प्रज्ञानं ब्रह्म

## MANIPAL INSTITUTE OF TECHNOLOGY, MANIPAL 576104

Reg. No.

(Constituent College of Manipal University)

FIFTH SEMESTER B.Tech. (IT) DEGREE MAKE UP EXAMINATION, JANUARY 2016 SUBJECT: SYSTEMS PROGRAMMING - ICT 307 (REVISED CREDIT SYSTEM)

| TIME: 3 HOURS                | 04/01/2016                 | MAX. MARKS: 50 |
|------------------------------|----------------------------|----------------|
|                              | Instructions to candidates |                |
| • Answer any <b>FIVE FUL</b> | L questions.               |                |
| • Missing data, if any, ma   | y be suitably assumed.     |                |

1A. Consider the table shown in Table Q.1A and do the following.

- i. Construct dependency graph for the input : 110110
- ii. Draw the annotated parse tree for the input: 11111

| Productions          | Semantic Rule                    |
|----------------------|----------------------------------|
| $N \rightarrow S L$  | $L.pos \coloneqq 0$              |
|                      | $N.val \coloneqq S.neg * L.val$  |
| S→+                  | $S.neg \coloneqq 1$              |
| S→-                  | S.neg $\coloneqq$ -1             |
| $L \rightarrow L1 B$ | $L1.pos \coloneqq L.pos + 1$     |
|                      | $B.pos \coloneqq L.pos$          |
|                      | $L.val \coloneqq L1.val + B.val$ |
| $L \rightarrow B$    | $B.pos \coloneqq L.pos$          |
|                      | $L.val \coloneqq B.val$          |
| $B \rightarrow 0$    | $B.val \coloneqq 0$              |
| B→1                  | $B.val \coloneqq 2^{B.pos}$      |

## Table O.1A

- 1B. Differentiate between synthesized and inherited attributes.
- 1C. hat is the necessity of augmenting the grammar in LR parsers?
- 2A. Construct LALR parsing table for the following grammar also parse the given string.

| $S \rightarrow L=R \mid R$ |  |  |
|----------------------------|--|--|
| $L \rightarrow *R \mid id$ |  |  |
| $R \rightarrow L$          |  |  |
| String : id=*id            |  |  |

What is input buffering? Explain. 2B.

- 2C. Define and give an example for L-Attributed definitions.
- 3A. Consider the following grammar and do the following.
  - Construct the operator precedence table i.
  - ii. Construct the precedence functions table
  - iii. Show the movement of the parser for the input (a,a,a)  $S \rightarrow (L) \mid a$  $L \rightarrow L, S \mid S$
- 3B. Verify the following grammar is ambiguous or not by constructing the parse tree / s.  $E \rightarrow E + T \mid T$ 
  - $T \boldsymbol{\rightarrow} T \ast F \mid F$
  - $F \rightarrow (E) \mid P$
  - P→ ~ E | num Input: (~ num + num \* num)



(5+3+2)

(5+3+2)

- 3C. Consider the following expressions represent in terms of implementations of three address code. a = b + c \* d;c = b + e / r;
- 4A. Construct the SLR parsing table and parse the string for the following grammar. (5+3+2)
  - $S \rightarrow AxB \mid B$   $A \rightarrow yB \mid z$   $B \rightarrow A$ String: yzyz

4B. Eliminate the left recursion from the following grammar.  $A \rightarrow B \mid a \mid CBD$   $B \rightarrow C \mid b$   $C \rightarrow A \mid c$  $D \rightarrow d$ 

4C. Define sentence and sentential form of grammars. Give an example for each.

(5+3+2)

- 5A. With a neat diagram explain the working of non-recursive predictive parser.
- 5B. Obtain a DFA to accept strings of a's and b's having odd number a's and odd number of b's on  $\sum = \{a,b\}$ .
- 5C. What is the necessity of augmenting the grammar in LR parsers? (5+3+2)

6A. Consider the following code. Write the three address code and optimize the code using optimization techniques. Assume A is allocated static storage and there are 4 bytes per word. X:=20
WHILE x<10 DO x:=x-1; A[x]:=10; IF x=4 THEN x:=x-2; ENDIF ENDDO Y:=x+5;</li>

- 6B. With a neat diagram explain the phases of a compiler.
- 6C. What is a literal table? Explain

(5+3+2)