COMPSCI 1JC3

Introduction to Computational Thinking

Fall 2017

Assignment 5

Dr. William M. Farmer

McMaster University

Revised: November 18, 2017

The purpose of Assignment 5 is implement a practical higher-order function. The requirements for Assignment 5 are given below. Please submit Assignment 5 as a single `.hs` file to the Assignment 5 folder on Avenue under Assessments/Assignments. Assignment 5 is due **December 2, 2017 before noon.** Unlike the previous assignments, there is not an extra credit version of Assignment 5.

Although you are allowed to receive help from the instructional staff and other students, your submitted program must be your own work. Copying will be treated as academic dishonesty!

1 Background

Using the *trapezoidal rule*, the value of a definite integral

$$\int_a^b f(x)\,dx$$

can be approximated by the summation

$$\sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} * \frac{b-a}{n}$$

where

$$a = x_0 < x_1 < \cdots < x_n = b$$

and $x_i - x_{i-1} = (b-a)/n$ for all i with $1 \leq i \leq n$. Notice that

$$\frac{f(x_{i-1}) + f(x_i)}{2} * \frac{b-a}{n}$$

is the area of a trapezoid that approximates the area under the graph of f from x_{i-1} to x_i. The approximation becomes more accurate as the parameter n increases in value.

2 Assignment 5

The purpose of this assignment is to create a Haskell module for approximating the definite integral of a function $f : \mathbb{R} \to \mathbb{R}$.

2.1 Requirements

1. The name of your Haskell file is `Assign_5_`*`YourMacID`*`.hs` where *YourMacID* is your actual MacID.
2. Your name, MacID, the date, and "Assignment 5" are given in comments at the top of your file.
3. The first uncommented line of the file should be

   ```
   module DefiniteIntegral where
   ```

4. The file includes a function `definiteIntegral` of type

   ```
   Double -> Double -> (Double -> Double) -> Integer -> Double
   ```

   such that

   ```
   definiteIntegral a b g n
   ```

   computes an approximation to the definite integral

   $$\int_{\mathtt{a}}^{\mathtt{b}} f(x)\,dx$$

   using the trapezoidal rule with `n` partitions and using `g` to represent the function $f : \mathbb{R} \to \mathbb{R}$.

5. The file includes a function `circleArea` of type

   ```
   Double -> Double
   ```

   such that `circleArea r` computes the area of a circle of radius `r` using `definiteIntegral`.

6. The file includes a function `sphereVolume` of type

   ```
   Double -> Double
   ```

   such that `sphereVolume r` computes the volume of a sphere of radius `r` using `definiteIntegral`.

7. Your file can be imported into GHCi and all of your functions perform correctly.

2.2 Testing

Include in your file a test plan for all the functions mentioned above. The test plan must include at least three test cases for each function. Each test case should have following form:

`Function:` Name of the function being tested.
`Test Case Number:` The number of the test case.
`Input:` Inputs for function.
`Expected Output:` Expected output for the function.
`Actual Output:` Actual output for the function.

In addition, your test plan must include at least three QuickCheck cases for `definiteIntegral`. Each QuickCheck case should have following form:

`Function:` Name of the function being tested.
`Property:` Code defining the property to be tested by QuickCheck.
`Actual Test Result:` Pass or Fail.

The test plan should be at the bottom of your file in a comment region beginning with a `{-` line and ending with a `-}` line.