# 3rd Year, Comp. Sc. & Engg. 2nd Semester Examination, 2013
## Compiler Design

**Time – 3 Hours** **Full Marks – 100**

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

### Part-I
Answer question 1 and any two from the rest

1.

a. What is a just in time compiler?
b. What are dynamic semantic rules? Give examples.
c. What is a control stack?
d. What is the significance of intermediate code generation during compilation?
e. What do you mean by the terms *l-value* and *r-value*. Give suitable examples.

2+2+2+2+2=10

2.

a. Given an alphabet $\Sigma = \{0, 1\}$, L is the set of all strings of alternating 1's and 0's. L={∈, 1, 10, 101, 1010, ...}
   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.

b. Explain with justification in the context of the following C program snippet that C does not support call by reference parameter passing mechanism.

```
void swap(int *x, int *y){
        int temp;
        temp=*x;
        *x=*y;
        *y=temp;
}

main(){
        int a=4, b=8;
        swap(&a, &b);
}
```

c. Explain how display provides faster access to nonlocal data items than with access link.

(2+3+7)+4+4=20

3.

a. What is a regular definition? 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. Define static scope and dynamic scope. Determine output of the following program snippet for a programming language having C like syntax considering both static and dynamic scope.

```
float r;
void show(void){
        printf("%f ", r);
}
void small(void){
        float r;
        r=0.12;
        show();
}
main(){
        r=1.48;
        show();
        small();
}
```

c. Explain the "Deep Access" and "Shallow Access" method for implementing dynamic scope using the example above. Show the activation records in the control stack in both the cases.

(2+4)+(2+4)+8=20

4.

a. Show the activation record for a call to function f. Determine the offset of the identifiers x, c, f, a[0][3][4], and y with respect to frame pointer (fp). Assume that four, four, one, and eight bytes are required for storage of integers, addresses, characters and double-precision floating point numbers and row major storage for multidimensional array.

```
int f(char c[3], double f, int x[]){
        int a[1][4][8]; double y;
        .....
}
```

b. Consider a programming language which supports static scope with nested procedures. Assume that procedure p at nesting depth $n_p$, calls a procedure x at nesting depth $n_x$. Write the rules for setting up access link.

The following table shows nesting depths of different procedures in a program.

| Procedure | Nesting Depth |
|---|---|
| S | 1 |
| R | 2 |
| E | 2 |
| Q | 2 |
| P | 3 |

Show how the access links are set up for the following sequence of procedure invocations, S→Q→Q→P→E.

c. Describe how C compiler handles variable number of arguments to functions like printf, scanf etc.

6+(4+6)+4=20

## Part-II

Answer question no. 1 and any four from the rest

1. Answer any four questions:-
   a. Define context sensitive grammar and explain with suitable example why it is termed as context sensitive.
   b. What are the implications of computing First and Follow sets in the context of top-down parsing?
   c. What do you mean by Syntax-directed Translation? How is a translation scheme written for an inherited attribute?
   d. What is an L-attributed definition? What restrictions are imposed in using attributes in case of L-attributed definitions and why?
   e. What are the different stages in a compiler?

   2.5 x 4 = 10

2.
   a. Define Context-Free Grammar. Explain the terms "sentence" and "sentential form".
   b. Is the following grammar LL(1)? Justify your answer.

   $E \rightarrow - E$
   $E \rightarrow ( E )$
   $E \rightarrow V T$
   $T \rightarrow - E$
   $T \rightarrow \lambda$
   $V \rightarrow ID\ R$
   $R \rightarrow ( E )$
   $R \rightarrow \lambda$

   Write at least one string (with not less than five tokens) that can be generated from the grammar along with the leftmost derivation of the string.

   4+4+2=10

3.
   a. For the grammar given in Question No. 2, construct the LL(1) parsing table.
   b. Translate the arithmetic expression 2*x*y+x*y-c into:
   (i) Abstract syntax tree, (ii) Quadruple, and (iii) Triple.

   6+4=10

4.
   a. Construct an LALR parsing table for the following grammar.

   $S \rightarrow id \mid V := E$
   $V \rightarrow id$
   $E \rightarrow V \mid n$

   What kind of conflict do you find? Explain why.

   10

5.
   a. What is a three-address code? What is a static single assignment form? Discuss the benefits of each.
   b. Write the three-address code of the following program fragment:

```
while (x <= y)
{ x = x * 2;
}
y = x + 4 * y + z;
```

c. Write the three address code and static single assignment form of the following program fragment:

```
z = x * 2 + y * 2;
y = sin (z);
z = x * 4 + (a + b)/4;
x = cos (z);
x = x * x + y * y;
```

3+2+5=10

6.

a. Discuss the different scopes for code optimization techniques.

b. Give an example of each of the following code optimisation techniques and explain how the program execution is benefited.
(i) Loop fission, (ii) Loop invariant code motion, (iii) Common sub-expression elimination, (iv) Loop unrolling.

2+2x4=10

7.

a. Consider the following attribute grammar:

| Grammar Rule | Semantic Rules |
|---|---|
| $S \rightarrow A\ B\ C$ | B.u = S.u |
| | C.u = A.v |
| | A.u = B.v + C.v |
| | S.v = A.v |
| $A \rightarrow a$ | A.v = 2 * A.u |
| $B \rightarrow b$ | B.v = B.u |
| $C \rightarrow c$ | C.v = C.u = 2 |

Draw the annotated parse tree for the string abc and draw a dependency graph for the associated attributes. If the evaluation begins with the value S.u = 3, what value does S.v have after attribute evaluation. Find whether the above grammar follows S-attributed or L-attributed or none of these two. Justify your answer.

4+4+2=10

8.

a. Distinguish between backtracking parsing technique and predictive parsing technique.

b. What is handle pruning? Explain with an example.

c. Consider the following grammar:
$S \rightarrow AA$
$S \rightarrow bc$
$A \rightarrow baA$
$A \rightarrow c$
Construct the DFA of LR(0) items for this grammar.

2+3+5=10

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