5/12/18

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR

B.Tech-M.Tech Dual Degree (CST) 7^{th} Semester Examination, 2018

Compiler Design (CS - 701)

Time: 3 hours **Full Marks: 70**

(Answer Question No 1 and any five from the remaining)

1. (a) What is code motion?

   (b) What do you mean by machine dependent and machine independent optimization?

   (c) List the different storage allocation strategies.

   (d) Convert the expression $a = b * -c + b * -c$ into three address statements.

   (e) Why is it necessary to generate intermediate code instead of generating target program directly?

   $[2+2+2+2+2=10]$

2. (a) Give a regular definition for e-mail addresses. An address consists of a name, followed by the separator @, and a domain. A domain is a sequence of at least two names, which are separated by a dot. Names are non-empty sequences of lowercase letters, digits and underscores (but underscores are not allowed in a leading position). For instance, **first_family@here.there** is valid, but **name@domain** (missing dot in domain) and **_name@ac._de** (leading underscores) are not valid.

   (b) What is a symbol table? List the functionalities of a symbol table. Describe any two methods of implementing a symbol table.

   $[3+9=12]$

3. Consider the following Grammar

$$E \to 5 + T | 3 - T$$
$$T \to V | V * V | V + V$$
$$V \to a | b$$

   where E is the start symbol, set of non-terminals are {E, T, V} and set of terminals are {5, 3, a, b}

   (a) Do the left factoring for the above grammar and compute the *FIRST* and *FOLLOW* set.

   (b) Construct a Predictive parsing table and check if the grammar is LL(1).

   (c) Show the parsing for the sentence, $5 + a * b$.

   $[6+3+3=12]$

4. Consider the following grammar

$$S \to CC$$
$$C \to cC | d$$

   where S is the start symbol, set of terminals are {c, d}

   (a) Construct the sets of LR(1) items.

   (b) Construct the LALR parsing table for this grammar.

   (c) Show the parsing actions using the parsing table generated in (b) on the input "*cdd*".

   $[5+4+3=12]$

5. (a) What do you mean by attributed grammars? Discuss the translation scheme for converting an infix expression to its equivalent postfix form.

   (b) Draw a flow diagram for the *FOR* loop and subsequently propose the semantic rules to generate three address code for the same.

   (c) Write the rules to partition a sequence of 3 address statements for forming basic blocks.

   $[4+4+4=12]$

6. (a) What is activation record? Explain the task of caller and callee when a procedure is called and returned from the procedure?

   (b) Consider the following program:

```
1. int foo(int a){
2.       int b = a++;
3.       return b*a;
4. }
```

```
5. int bar(int b){
6.       return foo(b);
7. }
```

```
8.  int main(){
9.       int c;
10.      c=6;
11.      return bar(c);
12. }
```

Assume that the compiler allocates all the variables in the stack. The single arguments of *foo* and *bar* can be passed either by-value or by-reference. For each of the four possible combinations (i.e. both arguments by-value, foo's argument by-value & bar's argument by-reference, etc.) describe:

i. What kind of data should the compiler put in the argument slots of the activation records for the calls on lines 6 and 11?

ii. What kind of assembly code will be necessary to retrieve the value of variable a on line 2?

$[6+(3+3)=12]$

7. (a) Consider the following basic block, in which all variables are integers and ** denotes exponentiation:

```
a = b + c
z = a ** 2
x = 0 * b
y = b + c
w = y * y
u = x + 3
v = u + w
```

Assume that the only variables that are live at the exit of this block are v and z. In order, apply the following optimizations to this basic block. Show the result of each transformation.

i. algebraic simplification

ii. common sub-expression elimination

iii. copy propagation

iv. constant folding

   (b) Discuss how induction variables can be detected and eliminated from the given intermediate code.

```
B2:  j:= j+1
     t1:=4*j
     t2:=a[t1]
     if t2<10 goto B2
```

$[(2+2+2+2)+4=12]$