

# GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER-VII (OLD) EXAMINATION – WINTER 2018

**Subject Code: 170701**

**Date: 03/12/2018**

**Subject Name: Compiler Design**

**Time: 10:30 AM TO 01:00 PM**

**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) Discuss the phases of a compiler with sketch. **07**  
(b) I. Discuss Input Buffer Pairs for Lexical Analyzer. **07**  
II. List and Explain Cousins of Compiler.

- Q.2** (a) Explain working of non-Recursive Predictive Parser with diagram. **07**  
(b) What do you mean by ambiguous grammar? Show that following is an ambiguous grammar. **07**  
 $E \rightarrow E+E \mid E^*E \mid E-E \mid E/E \mid (E) \mid id$

**OR**

- (b) Construct a DFA without constructing NFA for the following regular expression. **07**  
 $a(a \mid b)^*ab$

- Q.3** (a) How do you check whether the grammar is LL (1) or not? Generate LL(1) parsing table for given Grammar **07**  
 $S \rightarrow iEtS \mid iEtSeS \mid a$   
 $E \rightarrow b$   
Is Grammar LL(1) or not?

- (b) Explain subset construction method for constructing DFA from an NFA with an example. **07**

**OR**

- Q.3** (a) Show that the following grammar **07**  
 $S \rightarrow AaAb \mid BbBa$   
 $A \rightarrow \epsilon$   
 $B \rightarrow \epsilon$   
is LL(1) but not SLR(1).

- (b) Write a short note on operator precedence parsing for +, \*, \$, id. **07**

- Q.4** (a) Explain syntax directed translation scheme for Infix to Postfix conversation with example. **07**  
(b) Write a syntax directed definition of a simple desk calculator and draw an annotated parse tree for  $4 * 3 + 2n$ . **07**

**OR**

- Q.4** (a) Explain Inherited attributes with the help of an example. **07**  
(b) I. Explain various parameter passing methods. **07**  
II. What is left factoring? Discuss it with the help of an example.

- Q.5** (a) What is Intermediate Code? Discuss various representations of three address code for  $a = (a + b * c) * (b * c) + (b + c) \wedge a$  **07**  
(b) Explain activation record. How is task divided between calling & called program for stack updating? **07**

**OR**

- Q.5** (a) Discuss code optimization with an example. **07**  
(b) Discuss Design Issues of Code Generator. **07**

\*\*\*\*\*