

Homework 4 — Puzzles and programs in ML — assigned Monday 10 March, due Monday 24 March

Total number of points available on this homework is 180. Full credit is equivalent to 100 points.

Some problems in this homework set are puzzles that are best solved without the use of a computer; use an ML system only to verify your answers.

Some problems in this homework set are extensions of problems from the first three sets. Therefore you can reuse your code. Try, however, to make the code more elegant, and more in the ML style of programming; use the full wealth of purely functional ML programming constructs.

Reading assignment

Re-read Chapter 5 of *ML for the Working Programmer*.

4.1 Types (15pts)

Fill in the blanks so that the program is correct.

```
fun a [] = []  
  | a (h::t) = a t @ [h]  
  
fun b (x: ____): ____ = List.foldl (fn ((x: ____, y: ____), a: ____)  
                                     => a+x*y)  
  0  
  (ListPair.zip (x, a x))  
  
val (x: ____) = (b: ____) [1, 2]
```

We declare a single datatype `c`, which allows the following value declaration to be type-checked.

```
fun d (A (x,y)) = 1 + d y  
    | d (B (x,y)) = 1 + List.foldl (op+) 0 (List.map d y)  
  
    | d (C x) = List.foldl (op+) 0 (List.map (fn (_,x) => List.foldl (op+) 0 (List.map d x)) x)
```

What is the type of `d`? Reconstruct a possible declaration of `c`. Which aspects of the declaration are necessary, and which are arbitrary?

4.3 Lists (15pts)

Rewrite the following declaration of `s` using a list fold:

```
fun s nil = 0.0
  | s ((x,y)::t) = x*y + s t
```

4.4 Lists (20pts)

Fill in the blanks in the second declaration so that the meaning of the declared value is the same as in the first declaration.

[illegible]

4.5 Trees (20pts)

We declare a datatype for binary trees:

```
datatype 'a t = E | N of 'a t * 'a * 'a t
```

4.5.1 (5pts)

We then declare a value:

```
fun s E = 0
  | s (N (l,i,r)) = s l + i + s r
```

What is the type of s?

4.5.2 (15pts)

We declare a fold for this datatype:

```
fun tfold comb z E = z
  | tfold comb z (N (l,i,r)) =
    comb (i, tfold comb z l, tfold comb z r)
```

Declare a function s' using the fold `tfold`, such that s' has the same meaning as `s`.

4.6 Expressions using long integers (30pts)

This is a continuation of exercises 3.2 and 3.3.

Modify the expression evaluator to work with long integers (as defined in exercise 3.3).

Modify the expression parser to work with long integers. It should be able to read in decimal numerals of arbitrary length and produce long integers in radix 10.

For this exercise, you should not handle input in any radix other than 10.

4.7 ★★ Tic-tac-toe (30pts)

This is a continuation of exercise 3.4.

Recall that a strategy is a set of items of the form (i, o) , where i is of the form $[i_1, \dots, i_n]$. A pair of items (i, o) and (i', o') , such that i is a permutation of i' and $i \neq i'$ and $o \neq o'$, is said to have an *unresolvable conflict*. A strategy is said to be *feasible* if it contains no two items that have an unresolvable conflict.

Among the valid strategies, some are feasible. How many feasible strategies are there? Construct some feasible strategies.

How many strategies are there that are both feasible and favorable? Construct some.

4.8 ★ Transmission codes (30pts)

This is a continuation of exercises 1.6 and 3.5.

Construct a safe subset that is as large as possible, for the following values of the parameters: $K = 4, N = 18, D = 4$. Use the alphabet $\Sigma = \{0, 1, 2, 3\}$.

We modify the safety condition as follows. For two strings x and y to be distinguishable, they must still obey $H(x, y) \geq D$, but now they must also obey $H(x, y^C) \geq D$.

C is defined as follows, in ML syntax:

```
val D = 4
fun C (y: int list): int list =
  List.rev (List.map (fn a => D - 1 - a) y)
```

Points will be awarded according to the size of the safe subset computed, as follows. Among all submitted solutions, all sets will be checked, and all sizes will be sorted in descending order. The largest set will win 30 points, the second largest 25 points, the third largest 20 points; the remaining sets will win 19, 18, ..., 2, 1, 0 points, respectively.

How to turn in

Turn in your code by running

```
~roshan/handin your-file
```

on a regular UNM CS machine.

You should use whatever filename is appropriate in place of your-file. You can put multiple files on the command line, or even directories. Directories will have their entire contents handed in, so please be sure to clean out any cruft.