Reading

1. Read Mitchell, Chapter 7

Problems

1. (40 points) ......................... Activation Records for Inline Blocks

   Mitchell, Problem 7.1

2. (8 points) ............................. Time and Space Requirements

   Mitchell, Problem 7.3

3. (15 points) ............................. Ada Parameter Modes

   The Ada programming language permits parameters to be labeled as in, out, or in out, as in
   the following procedure definitions, where T is some type:

   procedure test1(in x: T) is begin ... end
   procedure test2(out x: T) is begin ... end
   procedure test3(in out x: T) is begin ... end

   The modifiers, or modes, have the following meaning:

   • in: The value of the parameter x cannot be changed inside the procedure. If we call
     test1(y), the value of y is the same before and after the call.
   • out: The parameter x can be written to, but it cannot be read. If we call test2(y), the
     value of y after the call is the last value written to x in the procedure.
   • in out: The parameter x can be both read and written, and the value of y after a call to
     test3(y) is the last value written to x in the procedure.

   The language definition does not specify how each mode should be implemented, and the compiler
   may use any appropriate parameter passing mechanism to implement them.

   (a) Which parameter passing mechanism could be used to implement test1, test2, and test3?
   The choices are pass-by-reference, pass-by-value, and pass-by-value-result (as described in
   problem 7.6). If more than one is possible, describe the advantages/disadvantages of each.

   (b) Consider the following procedure that takes two parameters. Does the following program
   print the same value for all strategies you outlined for in out parameters above?

   procedure incTwo(in out x:integer, in out y:integer) is
     begin
       x := x + 1;
       y := y + 1;
     end
procedure main() is
    w : integer = 3;
    begin
        incTwo(w, w);
        print w;
    end

(c) Discuss the advantages and disadvantages of permitting the compiler such flexibility in how it implements parameter modes.

4. (10 points) .......................................................... Static and Dynamic Scope
   Mitchell, Problem 7.8

5. (10 points) ............................................................ Eval and Scope
   Mitchell, Problem 7.10

6. (18 points) ......................................................... Lambda Calculus and Scope
   Mitchell, Problem 7.11

7. (15 points) ....................... Function Calls and Memory Management
   Mitchell, Problem 7.12

8. (15 points) ....................... Function Returns and Memory Management
   Mitchell, Problem 7.13