Ex/CSE/T/321/74/2012(S)

3rd Year Comp. Sc. & Engg., 2nd Semester Examination, 2012 (Suppl.)
Compiler Design

Time – 3 Hours **Full Marks – 100**

Use separate answer scripts for two parts
Answer all parts of a question together

Part-I

Answer question 1 and any two from the rest

1.

a. What is the usefulness of the code optimization phase of a compiler?
b. How does "input buffering" improves performance of lexical analyzers?
c. What is a "control stack"?
d. What is a lexeme?
e. State why it is not safe to return address of a local variable from a function.

2x5=10

2.

a. Describe briefly different phases of a compiler.
b. Given the regular expression *r*=ab(a|b)*
   i. Construct an equivalent NFA using Thompson's Construction
   ii. Convert this NFA to a DFA using Subset Construction algorithm
c. Derive the regular definition of currency in dollars, represented as a positive decimal number. Such numbers begin with the character $, have commas separating each group of three digits. Example: $3,456

4+3+8+5=20

3.

a. Define static scope and dynamic scope. State advantages and disadvantages of the two methods. Give suitable examples of each.
b. Explain the "Deep Access" and "Shallow Access" method for implementing dynamic scope. Give examples of each.
c. Differentiate between heap storage allocation strategy and stack storage allocation strategy.
d. Give your arguments in favor/against the assertion "C supports call by reference".

6+6+4+4=20

4.

a. Describe four parameter passing methods call by value, call by reference, call by name, call by value result with suitable examples.
b. Identify the roles of the "caller" and the "called" procedure in the "calling sequence" and "return sequence".
c. Consider the following C function f.

```
void f(int n){
        char *p;
        p=(char *) malloc(n);
        .....
```

}

Show the content of the activation record for the call f(2). Also show how the content of the activation record changes for another call f(6).

10+5+5=20

Part-II

Answer question 1 and any two from the rest

1. Answer any five questions:
   a. Define context-sensitive grammars.
   b. Give an example of Loop Fission. Explain how it improves the code.
   c. What is Shift-reduce parsing?
   d. Explain the differences between a parse tree and an abstract syntax tree.
   e. Give two examples of improvement of code performance through strength reduction.
   f. Why is semantic analysis important?

   2x5=10

2.
   a. Write the sequence of three-address code instructions corresponding to each of the following expressions:
      i. a*b+a*b*c
      ii. 2+(3+(4+5))
      iii. if (v == 0) then x = y
      iv. x = add (x, y)
   b. Consider the following grammar:

      E→B A
      A→& B A | ε
      B→true | false

      Construct the FIRST and FOLLOW sets for the grammar. Construct the LL(1) parsing table for the above grammar. Also show the error recovery entries in the parsing table.
      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.

      (1.5x4)+(4+4+2+4)=20

3.
   Consider the following grammar:

      S →Aa | bAc | dc | bda
      A →d

   a. Construct the DFA of LR(0) items for this grammar.
   b. Construct the SLR(1) parsing table. Is there any conflict in the parsing table? If so then discuss the type of the conflict and reason for such conflict.
   c. Construct the DFA of LALR(1) items for the above grammar.

   6+6+2+6 = 20

4.
   a. What are attribute grammars? Define synthesized and inherited attributes in attribute grammars? Explain S-attributed and L-attributed definitions with examples.
   b. Suppose we have a production A→BCD. Each of the four non-terminals A, B, C and D have two attributes: s is a synthesized attribute, and i is an inherited attribute. For each sets of rules below, tell whether (i) the rules are consistent with an S-attributed definition, (ii) the rules are consistent with an L-attributed definition.
      i. A.s = B.i + C.s
      ii. A.s = B.i + C.s and D.i = A.i + B.s

iii. A.s = B.s + D.s

iv. A.s = D.i, B.i = A.s + C.s, C.i = B.s and D.i = B.i + C.i

Justify your answer with appropriate representation of the rules in annotated parse tree form.

c. Consider the grammar A → AA | (A) | ε, Give some examples of strings generated by this grammar. Show that the grammar is ambiguous.

(2+2+4)+2x4+4=20

5.

a. What is parsing? What are the different techniques for top-down parsing?

b. Consider the following grammar:

decl → type list

type → int | float | char

list → ident , list | ident

i. Left factor the grammar.

ii. Construct the First and Follow sets for the non-terminals of the resulting grammar.

iii. Construct the LL(1) parsing table for the resulting grammar.

iv. Show the actions of the parser for the input string int x, y.

3+(2+6+6+3)=20

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