Programming Languages
CSCI-GA.2110.001 Fall 2015

Homework 1
Due Sunday, October 11

You should write the answers using word, latex, etc., and upload them as a PDF document. No implementation is required. Since there are drawings, if you prefer, you can hand write the answers and scan them to a PDF.

1. Provide regular expressions for defining the syntax of the following.
   (a) Passwords consisting of letters and digits that contain at least two upper case letters and one digit. They can be of any length (obviously at least three characters).
   (b) Floating point literals that specify an exponent, such as the following: 2.43876E13 (representing 243.867×10^{13}).
   (c) Procedure names that: must start with a letter; may contain letters, digits, and _ (underscore); and must be no more than 10 characters.

2. (a) Provide a simple context-free grammar for the language in which the following program is written. You can assume that the syntax of names and numbers are already defined using regular expressions (i.e. you don't have to define the syntax for names and numbers).

```
program one;

  var x;

  function f(var x, var y)
    var z;
  begin
    z := x+y-1;
    return z*2;
  end f;

  procedure g()
    var a;
  begin
    a := 3;
    x := a;
  end g;

begin
 g();
 print(f(x));
end one;
```

   You only have to create grammar rules that are sufficient to parse the above program.

   (b) Draw the parse tree for the above program.

3. (a) Define the terms *static scoping* and *dynamic scoping.*
   (b) Give a simple example, in any language you like (actual or imaginary), that would illustrate the difference between static and dynamic scoping. That is, write a short piece of code whose result would be different depending on whether static or dynamic scoping was used.
   (c) In a block structured, statically scoped language, what is the rule for resolving variable references (i.e. given the use of a variable, how does one find the declaration of that variable)?
   (d) In a block structured but dynamically scoped language, what would the rule for resolving variable references be?

4. (a) Draw the state of the stack, including all relevant values (e.g. variables, return address, dynamic link, static link), during the execution of procedure S in the following program.

```
procedure P;
  procedure Q(procedure R)
    procedure  S(x:integer);
    begin
      writeln(x);
    end;
 begin (* Q *)
   R(S);
 end;
 procedure T;
   procedure U(procedure V);
    begin
       V(6);
    end;
  begin (* T *)
      Q(U);
  end;
begin (* P *)
 T;
end;
```

   (b) Explain why closures on the heap are needed in some languages, and give an example of a program (in any syntax you like) in which a closure would need to be allocated on the heap.

5. For each of these parameter passing mechanims,

   (a) pass by value
   (b) pass by reference
   (c) pass by value-result
   (d) pass by name

   state what the following program (in some Pascal-like language) would print if that parameter passing mechanism was used:

```
program foo;
  var i,j: integer;
      a: array[1..6] of integer;

  procedure f(x,y:integer)
  begin
    x := x * 3;
    i := i + 1;
    y := a[i] + 2;
  end

begin
  for j := 1 to 6 do a[j] = j;
  i := 1;
  f(i,a[i]);
  for j := 1 to 6 do print(a[j]);
end.
```

6. (a) In Ada, define a procedure containing two tasks, each of which contains a single loop. The loop in the first task prints the numbers from 1 to 500, the loop in the second task prints the numbers from 501 to 1000. The execution of the procedure should cause the tasks to alternate printing fifty numbers at a time, so that the user would be guaranteed to see:

   `1 2...50 501 502 ...  550 51 52 ...  100 551 552 ...  600 ...`

   Be sure there is only one loop in each task.

   (b) Looking at the code you wrote for part (a), are the printing of any of the numbers occurring concurrently? Justify your answer by describing what concurrency is and why these events do or do not occur concurrently.