GUJARAT TECHNOLOGICAL UNIVERSITY
BE - SEMESTER–VII (NEW) EXAMINATION – WINTER 2018

Subject Code: 2170701   **Date: 15/11/2018**

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.**

			MARKS
Q.1	**(a)**	List the cousins of compiler and explain the role of any one of them.	**03**
	(b)	Write a brief note on input buffering techniques.	**04**
	(c)	Explain front end and back end of compiler in detail.	**07**
Q.2	**(a)**	Define the following terms and give suitable example for it. 1) Handle 2) Handle pruning 3) Left Factoring	**03**
	(b)	Explain all error recovery strategies used by parser.	**04**
	(c)	Construct LL(1) parsing table for the following Grammar: E -> E+T | T T -> T*F | F F -> (E) | a	**07**

OR

	(c)	Construct NFA for following regular expression using Thompson's notation and then convert it into DFA. (a/b)*abb#	**07**
Q.3	**(a)**	Define the following terms and give suitable example for it. 1) Augmented Grammar 2) LR(0) Item 3) LR(1) Item	**03**
	(b)	Differentiate Top Down Parsing and Bottom up parsing	**04**
	(c)	Construct SLR parsing table for the following grammar : S ->(L)|a L->L,S|S	**07**

OR

Q.3	**(a)**	Give the difference between SLR and CLR Parser.	**03**
	(b)	List the different conflicts that occur in Bottom up parsing and give examples for that.	**04**
	(c)	Implement the following grammar using Recursive Descent Parser. S -> Aa | bAc | bBa A -> d B -> d	**07**
Q.4	**(a)**	What is Ambiguous Grammar? Describe with example.	**03**
	(b)	Give the difference between synthesized attributes and inherited attributes	**04**
	(c)	Construct CLR parsing table for the following grammar : S ->AA A->aA|b	**07**

OR

Q.4	**(a)**	List the different issues in code generation phase end describe any two issues.	**03**
	(b)	Explain parameter passing techniques for procedure.	**04**
	(c)	Explain Quadruple, triple and indirect triple with suitable example.	**07**

Q.5 **(a)** Draw syntax tree and DAG for the statement **03**
x=(a+b)*(a+b+c)*(a+b+c+d)

(b) Explain dynamic memory allocation strategy. **04**

(c) What is an activation record? Explain how they are used to access local and **07**
global variables.

OR

Q.5 **(a)** Write a note on stack allocation strategy. **03**

(b) Give the translation scheme that converts infix to postfix expression for the **04**
following grammar and also generate the annotated parse tree for input
string "id+id*id"
E -> E+T | T
T -> T*F | F
F -> id

(c) Discuss various code optimization techniques with examples. **07**

****