Seat No.: _____ *Enrolment No._____*

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER–VII • EXAMINATION – SUMMER • 2015

Subject code: 170701 **Date: 01/05/2015**

Subject Name: Compiler Design

Time: 02.30pm-05.00pm **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)** What are regular expressions? Find the regular expression described by DFA {{A,B},{0,1},δ,A,{B}},where δ is detailed in following table . **07**

	0	1
A	A	B
B	ϕ	A

Please note B is accepting state.
Describe the language defined by the regular expression.

(b) Construct the NFA using thompson's notation for following regular expression and then convert it to DFA. **07**
a^{+} (c |d) b^{*}f #

Q.2 **(a)** List the errors generated by the syntax analysis phase. Discuss error handling methods in the syntax analysis phase. **07**

(b) For the following grammar **07**
D → T L ;
L → L , id | id
T → int | float

1)Remove left recursion (if required)
2)Find first and follow for each non terminal for Resultant grammar
3)Construct LL(1) parsing table
4)Parse the following string (show stack actions clearly) and draw parse tree for the input:
int id, id;

OR

(b) How top down and bottom up parser will parse the string **'bbd'** using grammar A → bA | d. Show all steps clearly. **07**

Q.3 **(a)** Explain the phases of compiler with an example. **07**

(b) Construct the collection of sets of LR(0) items for the following grammar. **07**

S-> Aa | bAc | dc | bda
A->d

OR

Q.3 **(a)** Construct DFA by syntax tree construction method. **07**
$a^+ b^* (c|d) f \#$
Optimize the resultant DFA.

(b) Develop a predictive parser for the following grammar. **07**
S'->S
S->aA|b|cB|d
A->aA|b
B->cB|d

Q.4 **(a)** Explain the following: **07**
1) The Handle
2) Left Factoring
3) Directed Acyclic Graph
4) Conflicts in LR Parsing
5) Parser Generator
6) Dependency Graph
7) Locality of reference

(b) Construct an SLR Parsing table for the following grammar. **07**
E->E-T|T
T->F↑T|F
F->(E) | id

OR

Q.4 **(a)** Elaborate the term "Activation Record" in detail. **07**

(b) Discuss various code optimization techniques. **07**

Q.5 **(a)** Write a context free grammar for arithmetic expressions. Develop a syntax directed definition for the grammar. Draw an annotated parse tree for the input expression: (3*2+2)*4 **07**

(b) Convert the following statement into triple, indirect triple and quadruple forms. **07**
A= (B+C) $ E + (B+C) *F

OR

Q.5 **(a)** Discuss synthesized and inherited attributes using a suitable grammar. **07**

(b) What is intermediate code? What is its importance? Discuss various representations of three address code. **07**

****