

Seat No.: _____

Enrolment No. _____

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER-VII (NEW) - EXAMINATION – SUMMER 2017

Subject Code: 2170701

Date: 29/04/2017

Subject Name: Compiler Design

Time: 02.30 PM to 05.00 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) Explain Semantic analysis and Syntax analysis phases of compiler with suitable example. Also explain the errors generated by these two phases. **07**
- (b) Construct the NFA using thompson's notation for following regular expression and then convert it to DFA. **07**
(a / b)* ab#
- Q.2** (a) Check following grammar is LL (1) or not? **07**
S → aB | ε
B → bC | ε
C → cS | ε
- (b) What is left factoring and left recursion? Explain it with suitable example. **07**
- OR**
- (b) Construct CLR parsing table for following grammar. **07**
S → aSA | ε
A → bS | c
- Q.3** (a) Show that following grammar is not a SLR (1) grammar. **07**
S → AaBa | BbBa
A → ε
B → ε
- (b) Develop a syntax directed definition for following grammar. **07**
E → TE'
E' → +TE' | ε
T → (E)
T → id
- OR**
- Q.3** (a) Write a grammar to declare variables with data type int or float or char. Also develop a syntax directed definition for that. Draw the dependency graph for same. **07**
- (b) Define operator precedence grammar. Construct precedence matrix and precedence graph for arithmetic grammar as shown below: **07**
E → E + T | T
T → T * F | F
F → (E) | id
- Q.4** (a) Explain Activation record and Activation tree in brief. **07**
- (b) Explain Quadruple, triple, and indirect triple with suitable example. **07**
- OR**
- Q.4** (a) Write a note on peephole optimization. **07**
- (b) Write a short note on symbol table management. **07**
- Q.5** (a) Define a following: Basic block, Constant folding, Natural loop, Handle **07**

(b) Construct DAG for $a + a * (b - c) + (b - c) * d$. also generate three address code for same. 07

OR

Q.5 (a) Discuss the issues in the design of code generation. 07

(b) Define dominators. Construct dominator tree for following graph. 07


