

COMPSCI 1JC3 (C01 and C02)
Introduction to Computational Thinking
Fall 2019

Assignment 1

Dr. William M. Farmer
McMaster University

Revised: September 12, 2019

The purpose of Assignment 1 is to write a program in Haskell that computes the solutions of a cubic equation. The requirements for Assignment 1 and for an extra credit assignment, Assignment 1 Extra Credit, are given below. You are required to do Assignment 1, but Assignment 1 Extra Credit is optional. Please submit Assignment 1 as a single `Assign_1.hs` file to the Assignment 1 folder on Avenue under Assessments/Assignments. If you choose to do Assignment 1 Extra Credit for extra marks, please submit it also as a single `Assign_1_ExtraCredit.hs` file to the Assignment 1 Extra Credit folder on Avenue in the same place. Both Assignment 1 and Assignment 1 Extra Credit are due **September 29, 2019 before midnight**. Assignment 1 is worth 4% of your final grade, while Assignment 1 Extra Credit is worth 2 extra percentage points.

Late submissions will not be accepted! So it is suggested that you submit a preliminary `Assign_1.hs` file well before the deadline so that your mark is not zero if, e.g., your computer fails at 11:50pm on September 30.

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

Recall from high school mathematics that a quadratic equation

$$ax^2 + bx + c = 0,$$

where a, b, c are real numbers with $a \neq 0$, has two solutions:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}.$$
$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

The solutions of the quadratic equation depend on the value of $b^2 - 4ac$ called the *discriminant* of the equation. If $b^2 - 4ac < 0$, the two solutions are non-real complex numbers; if $b^2 - 4ac = 0$, the two solutions are real numbers equal to each other; and if $b^2 - 4ac > 0$, the two solutions are distinct real numbers.

Less well known is that a cubic equation

$$ax^3 + bx^2 + cx + d = 0,$$

where a, b, c, d are real numbers with $a \neq 0$, has three solutions:

$$\begin{aligned} x_1 &= S + T - \frac{b}{3a}. \\ x_2 &= -\frac{S+T}{2} - \frac{b}{3a} + \frac{i\sqrt{3}}{2}(S-T). \\ x_3 &= -\frac{S+T}{2} - \frac{b}{3a} - \frac{i\sqrt{3}}{2}(S-T). \end{aligned}$$

where:

$$\begin{aligned} Q &= \frac{3ac - b^2}{9a^2}. \\ R &= \frac{9abc - 27a^2d - 2b^3}{54a^3}. \\ S &= \sqrt[3]{R + \sqrt{Q^3 + R^2}}. \\ T &= \sqrt[3]{R - \sqrt{Q^3 + R^2}}. \end{aligned}$$

The derivation of the solutions x_1 , x_2 , and x_3 is elegantly presented in the ProofWiki article “Cardano’s Formula” found at

https://proofwiki.org/wiki/Cardano's_Formula.

The solutions of the cubic equation depend on the value of $Q^3 + R^2$ called the *discriminant* of the equation as follows:

1. If $Q^3 + R^2 < 0$, then the three solutions are distinct real numbers. However, these solutions require complex number arithmetic to compute.
2. If $Q^3 + R^2 = 0$, then the three solutions are real numbers with $x_2 = x_3$. It is possible that $x_1 = x_2 = x_3$ in this case.
3. If $Q^3 + R^2 > 0$, then x_1 is a real solution, but x_2 and x_3 are two non-real complex solutions.

Historical note: The general solution for cubic equations was first devised by Niccolò Fontana Tartaglia (1499–1557) in 1530 and first published by Gerolamo Cardano (1501–1576) in 1545. The general solution for quartic equations (of degree four) was discovered by Lodovico Ferrari (1522–1565) in 1540. Niels Henrik Abel (1802–1829) gave the first complete proof in 1824 that general solutions do not exist for equations of degree five or greater.

2 Assignment 1

The purpose of this assignment is to compute — using only real number arithmetic — approximations of the real number solutions of a cubic equation with floating point coefficients.

2.1 Requirements

1. Download from Avenue `Assign1.Project.Template.zip` which contains the Stack project files for this assignment. Modify the `Assign_1.hs` in the `src` folder so that the following requirements are satisfied.
2. Your name, the date, and “Assignment 1” are in comments at the top of your file. `macid` is defined to be your MacID.
3. The file includes a function named `cubicQ` of type `Double -> Double -> Double -> Double` that computes Q from a , b , and c . The file also includes a function named `cubicR` of type `Double -> Double -> Double -> Double` that R computes from a , b , c , and d .
4. The file includes a function named `cubicDisc` of type `Double -> Double -> Double` that computes the discriminant from q and r .
5. The file includes a function named `cubicS` of type `Double -> Double -> Double` that computes S from q and r . The file also includes a function named `cubicT` of type `Double -> Double -> Double` that computes T from q and r . Using the `**` operator, define your own cube root function of type `Double -> Double` that finds cube roots of both negative and positive numbers. `cubicS` and `cubicT` should use only floating point operations and no complex number operations.
6. The file includes a function named `cubicRealSolutions` of type `Double -> Double -> Double -> Double -> [Double]` that computes a list of real solutions from a , b , c , and d using `cubicQ`, `cubicR`, `cubicDisc`, `cubicS`, and `cubicT`. The list is `[]` when the discriminant is negative, `[x_1, x_2, x_2]` when the discriminant is zero, and is `[x_1]` when the discriminant is positive. Here x_1 and x_2 are defined as above in terms of a , b , S , and T . (Note that the computation of the missing solutions require complex number arithmetic.)

7. If the functions `cubicS` and `cubicT` are implemented properly, they should be undefined when the discriminant, $Q^3 + R^2$, is negative since then the value of $\sqrt{Q^3 + R^2}$ is not a real number. That is, these functions should return `NaN` — which stands for “not a number” — when $Q^3 + R^2 < 0$.

For example, if the cubic equation is

$$x^3 - 3x = 0,$$

then by factoring it is clear that $x_1 = 0$, $x_2 = \sqrt{3}$, and $x_3 = -\sqrt{3}$. However, $Q^3 + R^2 = -1$, so

```
cubicRealSolutions 1 0 (-3) 0
```

should return `NaN` instead of 0. It is a useful exercise to compute x_1 , x_2 , and x_3 for this equation using complex number arithmetic.

8. Your file loads successfully into GHCi and all of your functions perform correctly.

2.2 Testing

You should test on your own all of the functions in the file you submit. You will be required to submit formal test plans for all subsequent assignments.

3 Assignment 1 Extra Credit

The purpose of this extra credit assignment is to compute approximations of the complex solutions of a cubic equation with floating point coefficients. This is a very challenging assignment; do not be discouraged if you cannot complete it.

3.1 Requirements

1. Add the Extra Credit functions to the `Assign_1.ExtraCredit.hs` file in the `src` folder (not `Assign_1.hs`). Modify this file so that the following requirements are satisfied.
2. Your name, the date, and “Assignment 1 Extra Credit” are in comments at the top of your file. `macid` is defined to be your MacID.
3. The file includes the same functions `cubicQ`, `cubicR`, and `cubicDisc` as in the file for Assignment 1.
4. The file includes two functions `cubicComplexS` and `cubicComplexT` of type `Double -> Double -> Complex Double` where `Complex Double` is the type of complex numbers whose real and imaginary parts are of type `Double`. You can either use a complex number type from a

Haskell library or define your own. When $Q^3 + R^2 \geq 0$, these two functions behave exactly like `cubicS` and `cubicT` except they return a real number represented as `Complex Double` value.

5. The file includes `cubicComplexSolutions` of type `Double -> Double -> Double -> Double -> [Complex Double]` that computes a list of the three solutions from a , b , c , and d using `cubicQ`, `cubicR`, `cubicDisc`, `cubicS`, `cubicT`. The list is $[x_1, x_2, x_3]$ where x_1, x_2, x_3 are values of type `Complex Double` defined as above in terms of a , b , S , and T .
6. Your file successfully loads into GHCi and all of your functions perform correctly.

3.2 Testing

Test on your own all of the functions in the file you submit. You will be required to submit formal test plans for all subsequent assignments.