Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

M. E. - SEMESTER – II • EXAMINATION – WINTER • 2014

Subject code: 1720103 Date: 04-12-2014

Subject Name: Advance Compiler Design

Time: 02:30 pm - 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)** Define: Interpreter, Symbol table, Bottom-up parsing, L-attribute, Loader, Predictive parser, Handle. **07**

(b) Write a regular expression for given language. **07**

All strings of *a* and *b* having even number of *a*ôs.

Construct optimized DFA for the same without constructing NFA.

Q.2 **(a)** What are three address statements? Explain the methods to implement them with suitable examples. **07**

(b) Write the input, output and action performed by each phase of compiler for following C language statement. (Also mention the entry in data structure if required.) **07**

x = y % 10 + 5;

OR

(b) Construct a syntax directed translation scheme that translates arithmetic expressions from postfix notations into infix notation. Give annotated parse tree for the inputs 95-2*. **07**

Q.3 **(a)** Write a Lex program to replace all non-null sequence of white spaces by single blank character. **07**

(b) Write a regular expression for following language and draw DFA. **07**

All strings of *0's* and *1's* that do not contain the substring 011.

OR

Q.3 **(a)** Write a Yacc program to evaluate an arithmetic expression involving addition and multiplication. **07**

(b) Find LR(1) item-sets for the following grammar. **07**

S -> CC

C -> aC | b

Q.4 **(a)** Consider the following grammar: **07**

S -> (L) | a

L -> L,S | S

1. What are the terminals, non-terminals and start symbol?
2. Construct left-most derivation and draw parse tree for sentence (a,(a,a))

(b) Consider the following grammar: **07**

S -> AS | b

A -> SA | a

Construct SLR parsing table.

OR

Q.4 **(a)** Remove left recursion and do left factoring for following grammar. **07**

E -> E + E | E ï E | E * E | E / E | (E) | id

(b) Translate the expression ï (a+b)*(c+d)+(a+b+c) into quadruples and indirect triples. **07**

Q.5 **(a)** What are the issues in the design of code generator? **07**

(b) What do you mean by peephole optimization? Explain any three characteristics of it with suitable example. **07**

OR

Q.5 **(a)** Explain following major code optimization methods with suitable example. **07**

1. Common sub-expression elimination
2. Strength reduction
3. Dead code elimination

(b) Construct the DAG for following basic block. **07**

d = b * c
e = a + b
b = b * c
a = e - d

□□□□□□□□□□□□