

GUJARAT TECHNOLOGICAL UNIVERSITY
BE - SEMESTER-VII EXAMINATION – WINTER 2015

Subject Code: 170701

Date: 12/12/2015

Subject Name: Compiler Design

Time: 10:30am to 1:00pm

Total Marks: 70

Instructions:

1. Attempt all questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.

Q.1 (a) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment. Indicate corresponding token and pattern. **07**

```
void swap (int a, int b)
{
    int k;
    k = a;
    a = b;
    b = k;
}
```

(b) Explain Semantic Analysis and Syntax Analysis phases of compiler with suitable example. Also explain the reporting errors by these two phases. **07**

Q.2 (a) Write a short note on Symbol Table Management. **07**

(b) Construct DFA for following Regular expression. Use firstpos, lastpos and followpos functions to construct DFA. **07**

$(a^* | b^*)^*$

OR

(b) Construct NFA for following Regular Expression using Thomson's Construction. Apply subset construction method to convert into DFA. **07**

$(a | b)^*abb$

Q.3 (a) Construct LL(1) parsing table for the following Grammar: **07**

$S \rightarrow (L) \mid a$
 $L \rightarrow L , S \mid S$

(b) Check whether the following grammar is CLR or not. **07**

$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$
 $A \rightarrow d$
 $B \rightarrow d$

OR

Q.3 (a) Define: Left Recursive. State the rule to remove left recursive from the grammar. **07**
Eliminate left recursive from following grammar.

$$\begin{aligned} S &\rightarrow Aa \mid b \\ A &\rightarrow Ac \mid Sd \mid f \end{aligned}$$

(b) Construct SLR Parsing Table for the following grammar. **07**

$$S \rightarrow 0S0 \mid 1S1 \mid 10$$

Q.4 (a) Explain Operator Precedence Parsing method with example. **07**

(b) Show syntax directed definition for simple desk calculator. Also show annotated parse tree for $3*5+4n$, where n indicates newline. **07**

OR

Q.4 (a) Explain LALR parser in detail. Support your answer with example. **07**

(b) Give the translation scheme that converts infix to postfix expression for the following grammar and also generate the annotated parse tree for input string $7+3+2$. **07**

$$\begin{aligned} E &\rightarrow E+T \\ E &\rightarrow T \\ T &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

Q.5 (a) Translate following arithmetic expression **07**

$$-(a * b) + (c + d) - (a + b + c + d) \text{ into}$$

- 1] Quadruples
- 2] Triple
- 3] Indirect Triple

(b) Explain various dynamic storage allocation techniques. **07**

OR

Q.5 (a) Explain any three code optimization techniques with example. **07**

(b) Explain various issues in design of code generator. **07**
