CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

Seventh Semester of B. Tech. Examination (CE)
December 2012
CE-401 Compiler Construction (CC)

Date: 04.12.2012, Tuesday Time: 10:00 a.m. To 01:00 p.m. Maximum Marks: 70

Instructions:

1. The question paper comprises of two sections.
2. Section I and II must be attempted in separate answer sheets.
3. Make suitable assumptions and draw neat figures wherever required.
4. Rough work is to be done in the last page of main supplementary, please don't write anything on the question paper.
5. Indicate clearly, the option(s) you attempt along with its respective question no.
6. Figures to the right indicate marks.

SECTION-I

Q-1 **Answer the following questions.**

1. Prepare NFA and DFA for: (0 | 1)* 101 (0 | 1)* 3
2. Explain the processing of various phases of 'C' compiler on the following statement. 4
   answer= data1 *(int)data2;
3. For the following C++ fragment, identify the lexemes that make up the tokens and also show symbol table entries for this module. 4

```
void main(int argc, int argv[])
  {char a,b,c;
   int m=a++2;  /* float b=5.6*/}
```

Q-2

[A] Write an algorithm for scanner, which can identify following 'C' language tokens. 4

(i) Keywords (if, else, while)
(ii) String constants e.g. "this is constant"
(iii) single line comments-/* this is comment*/
(iv) punctuation symbols – ";" ',', %,:,{,}

[B] 1. 4

```
A := 6 * (B+C);
D := 3 + 7 * (B+C);
E := A * (B+C);
```

2.

```
while (count < limit) do
INPUT SALES;
VALUE := SALES * ( MARK_UP + TAX );
OUTPUT := VALUE;
COUNT := COUNT + 1;
end;
```

1. Which kinds of optimizations possible on the code.
2. Eliminate common sub expression from/for each basic block
3. Find the induction variables of each loop and eliminate them.

OR

[B] Explain the Linking Loader and its significance in the execution of the program in a compiler. Explain the structure of Activation Record. 4

[C] Find out the operator precedence parsing for the following Input string. 4
id * (id ↑ id) – id /id. Assume the Grammar. **OR** Explain the Operator Precedence Parsing by giving your own example.

Q-3

[A] Discuss the importance of symbol table in compiler design. How is symbol table manipulated at various phases of the compilation? What should be the typical 4

entries in the symbol table for the 'C' language? Explain how scope rules and block structure of a programming language influence the organization strategies for symbol table.

[B] What is the difference between parse tree and syntax tree? Draw the parse tree for following expression: 4
a= a + a * b + a * b * c – a / b + a * b and write three address code for it.

OR

[B] Write quadruples for the expression: (a+b) * (c+d) – (a+b+c) 4

[C] What are the challenges in the design of the compiler? Which are the factors to be considered in the task of code generation? 4

OR

[C] Draw the block diagram of communication between Scanner(s) and Parser. Explain the types of errors detected by lexical analysis phase. 4

SECTION-II

Q-4

1. Show that the following grammar is LR(1) but not LALR(1): 4
   S→Aa | bAc | Bc | bBa
   A→ d
   B→ d
2. Test whether the following grammar is LL (1) or not. Construct predictive parsing table for it. 4
   S'→ S#
   S→ aSA | Є
   A→ bS | c
   Parse the input string aabacc#.
3. Left factor the following Grammar: 3
   S→iEtS |iEtSeS | a
   E→ b

Q-5

[A] What language does following grammar generate? Justify your answer. 4
S→ aSbS | bSaS | Є
Is the following grammar ambiguous? Give reason.
S→ aS | aSbS | c

[B] S→ (L) | a 4
L→ L,S | S
What are the terminal and non terminal symbols? Contract the left-most and right-most derivations and prepare the corresponding parse trees (derivations) for the sentence: (a,(a,a)).

OR

[B] A RISC processor has smaller number of instructions and more number of registers than a CISC machine. Describe the pros and cons of compiler design targeted to a RISC machine as compared to a CISC machine. 4

[C] Give the translation of three address code for the following stmt: 4

```
while a < b do
 if c < d then
   x:= y + z
 else
   x:= y-z
```

OR

[C] Which type of compiler is better- One running slowly but producing optimized code and another one running very fast but producing unoptimized code? Can a human being optimize a program better than an automated compiler? Justify your answers. 4

Q-6

[A] Eliminate the left recursion from the following grammar: 4

S→a | ^ | (T)

T→ T,S | S

Whether the transformed grammar is LL (1)?

[B] The following grammar for if-then-else statements is proposed. 4

Stmt→ if expr then stmt | stmt | if expr then stmt else stmt.

Show that the grammar is ambiguous. Write an equivalent unambiguous grammar for the same.

OR

[B] Consider the grammar that has the productions. 4

S→ bA | aB

A→ bAA | aS | a

B→ aBB | bS | b.

Find the equivalent grammar in CNF.

[C] Write Syntax Directed Definition for translating following grammar for postfix notation. Also draw annotated parse tree for 9-5+2. 4

expr → expr + term

expr → expr – term

expr→ term

term → 0 | 1 |.....|9

OR

[C] S→ S#

S → ABC

A→ a | bbD

B→ a | ∈

C→ b | ∈

D→ c | ∈

Construct the FIRST and FOLLOW sets for the given Grammar.

Design LL (1) Parsing table.