Homework 4 — ML core language — assigned Monday 16 February — due Tuesday 24 February

Reading assignment

Read Chapter 4 of *ML for the Working Programmer*.

4.1 Simple arithmetic expression evaluator: writing recursive functions over algebraic datatypes (20pts)

We can use the following declaration to introduce a language of simple arithmetic expressions:

```
datatype expr = Num of int
              | Add of expr * expr
              | Mul of expr * expr
```

For instance, the expression $(3+4)\times 5$ is represented by the ML term `Mul (Add (Num 3, Num 4), Num 5)`.

4.1.1 (10pts)

Write a function `eval`, with type `expr -> int`, which computes the arithmetic value of an expression, as a recursively defined function.

4.1.2 (10pts)

Write a *fold* function for the type `expr`, and then write `eval` in terms of *fold*.

4.2 A smarter arithmetic expression evaluator (30pts)

We can use the following data type declaration to introduce a language of simple arithmetic expressions, with variable names:

```
datatype expr = Num of int
              | Var of string
              | Let of {var: string, value: expr, body: expr}
              | Add of expr * expr
              | Sub of expr * expr
              | Mul of expr * expr
              | Div of expr * expr
type env = string -> int
exception Unbound of string
val emptyEnv: env = fn s => raise (Unbound s)
fun extendEnv oldEnv s n s' = if s' = s then n else oldEnv s'
exception ExprDivByZero
```

Write a function `evalInEnv`, with type `env -> expr -> int`, which computes the arithmetic value of an expression (which may have free variables) in a given environment (a mapping from variables to `int` values).

Then you can define:

```
fun eval e = evalInEnv emptyEnv e
```

so that `eval` evaluates closed expressions as in the preceding exercise.

4.3 Types (20pts)

The following is a correct block of ML code from which some pieces have been excised.

missing piece 1

```
type e = t * t

fun e (A x, A y) = x = y
  | e (B (x, v), B (y, w)) =
      x = y
      andalso (List.length v = List.length w)
      andalso List.foldr (fn (x, y) => e x andalso y)
                         true (ListPair.zip (v, w))
  | e _ = false

fun ee ((x, v), (y, w)) =
   e (x, y) andalso e (v, w)
```

missing piece 2

```
fun a [] e = e
  | a (v::w) e = c v (a w e)
```

Missing piece 1 is the declaration of `t`, and missing piece 2 is the declaration of `c`.

When we type in the whole block of code into SML/NJ, the following is reported (some lines omitted):

```
type e = t * t
val e = fn : t * t -> bool
val ee = fn : (t * t) * (t * t) -> bool
val a = fn : e list -> e list -> e list
```

1. (5pts) Give a complete possible declaration of `t`.
2. (5pts) What must be the type of `c`?
3. (5pts) Describe in words the purpose of the function `e`. Under what conditions on `x` does `e x` evaluate to `true`?
4. (5pts) Describe in words the purpose of the function `ee`. Under what conditions on `x` does `ee x` evaluate to `true`?

4.4 Drawing (30pts)

Write an ML function `generate_trefoil: unit -> string` that will compute a certain number of points of a trefoil figure $\rho = \cos 3\phi$ shown in Figure 1 and generate PostScript code that draws the trefoil. A sufficient number of points should be computed so that the figure looks smooth.

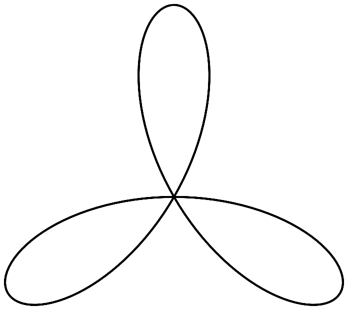

Figure 1: A trefoil, generated by the equation $\rho = \cos 3\phi$.

You should do this in two different ways:

4.4.1 "Statically computed" (15pts)

Only the PostScript commands of exercise 2.5 may be used. All trigonometric calculations must be performed within the ML function, i.e., ahead of "display time".

4.4.2 "Dynamically computed" (15pts)

No trigonometric calculations must be performed within the ML function. All trigonometric calculations must be performed by the generated PostScript code, i.e., at "display time".

How to turn in

Make sure that you have thoroughly tested your code, and include all your test runs!

Turn in your code by running

˜clint/handin your-file

on a regular UNM CS machine. You should use whatever filename is appropriate in place of your-file.

Include the following statement with your submission, signed and dated:
I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents' Policy Manual, including Section 4.8, Academic Dishonesty.