CS 553 Compiler Construction — Fall 2005 **Homework #2**
Parsing, Data-flow Analysis, and Control-Flow Analysis **Due September 23, 2005**

Write your answers on another sheet of paper. Homework assignments are to be completed individually. Hand written submissions are fine, but they must be readable. Due at the beginning of class. Total points: 140, 5% of course grade

1. [10 Points] True or False. If a statement is false, explain how so.
   (a) Unreachable code elimination and dead code elimination are two phrases for the same optimization.
   (b) It is better to store data-flow information at the entry and exit points of nodes in the control-flow graph rather than on the edges.

2. [20 Points] Given the following context-free grammar, draw the parse tree that is implicitly computed by bison for the given input.

```
StmtList:  StmtList Stmt  |   /*empty */ ;

Stmt:    IfStmt  |  AssignStmt  | ForStmt | ... ;

AssignStmt:  T_ID = Expr;

// T_EQUAL  is "=="
Expr:    T_ID  |  Expr '+' Expr  |  Expr T_EQUAL Expr ;


IfStmt:   T_IF  '(' T_TRUE ')'   '{' StmtList '}'
    |  T_IF  '(' T_FALSE ')'   '{' StmtList '}' '
```

```
=====
Input
=====
if ( true ) {
    if ( false ) {
        x = a + b + c;
    }
}
```

3. [70 Points] Here is an example program.

```
    a = b+c;
    for (i=0; i<10; i++) {
        x = a * 4;
        if (x>g) {
            b = 20;
        } else {
            b = 10;
        }
    }
    if (b < 0) { goto error; }
    do {
        printf("Ok, the first time around\n");
bok:
        b = b - 5;
    } while (b > 0);
    goto done;
error:
    b = 30;
    goto bok;
done:
    return 0;
```

   (a) Write the three-address code for the program.

   (b) Draw the control-flow graph for the three-address code. Put as many statements as possible within each node.

   (c) Label the following items in the control-flow graph: loop entry edge, loop exit edge, loop header node, back edges, loop tail node, and loop preheader node.

   (d) Draw the dominator tree for the control-flow graph.

   (e) Specify which nodes are in loops. For example, loop1 = node 1, node 2, ...

   (f) Is the control-flow graph reducible? If not use node-splitting to make it reducible.

   (g) List the nodes in pre-order, post-order, and reverse post-order (start with post-order and then reverse the order of nodes). Which order should be used for forward data-flow analyses and why?

4. [20 Points] Give an example three-address program that results in the worst-case running time for an iterative data-flow analysis algorithm while computing liveness. Describe the general case of which your example is a subset. (Assume that the nodes are visited in post-order).

5. [20 Points] Give an example three-address program that results in the worst-case running time for an iterative data-flow analysis algorithm while computing reaching constants. Describe the general case of which your example is a subset. (Assume that the nodes are visited in reverse post-order).