Ex/CSE/T/321/74/2011

3rd Year, Comp. Sc. & Engg. 2nd Semester Examination, 2011
Compiler Design

Time – 3 Hours **Full Marks – 100**

Use separate answer scripts for two parts
Answer all parts of a single question on contiguous pages

Part-I

Answer any two questions

1.

a. Describe briefly different phases of a compiler. State the reason of having an intermediate representation of the source program during compilation.

b. Draw a DFA that accepts the four keywords **case**, **char**, **const**, and **continue** of C programming language.

c. Given an alphabet $\sum = \{0, 1\}$, L is the set of all strings of alternating 0 and 1s. L={∈, 0, 01, 010, 0101, ...}

   i. Derive the regular expression *r* describing the above language.

   ii. Use Thompson's construction to convert the above regular expression into an NFA.

   iii. Convert the NFA to equivalent DFA using subset construction.

d. Compare the time and space complexity for determining whether a given binary string *x* is in L(*r*) using two approaches

   i. Constructing NFA from *r*.

   ii. Constructing DFA from *r*.

(4+2)+3+(2+3+7)+4=25

2.

a. Derive the regular definition of currency in dollars, represented as a positive decimal number rounded to the nearest one-hundredth. Such numbers begin with the character $, have commas separating each group of three digits to the left of the decimal point, and end with two digits to the right of the decimal point. Examples: $3,456.78 , $4,444,444.00 , $0.50

b. What do you mean by runtime environment of a program?

c. What do you mean by activation of a procedure/function? Draw the activation tree of the following program snippet.

```
int gcd(int u, int v){
if(v==0)                return(u);
        else                    return(gcd(v, u%v));
}
main(){
        gcd(100, 15);
}
```

d. Differentiate between static and dynamic scope rule. Give suitable examples.

e. Give output of the following program using each of the four parameter passing methods call by value, call by reference, call by name, call by value result. Give justifications for your answer.

```
#include<stdio.h>
int i=0;
void swap(int x, int y)
{
        x=x+y;
        y=x-y;
        x=x-y;
}
main()
{
        int a[3]={1, 2, 0};
        swap(i, a[i]);
        printf("%d %d %d %d\n", i, a[0], a[1], a[2]);
        return(0);
}
```

5+2+4+6+8=25

3.

a. Describe how variable length data are handled in stack storage allocation strategy.
b. Explain how static or lexical scope is implemented in case of programming languages which supports nested procedure definition.
c. Explain the "Deep Access" and "Shallow Access" method for implementing dynamic scope. Give examples of each.
d. What is a display? Explain one method for its implementation.

4+8+8+5=25

4.

a. What is an "activation record"? Describe different fields of an "activation record".
b. Show the activation record for a call to function f. Determine the offset of the variables x, c, a[8], and y with respect to frame pointer (fp). Assume that two, four, one, and eight bytes are required for storage of integers, addresses, characters and double-precision floating point numbers. Also state the activities to be performed by the "caller" and the "called" procedure in the "calling sequence" and "return sequence".

```
int f(int x, char c, double d){
        double y; int a[10];
        .....
}
```

```
mani(){
        f(2, 'a', 12.09);
}
```

c. State necessity of "input buffering" in implementation of lexical analyzers.
d. Write regular expressions for the following
    i. An uppercase DOS path which only has letters numbers and underscores in its text
    ii. A valid IP address

6+(5+4)+4+(3+3)=25

Part-II

Answer any two questions

1.

a. Show that the following grammar is LL(1).

E → - E
E → (E)
E → V T
T → - E
T → **λ**
V → **id** R
R → (E)
R → **λ**

Construct the LL(1) parsing table for the above grammar. Write at least one string (with not less than five tokens) that can be generated from the grammar along with the leftmost and rightmost derivations of the string.

b. Discuss (with examples) the types of conflicts that may arise in SLR(1) parsing. Briefly discuss how these conflicts can be removed in canonical LR or LALR parsing.

c. Briefly discuss how the scope rules are implemented in a symbol table.

(8+4)+(4+4)+5=25

2.

a. Construct an SLR(1) parsing table for the following grammar.

stmt → IF | **other**
IF → **if (** exp **)** stmt | **if (** exp **)** stmt **else** stmt
exp → **0** | **1**

What kind of conflict do you find? Explain why.

b. Consider the grammar $S \rightarrow S\ S\ +\ |\ S\ S\ *\ |$ **a** . Give a rightmost derivation for the string "a a + a *" and draw the parse tree.

Is the grammar ambiguous? Justify your answer.

c. Briefly discuss on different types of intermediate representations.

(12+2)+(3+2)+6=25

3.

a. Discuss attribute grammars and explain why they are needed.

b. Explain S-attributed and L-attributed definitions.

Consider the following attribute grammar:

Grammar Rule	Semantic Rules
S → A B C	B.u = S.u
	A.u = B.v + C.v
	S.v = A.v
A → a	A.v = 2 * A.u
B → b	B.v = B.u
C→ c	C.v = 1

Draw the annotated parse tree for the string *abc* and draw a dependency graph for the associated attributes.

Find whether the above grammar follows S-attributed or L-attributed or none of these two. Justify your answer.

c. Define Context-Free Grammar. Explain the terms "sentence" and "sentential form".

(3+2)+(4+2+4+4)+6=25

4.

a. What are the different categories of code optimisation techniques?
Give an example of each of the optimisation technique mentioned below:
   i. Common sub-expression elimination
   ii. tail recursion removal
   iii. loop fission
   iv. loop fusion
   v. strength reduction.

b. Find the FIRST and FOLLOW sets for the following grammar; then construct the LL(1) parsing table.

$S \rightarrow u\ B\ D\ z$
$B \rightarrow B\ v$
$B \rightarrow w$
$D \rightarrow E\ F$
$E \rightarrow y \mid \varepsilon$
$F \rightarrow x \mid \varepsilon$

c. Translate the arithmetic expression a*b+a*b*c into:
   i. Quadruple
   ii. Triple
   iii. Indirect triple

(3+7)+(4+6)+5=25

--------|-------