Date: 15/11/2017

Exam No: _____

GANPAT UNIVERSITY

M. TECH SEM-III(COMPUTER ENGINEERING/INFORMATION TECHNOLOGY)

REGULAR EXAMINATION NOV-DEC 2017

3CE308/3IT308: COMPILER DESIGN

MAX. TIME: 3 HRS **MAX. MARKS: 60**

Instructions: (1) This question paper has two sections. Attempt each section in separate answer book.
(2) Figures on right indicate marks.
(3) Be precise and to the point in answering the descriptive questions.

SECTION: I

Q.1

A Answer the following: **10**

1. What does the preprocessor do? Explain their role(s) in compilation process.
2. What is ambiguity? Show that S → xS | Sx | x is an ambiguous grammar.
3. Construct CFG for the language L = $\{a^n b^n / n \geq 1\}$.
4. Check whether the given grammar is LL (1) or not?
   A→ AaB | x
   B →BCb | Cy
   C→ Cc | ϵ
5. Define following terms:
   a. Linker b. Loader

OR

Q.1

A List the phases that constitute the front end of a compiler. For the statement "c = a + b * d / 2.6", what will be the output after completion of front end process of a compiler? **6**

B What is the importance to perform left factoring on CFG? Explain it with example. **4**

Q.2

A Find First ( ), Follow ( ), Select ( ) and construct M-Table for the following grammar and check whether it is LL (1) or not? **6**

S → AB
A → Ca | ϵ
B → baAC | c
C → b | ϵ

B Do as directed: **4**

1. Remove left recursion from the following grammar:
   S → Sx | SSb | xS | a
2. Consider the following grammar:
   S → (S) | a
   Let the number of item sets in SLR (1), CLR (1) and LALR (1) parsers for the grammar are n1, n2 and n3 respectively. Which one of the following is correct?
   a. n1< n2 < n3 b. n1 = n2 = n3
   c. n1 = n3 < n2 d. n1 ≥ n3 ≥ n2

OR

Q.2

A For the following grammar find First( ), Follow( ), Select( ), construct predictive parsing table and parse the string " ,abbn; " **6**

S → ,GH;
G → aF
F → bF | ϵ
H → KL
K → m | ϵ
L → n | ϵ

B Perform left factoring on following grammar: **4**

1. $E \rightarrow 5 + T \mid 3\text{-}T$
   $T \rightarrow Vax \mid Vaxy \mid Va \mid V$
   $V \rightarrow acd \mid ab$
2. $A \rightarrow aBcC \mid aBb \mid aB \mid a$
   $B \rightarrow \epsilon$
   $C \rightarrow \epsilon$

Q.3

A "Every left recursion grammar is not LL (1) but grammar without left recursion is always LL (1) " – true or false. Justify your answer with example. **6**

B What is operator grammar? Construct operator precedence parsing table for the following grammar and parse the string "a + a * a". **4**

$B \rightarrow B + B \mid B * B \mid a$

SECTION: II

Q.4

A Show that the following grammar is LALR (1) but not SLR (1): **10**

$S \rightarrow Ma \mid bMc \mid dc \mid bda$

$M \rightarrow d$

OR

Q.4

A Construct LR (0) parsing table for the following grammar and check whether it is LR (0) or not? **6**

$P \rightarrow bD;Se$

$D \rightarrow d;d$

$S \rightarrow s;s$

B Construct CLR(1) parsing table for the following grammar: **4**

$A \rightarrow (A) \mid a$

Q.5

A Explain S-R and R-R conflict of SLR (1) and CLR (1) with example. Why S-S conflict is not possible in all the parsing technique? Justify your answer with the help of example. **6**

B Do as directed: **4**

1. Apply loop splitting optimization technique on following code segment:

```
p = 10;
for (i=0; i<10; ++i)
{  y[i] = x[i] + x[p];
   p = i; }
```

2. Explain dead code elimination optimization technique with example.

OR

Q.5

A Consider the following grammar: **6**

$S \rightarrow axF$

$F \rightarrow ,JF \mid \epsilon$

$J \rightarrow ax \mid x$

Construct CLR(1) parsing table and parse the string "ax,ax"

B Define: Handle. Consider the following grammar and show the handle of each right sentential form for the string "id + num * id". **4**

$S \rightarrow E$

$E \rightarrow T + E \mid E * T \mid num$

$T \rightarrow id$

Q.6

A Answer the following: **5**

1. Which parsing technique is known as look-head LR?
2. Count number of tokens from the following program segment:

```
int x, y;
/* find max from x & y */
{   return (x > y ? x : y);  }
```

3. $A \rightarrow Aa \mid (A) \mid \epsilon$ is not LL (1) grammar because it is _____.
   a. Ambiguous b. Left-recursive c. Right-recursive d. Operator grammar
4. Write the condition of S-R conflicts for LALR (1) parser.
5. Identify the associativity of operators + and * from the following grammar:

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow id$

B List the name of error recovery techniques. Explain any one in detail. **3**

C Explain three address code with example. **2**

----------END OF PAPER----------