

09-06-19

Total No. of printed pages = 4

CS 131802

BINA CHOWDHURY CENTRAL LIBRARY
(GIMT & GIPS)

Azara, Hatkhowapara,

Guwahati - 781017

Roll No. of candidate

--	--	--	--	--	--	--	--	--	--

2019

B.Tech. (CSE) 8th Semester End Term Examination

COMPILER DESIGN

(Old Regulation)

Full Marks – 100

Time – Three hours

The figures in the margin indicate full marks
for the questions.

Answer Question No.1 and any Six from the rest.

1. Answer the following questions: (1 × 10 = 10)

- (i) A bottom up parser generates
 - (a) Right most derivation
 - (b) Rightmost derivation in reverse
 - (c) Leftmost derivation
 - (d) Leftmost derivation in reverse
- (ii) What is YACC?
- (iii) _____ is a top-down parser
 - (a) Operator precedence parser
 - (b) An LALR (k) parser
 - (c) An LR (k) parser
 - (d) Recursive descent parser

[Turn over

- (iv) A grammar that produces more than one parse tree for some sentence is called
- (a) Ambiguous
 - (b) Unambiguous
 - (c) Regular
 - (d) None of the mentioned
- (v) What is Semantic analysis?
- (vi) To convert an arbitrary CFG to an LL(1) grammar
- (a) Only factor the grammar alone
 - (b) Only remove left recursion alone
 - (c) Remove left recursion as well as factor the grammar
 - (d) None of the above
- (vii) In a compiler, _____ analyzer checks every character of the source text.
- (viii) Shift reduce parsers are
- (a) Top down Parser
 - (b) Bottom Up parser
 - (c) May be top down or bottom up
 - (d) None of the mentioned
- (ix) Which of the following is a phase of a compilation process?
- (a) Lexical analysis
 - (b) Code generation
 - (c) Syntax analysis
 - (d) All of the above
- (x) Define 'Handle Pruning' in bottom-up parsing.
2. (a) (i) Using a suitable example, explain in detail the phases of compiler. (10+5=15)
- (ii) Consider the following program and write down the lexemes, tokens and attributes:
- ```
main()
{
int a,b,c;
c=a+b;
}
```

- (b) (i) Compute the FIRST and FOLLOW of the following CFG: (5+5+5=15)  
 $A \rightarrow Ac \mid Aad \mid bd \mid \varepsilon$
- (ii)  $S \rightarrow () \mid a \mid (A)$   
 $A \rightarrow S \mid A + S$
- (iii)  $S \rightarrow Bc \mid DB$   
 $B \rightarrow ab \mid cS$   
 $D \rightarrow d \mid \varepsilon$ .
- (c) (i) Construct a predictive parsing table for the grammar given below (7+8=15)  
 $E \rightarrow TE'$   
 $E' \rightarrow TE' \mid \varepsilon$   
 $T \rightarrow FT'$   
 $T' \rightarrow FT'$   
 $T' \rightarrow *FT' \mid \varepsilon$   
 $F \rightarrow (E) \mid id$
- (ii) What is LL(1) grammar? Also test whether the following grammar is LL(1) or not.  
 $S \rightarrow AaAb \mid BbBb$   
 $A \rightarrow \varepsilon$   
 $B \rightarrow \varepsilon$
- (d) (i) Justify that no LL(1) grammar can be ambiguous. (7+8=15)
- (ii) Check whether the following grammar is SLR(1) or not:  
 $X \rightarrow Y1 \mid 2Y3 \mid Z3 \mid 2Z1$   
 $Y \rightarrow 4$   
 $Z \rightarrow 4$
- (e) (i) State the advantage and disadvantage of LR parser. (4+3+8=15)
- (ii) What is the need of augmented grammar in LR parser.
- (iii) Show that the following grammar is CLR(1) but not LALR(1)  
 $S \rightarrow Aa \mid bAc \mid Bc \mid bBa$   
 $A \rightarrow d$   
 $B \rightarrow d$

BINA CHOWDHURY CENTRAL LIBRARY  
 (GIMT & GIPS)  
 Azara, Hatkhowapara,  
 Guwahati -781017

- (f) (i) Draw the syntax tree, DAG and three address code for the following expression:

$$(6+6+3=15)$$

$$a + a * (b - c) + (b - c) * d$$

- (ii) Translate the expression into quadruple, triple & indirect triple:

$$a = b * -c + b * -c$$

- (iii) What is short circuit code?

- (g) (i) Give Three Address Code for the following:

do  $i=i+1$ ; where  $(a[i] < v)$  (5+10=15)

- (ii) Generate the flow of the program fragment given below:

switch  $(i+j)$

{

case 1 :  $x = y + z$ ; break

case 2 :  $u = v + w$ ; break

default :  $p = q + r$ ;

}

- (h) Write short notes on (any five): (5 × 3 = 15)

- (i) Peephole Optimization.
- (ii) Symbol table management.
- (iii) Error recovery strategies in LR Parser.
- (iv) The role of flow graph in basic blocks.
- (v) Intermediate code generation.
- (vi) Thompson's construction algorithm.
- (vii) Syntax directed translation.