Homework 3

February 12, 2020

1 Introduction

For this homework, you will implement the type class instances of a data type `Vec` defined below, which uses a list to represent a vector.

```
data Vec a = Vec [a]
```

You should implement the instances of `Vec` for the following type classes.

1. `Show` class, where you should implement the `show` function. The show function should print `Vec [1,2,3]` as `[1 2 3]`.
2. `Num` class, where you should implement the operators `(+), (-), (*)` and the functions `negate, abs, signum, fromInteger`. All but `fromInteger` perform elementwise operations on their operands while `fromInteger x` converts an integer x to a vector with infinite number of (`fromInteger x`).
3. `Fractional` class, where you should implement `(/), fromRational` functions. The division operator is also elementwise while `fromRational x` converts x to a vector with infinite number of (`fromRational x`).
4. `Floating` class, where you should implement `pi, exp, log, sin, cos, asin, acos, atan, sinh, cosh, asinh, acosh, atanh` functions. `pi` is a vector of infinite pi while other functions are elementwise operations on their operands.
5. `Foldable` class, where you should implement `foldr` function, which folds the list in the Vec.

You should also implement the following functions.

1. `pure`, which converts a constant x to a Vec of infinite number of x.
2. `realV`, which converts a Vec of numbers into a Vec of complex numbers where the imaginary parts are 0.
3. `imagV`, which converts a Vec of numbers into a Vec of complex numbers where the real parts are 0.

You should import `Data.Complex` to have access to `Complex` type. A complex value is written as `x :+ y` where `x` is the real part and `y` is the imaginary part.

2 Testing

You can test the implementation using the following `main`

```
main = do
         let v1 = Vec [1,2,3]
         let v2 = Vec [2,3,4]
         let v3 = Vec [-10,0,10]
         print $ v1 + v2
         print $ v1 - v2
         print $ v1 * v2
         print $ v1 / v2
         print $ negate v1
         print $ signum v3
         print $ abs v3
         print $ v1 + 10
         print $ v2 + 1.2

         print $ v1 + (pure $ sqrt 2)
         print $ realV v1
         print $ imagV v1
         print $ realV v1 + imagV v2
         print $ sin $ v1 * (pi / 2)
         print $ sum v1
```

After running the main, you should expect the following output

```
[3.0 5.0 7.0]
[-1.0 -1.0 -1.0]
[2.0 6.0 12.0]
[0.5 0.6666666666666666 0.75]
[-1.0 -2.0 -3.0]
[-1 0 1]
[10 0 10]
[11.0 12.0 13.0]
[3.2 4.2 5.2]
[2.414213562373095 3.414213562373095 4.414213562373095]
[1.0 :+ 0.0 2.0 :+ 0.0 3.0 :+ 0.0]
[0.0 :+ 1.0 0.0 :+ 2.0 0.0 :+ 3.0]
[1.0 :+ 2.0 2.0 :+ 3.0 3.0 :+ 4.0]
[1.0 1.2246063538223773e-16 -1.0]
6.0
```

3 Submission

Please write your solution in a file – `hwk3.hs` and submit it to the dropbox.