CSCE 330 Fall 2005
SECOND MIDTERM EXAM
Wednesday 2005-11-16
Closed Book, except for the denotational semantics handout

1 Syntax–10 points

1. (2 points) What does it mean for a (context-free) grammar to be *ambiguous*?

   Answer: There is at least one sentential form that can be generated by two distinct parse trees.

2. (8 points, Robert Sebesta) Prove that the following grammar is ambiguous:
   ```
   <S> -> <A>
   <A> -> <A> + <A> | <id>
   <id> -> a | b | c
   ```
   Answer: Provide two distinct parse trees derivations of a+b+c

2 Denotational and operational semantics–30 points

1. (2 points) What is aliasing? **Answer:** It is the situation in which two names denote the same entity at the same program point.

2. (3 points) Complete this sentence: The memory function, *mem* is a function from the set of *mysteries1* to the set of *mysteries2*. **Answer:** identifiers, integers (also acceptable: identifiers, values).

3. (5 points) Why is aliasing difficult to capture in denotational semantics? **Answer:** Because one has to enforce the requirement that some identifiers have the same value at all times.

4. (10 points).

   What is $dsem(read(y); x := y; write(x), < mem, i, o >)$, where $mem(x) = undef$, $mem(y) = undef$, $i =< 4 >$, and $o =<>$?

   Answer: There are two acceptable answers. If you take the argument to dsem to be a statement list, then: mem(x)= 4, mem(y) = 4, $i =<>$, $o =< 4 >$ If you take the argument to dsem to be a program, then: 4.

5. (10 points, Robert Sebesta) This is a pseudo-Pascal program. The only calls to procedures are the ones indicated. (E.g., for example, procedure C calls procedure B, while procedure B does not call any procedure.)

```
program BIGSUB
  procedure A;
    procedure B;
     begin
      ...
     end;
    procedure C;
     begin
      ...
      B;
      ...
     end;
    begin // A
     ...
     C;
     ...
    end; // A
  begin // BIGSUB
   ...
   A;
   ...
  end; // BIGSUB
```

(a) Draw the static nesting tree for this program. **Answer:** BIGSUB parent of A parent of both B and C

(b) Draw the stack with activation records for this program, showing static and dynamic links. Do *not* show return pointers. **Answer:** from the bottom, the stack has: BIGSUB, A, C, and B. Static link from C and B are to the top of A. Static link from A is to BIGSUB.

3 ML–23 points

1. What does the sml interactive system display in place of `type1`, `type2`, and `type3` in the session fragment copied below? (3 points)

```
opal.cs.sc.edu> sml
Standard ML of New Jersey, Version 110.0.3, January 30, 1998 [CM; autoload
enabled]
- false;
val it = false : type1
- 2.0 + 3.0;
val it = 5.0 : type2
- "five";
val it = "five" : type3
-
```

2. Recall that `op=` is the prefix version of `=`. What are type1, type2, and type3 in the following? (Be as general as possible!) (3 points)

```
- op=;
val it = fn : type1 * type2 -> type3
-
```

Answer:

```
- op=;
val it = fn : ''a * ''a -> bool
-
```

3. In the `if <test> then <then-part> else <else-part>` form, does ML evaluate `<test>`, `<then-part>`, and `<else-part>` by need (*lazy evaluation*) or by value (*eager evaluation*)? (2 points)

   Answer: By need (lazy evaluation).

4. Write a function `zip` that pairs corresponding members of two non-empty lists of equal size, e.g., `zip ([1,2,3], ["a","b","c"]) is [(1,"a"),(2,"b"),(3,"c")]`. Let `zip([],[])` be `[]`. Your function does not need to work if one of the argument lists is empty or if the arguments are lists of unequal size. (10 points)

   Answer:

```
fun zip (x::xs, y::ys) = (x,y) :: zip (xs, ys)
|   zip _ = [];

OR:

- fun zip (nil, nil) = nil
= |   zip (x :: xs, y :: ys) = (x,y) :: zip (xs, ys);
stdIn:1.1-20.51 Warning: match nonexhaustive
  (nil,nil) => ...
    (x :: xs,y :: ys) => ...

    GC #0.0.0.0.1.7:   (15 ms)

val zip = fn : 'a list * 'b list -> ('a * 'b) list
```

5. Write a function that multiplies an integer by 3. Call it times3. (2 points).

   Answer:

   fun times3 x = 3 * x;

6. Use simpleMap and the times3 function to multiply every element of an integer list by three (3 points).

   Answer:

   simpleMap (times3, L);

   where L is a list of integers.