CSE 428, Summer 1999. Final exam, 27 May 1999

The exam consists of 9 exercises, totalling 100 points. Write your answers directly on these pages. **Please remember to write your name and SSN at the top of each sheet.**

Exercise 1 (15 pts)

Consider the following grammar, defining a fragment of the language of Types in ML:

```
Type ::= TVar               (type variable)
      | Type list           (list type)
      | Type -> Type        (functional type)
      | (Type)

Ide ::= alpha | beta | gamma
```

1. (7 pts) Show that this grammar has two different kinds of ambiguities.

2. (8 pts) Define an equivalent and unambiguous grammar in which the arrow operator is right associative, and list operator has priority over the arrow.

Exercise 2 (9 pts)

Consider the following sequence of declarations in a Pascal-like language:

```
function f (x: integer)
   begin f := f(x) end;

procedure p (mode x,y: integer)
   begin if x=0 then print(0) else print(y) end;
```

where `mode` represents one of the following parameter-passing modes: value, reference, or name.

Say what is the result (error, infinite loop, or a printed value) of the following procedure call

```
p(0,f(0))
```

in case of

1. (3 pts) pass by value.

2. (3 pts) pass by reference.

3. (3 pts) pass by name.

Exercise 3 (9 pts)

For each of the following procedure declarations in C++, say whether its execution leaves dangling pointers and/or memory leaks. Assume that `p` is a global variable declared as a pointer to `int` and initialized to `NULL`.

1. (3 pts) `void foo() { int x; p = &x; p = NULL; }`

2. (3 pts) `void foo() { int *q; p = new int; q = p; p = NULL; }`

3. (3 pts) `void foo() { int *q; p = new int; q = p; delete q; }`

Exercise 4 (10 pts)

Consider the following class declarations in C++:

```
class C { protected: int x; ...};
class C1: public C { private: int x1; ...};
class C2: public C1 { private: int x2; ...};
class C3: public C1 { protected: int x3; ...};
```

1. (2 pts) Draw a tree-like structure illustrating the class hierarchy

2. For each of the following instructions, say whether it is allowed in the body of `C2` or not

   (a) (2 pts) `x = x+1;`

   (b) (2 pts) `x1 = x1+1;`

   (c) (2 pts) `x2 = x2+1;`

   (d) (2 pts) `x3 = x3+1;`

Exercise 5 (14 pts)

Define in Java a class that implements a data structure "Stack" suitable for a multi-threaded use. The class should have (at least) the methods `pop()`, `push()`, and `is_empty()`

Exercise 6 (10 pts)

Define in ML a datatype `'a three-tree`, where each node has three subtrees, possibly empty. Furthermore, define a function `non_zero: int three-tree -> int` such that `non_zero(t)` returns the number of all the nodes in `t` which have a value different from `0`.

Exercise 7 (10 pts)

Define a function

```
map: ('a -> 'b) -> 'a list -> 'b list
```

such that `map f [x_1,x_2,...,x_n] = [f(x_1),f(x_2),...,f(x_n)]`. Example: If `fact` is defined as the factorial function, then `map fact [1,4,3]` should give as result the list `[1,24,6]`.

Exercise 8 (9 pts)

For each of the following ML definition of a functyion `f`, say whether `f` is typeable, and, in the affirmative case, give is its most general type

1. (3 pts) `fun f g x = g x x;`

2. (3 pts) `fun f g x = g [x];`

3. (3 pts) `fun f g x = g g x;`

Exercise 9 (14 pts)

Define a Prolog predicate `occurrences(X,L,N)` which holds iff `N` is the number of occurrences of the element `X` in the list `L`. Example:

```
?- occurrences(a,[a,b,a,b,b,c],N).
N = 2;
no

?- occurrences(b,[a,b,a,b,b,c],N).
N = 3;
no

?- occurrences(d,[a,b,a,b,b,c],N).
N = 0;
no
```

If you use auxiliary predicates, please define them and explain in words what's their intended meaning.