IEEE NITK

Functional Programming with Erlang

Assignment 5

Deadline – July 10, 2017 before 11:59 PM.

1. Currying in Erlang

   Currying is the partial application of a function. Consider add(X,Y) -> X+Y. Suppose we supply only one parameter (say add(5)) to it, the function returns a curried form of add (say add5(Y)). Now using that function, we can later apply the second parameter (add5(10) – returns 15).

   1. Implement curried form of add, subtract, multiply and divide functions
   2. This currying can be used to implement some interesting boolean functions.

      greaterThanX(A) -> returns a function that takes one argument that tells the relationship between A and this parameter. Example : gt3 = greaterThanX(3). gt3(4) -> true.

      Implement greaterThanX, lessThanX.
   3. The above curried functions can be used in the classic higher order functions like map, filter and fold

      Give 5 examples of such use with definitions

2. The derivate of a function f(x) with respect to variable x is defined as:

$$f'(x) \equiv \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

   Where *f* must be a continuous function. Write the function **derive**(*F*, *H*) that returns a new function that takes *X* as input, and which represents the derivate of *F* given a certain value for *H*. For example:

   f(x) = x^3

   f’(x) = $3x^2$

   f’(5) = 75

```
> F = fun (X) -> X * X * X end.
#Fun<erl_eval.6.49591080>
```

```
> DF = higherorder:derive(F, 0.001).
#Fun<higherorder.0.442056>
> DF(5).
75.0150
> DDF = higherorder:derive(DF, 0.001).
#Fun<higherorder.0.442056>
> DDF(5).
30.0060
```

3. Simpson's rule is a method for numeric integration:

$$\int_a^b f = \frac{h}{3}\left(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \cdots + 2y_{n-2} + 4y_{n-1} + y_n\right)$$

Where h = (b - a) ÷ n, for a given even positive integer n (if you increment the value of n you get a better approximation), and $y_k = f(a + k \times h)$. Write the function **integral**(A, B, N, F) that returns the value of the integral, using Simpson's rule.

For example:

$$\int_0^1 x^3 dx = \frac{1}{4}$$

would be written in Erlang as follows (with n = 10):

```
> higherorder:integral(0, 1, 10, fun (X) -> X * X * X end).
0.250000
```

The double integral:

$$\int_1^2 \int_3^4 xy \cdot dx \cdot dy = \frac{21}{4}$$

would be written in Erlang as follows (with n = 10):

```
> higherorder:integral(
    1, 2, 10,
    fun (X) -> higherorder:integral(
                3, 4, 10,
```

```
                fun (Y) -> X * Y end)
      end).
5.25000
```

4. Implement foldl using foldr

5. Research on foldr implementations using foldl and its limitations. Also research on universal nature of foldr.