Marius Furter
Institut für Mathematik
Universität Zürich

FS 2025

Logic and Foundation with Haskell

Exercise sheet 9

In this sheet you will explore rational and complex numbers in Haskell. You can import these types using

```
import Data.Ratio
import Data.Complex
```

Exercise 1. Newton's method provides an iterative scheme for finding roots of smooth functions. Starting from an initial guess x_0, the algorithm iteratively updates the estimate of the root according to

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)}$$

where f' denotes the derivative.

(a) Implement

```
newton :: (Rational -> Rational) -> (Rational -> Rational) ->
            Rational -> Int -> Rational
```

that takes a function f, its derivative f', a starting value x_0, along with a number of iterations N, and returns x_N.

(b) Implement

```
newton2 :: (Rational -> Rational) -> (Rational -> Rational) ->
    Rational -> Rational -> Rational
```

that replaces the number of iterations N with a tolerance ϵ. The function should continue iterating until $|x_{t+1} - x_t| < \epsilon$.

Exercise 2. The Mandelbrot set $M \subseteq \mathbb{C}$ consists of complex numbers c for which the function $f_c(z) = z^2 + c$ does not diverge to infinity when iterated starting at $z = 0$. Implement `mandelbrot :: Complex -> Bool` that (approximately) checks if a number c is contained in M.