Computer Science & Engineering 505
Concepts of Programming Languages
Assignment 1: Programming in a Purely Functional Language
Sept 28, 1994
Due: October 5, 1994

To program in Miranda, you'll need to think in a different way than to program an imperative language. Be sure and not wait until the last minute to begin work on this assignment!

1. *You don't need to hand in anything for this first question.*

   Experiment a bit with Miranda. Try testing some of the built-in functions. Experiment with Miranda's type system: find the type of some constants, of some built-in functions such as member and map, and of some partially applied functions, e.g.

   map (const 1).

   Try to enter some ill-typed expressions, such as

   [1,2,[3],4]

   or

   member [1] "fred".

   Also try answering the following question, which was on a previous 505 final. (First try doing it by hand, then check your answer on the machine.)

   Suppose that the following Miranda script has been filed in.

   ```
   my_map2 f [] [] = []
   my_map2 f (a:as) (b:bs) = f a b : my_map2 f as bs
   rev f x y = f y x
   alligator x y = 3+y
   ```

   What is the result of evaluating the following Miranda expressions? (If there is a compile-time type error, or a run-time error, or a non-terminating computation, say so.) If the expression is followed by `::`, then give the type, instead of the value.

   (a) `alligator (1/0) (6/2)`
   (b) `alligator (6/2) (1/0)`
   (c) `alligator ::`
   (d) `my_map2 alligator [1..]  [10..]`
   (e) `my_map2 ::`
   (f) `my_map2 alligator ::`
   (g) `my_map2 (rev alligator) ::`
   (h) `(rev alligator) 10 20`
   (i) `rev ::`
   (j) `rev my_map2::`

2. Write a Miranda function average that computes the average of a list of numbers. It is an error to give this function an empty list.

3. Write a Miranda function repeat_numbers that returns the following infinite list: [1,2,2,3,3,3,4,4,4,4,5,5,5,5, ...] (there is one 1, two 2's, three 3's, etc).

4. Make up, write, and test your own Miranda script that uses infinite data structures in an interesting way. (If you can't think of anything interesting, make up a program that uses them in an uninteresting way. Some samples from previous years are in the directory ~borning/505/miranda/infinite on lynx.)

   If you have an example that you like, please be prepared to demo it in class on October 5. I'd like students to show some interesting examples they have done. Don't be shy! Also class participation of this sort will be taken into account when making up the final grades.

5. Write and test a Miranda function multiply that multiplies two polynomials in x and returns the result. The two polynomials should be represented as strings. For example:

   multiply "x**3 + x**2 + x + 1" "x - 1"
   returns
   "x**4 - 1"

   Here are some polynomial pairs to test your function on:

```
"x**3 + x**2 + x + 1"
"x - 1"

"- 3 x**3 + x + 5"
"0"

"x**3 + x - 1"
"-5"

"- 10 x**2 + 100 x + 5"
"x**9999 - x**7 - x**5 + x + 3"
```

   You can change the format of the input strings if you prefer something different.

   Hint: you will probably find it easier to use a different internal representation of a polynomial for doing the multiplication, for example, a list of (coefficient exponent) pairs. If you do this, the top-level function is

   ```
   multiply p q = tostring (polymult (tolist p) (tolist q))
   ```

   where tolist and tostring convert to and from the internal representation.