar tools to create a lexical analyzer and Yacc/Bison tools to create a parser.
4 Design a compiler for a concise programming language.
5 Implement a lexer without using Flex or any other lexer generation tools.
6 Implement a parser such as a bottom-up SLR parser without using Yacc/Bison or any other compiler generation tools.
7 Implement semantic rules into a parser that performs attribution while parsing.

*LOCQ: Lower Order Cognitive Question; IOCQ: Intermediate Order Cognitive Question; HOCQ: Higher Order Cognitive Question

Department & Section	Submission Link
IT	Classroom link https://classroom.google.com/c/MjI3ODg5NzkxMjA5/a/MjI3ODg5ODc0NzEw/details