Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

B. E. - SEMESTER – VII • EXAMINATION – WINTER 2012

Subject code: 170701 Date: 26/12/2012

Subject Name: Compiler Design

Time: 10.30 am – 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)** Find errors and identify the phase of compiler detecting them for following C program segment. Justify your answers. **07**

```
int fi( int);
char a[10], * cptr;
int k = 1 ; int j = 2;
float f;
cptr = a;
if (k);
fi(k);
fi( j )
++k;
*(cptr + 1 ) = 0 ;
++ a;
n + *k ;
```

(b) What is a symbol table? Discuss any two data structures suitable for it & compare their merits / demerits. Also compare one pass & two pass compilers. **07**

Q.2 **(a)** Explain lexical analysis phase of a compiler and, for a statement given below, write output of all phases (except of an optimization phase) of a complier. Assume a, b and c of type float **07**

a = a + b * c * 2;

(b) Construct a DFA without constructing NFA for following regular expression. Find minimized DFA. **07**

a*b*a(a | b)*b*a#

OR

(b) Construct a NFA for following regular expression using Thompson's notation and then convert it into DFA. **07**

aa*(b | c) a*c#

Q.3 **(a)** Write unambiguous grammar for producing arithmetic expression consisting of symbols id, +, - , /, $. Find first & follow of non terminal symbols of the grammar for non recursive predictive parser. Construct parse table and parse following string. **07**

id - id + id - id $

(b) (i) Consider the grammar **07**

S -> SS+ | SS* | a

Show that the string aa+a* can be generated by the grammar.

Construct the parse tree for it. Is the grammar ambiguous? Justify.

(ii) Write production rules for producing following language.

Strings of 0's and 1's with equal numbers of 0's and 1's.

OR

OR

Q.3 **(a)** Write unambiguous production rules for producing arithmetic expression consisting of symbols id, *, -, ( ) and ^, where ^ represents exponent. Parse following string using shift – reduce parser: **07**

id – id * id ^ id * (id ^ id) ^ id

Explain various conflicts of a shift – reduce parser.

(b) A robot is to be moved to a unit step in a direction specified as a command given to it. The robot moves in the direction North, South, East, West on receiving N, S, E, W command respectively & in the direction North-East , North-West, South-East, South-West on receiving A, B, C, D commands respectively. The current position of the robot is initialized to (0,0) Cartesian coordinates on receiving command Start. Write production rules for producing sequence of commands and semantic rules for knowing position of a robot after receiving a sequence of commands. Draw annotated parse tree for following sequence: **07**

Start N N A A C C N

Q.4 **(a)** Explain SLR parser in detail with the help of an example. **07**

(b) Draw transition diagrams corresponding to production rules for operators +, - , *, / and id for a predictive parser. Explain how parsing takes place for it. **07**

OR

Q.4 **(a)** Explain operator precedence parser by giving example for constructing a precedence graph and table. **07**

(b) Write ambiguous and unambiguous production rules for if then else construct. Illustrate parsing using both types of rules by giving an example. Also explain left factoring and its use. **07**

Q.5 **(a)** Explain how type checking & error reporting is performed in a compiler. Draw syntax tree and DAG for following statement. Write three address codes from both. **07**

a = (a + b * c) ^ (b * c) + b * c

(b) (i) Explain activation record. How is task divided between calling & called program for stack updating? **07**

(ii) Explain various parameter passing methods.

OR

Q.5 **(a)** Explain heap, dynamic storage allocation techniques and synthesized attributes. **07**

(b) (i) Explain any three types of optimization techniques. **07**

(ii) Describe code generator design issues.

****