DSLsofMath 2017: Assignment 2

Optimisation using Newton's method

This assignment is based on the lectures from weeks 3 and 4 (the *FunExp* type, *eval*, *derive*, *D*, tupling, homomorphisms, *FD*, *apply*, ...) so it pays off to work through those notes carefully.

1. The evaluation of the second derivative is given by

$$\mathit{eval}'' = \mathit{eval}' \circ \mathit{derive} = \mathit{eval} \circ \mathit{derive} \circ \mathit{derive}$$

   (a) Show that eval'' is not a homomorphism from *FunExp* to $\mathit{FunSem} = \mathbb{R} \to \mathbb{R}$.

   (b) Given the following types

$$\begin{array}{lll} \textbf{type}\ \mathit{Tri}\ a & = (a, a, a) & \\ \textbf{type}\ \mathit{TriFun}\ a = \mathit{Tri}\ (a \to a) & \text{-- } = (a \to a, a \to a, a \to a) \\ \textbf{type}\ \mathit{FunTri}\ a = a \to \mathit{Tri}\ a & \text{-- } = a \to (a, a, a) \end{array}$$

   Define instances of *Num*, *Fractional*, *Floating*, for *Tri a* and define a homomorphism *evalDD* from *FunExp* to *FunTri a* (for any type *a* in *Floating*). You don't need to prove that it is a homomorphism in this part.

   (c) Show that *evalDD* is a homomorphism for the case of multiplication.

2. Newton's method allows us to find zeros of a large class of functions in a given interval. The following description of Newton's method follows Bird and Wadler [1988], page 23:

$$\begin{array}{l}
\textbf{type}\ \mathbb{R} = \mathit{Double} \\
\mathit{newton} :: (\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R} \to \mathbb{R} \\
\mathit{newton}\ f\ \epsilon\ x = \textbf{if}\ \mathit{abs}\ \mathit{fx} < \epsilon \\
\qquad\qquad\qquad\quad \textbf{then}\ x \\
\qquad\qquad\qquad\quad \textbf{else if}\ \mathit{fx}' \neq 0\ \textbf{then}\ \mathit{newton}\ f\ \epsilon\ \mathit{next} \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \textbf{else}\ \ \mathit{newton}\ f\ \epsilon\ (x + \epsilon) \\
\quad \textbf{where}\ \mathit{fx} \quad = f\ x \\
\qquad\qquad\ \mathit{fx}' \quad = \mathit{undefined} \quad \text{-- } f'\ x\ \text{(derivative of } f \text{ at } x\text{)} \\
\qquad\qquad\ \mathit{next} = x - (\mathit{fx}\ /\ \mathit{fx}')
\end{array}$$

   (a) Implement Newton's method, using $\mathit{Tri}\ \mathbb{R} \to \mathit{Tri}\ \mathbb{R}$ for the type of the first argument. In other words, use the code above to implement

$$\mathit{newtonTri} :: (\mathit{Tri}\ \mathbb{R} \to \mathit{Tri}\ \mathbb{R}) \to \mathbb{R} \to \mathbb{R} \to \mathbb{R}$$

   in order to obtain the appropriate value for $f'\ x$.

(b) Test your implementation on the following functions:

```
test0 x = x^2                  -- one (double) zero, in 0
test1 x = x^2 − 1              -- two zeros, in −1 and 1
test2 = sin                    -- many, many zeros (n ∗ π)
test3 n x y = y^n − constTri x -- test3 n x specifies the nth root of x
    -- where constTri is the embedding of Const
```

For each of these functions, apply Newton's method to a number of starting points from a sensible interval. For example:

$$map\ (newton\ test1\ 0.001)\ [-2.0, -1.5\,..\,2.0]$$

but be aware that the method might not always converge!

For debugging is advisable to implement *newton* in terms of the minor variation *newtonList*:

```
newton f ε x = last (newtonList f ε x)
newtonList :: (Fractional a, Ord a) ⇒ (a → a) → a → a → [a]
newtonList f ε x = x : if ... then [] else ...
```

3. We can find the optima of a twice-differentiable function on an interval by finding the zeros of its derivative on that interval, and checking the second derivative. If x_0 is a zero of f', then

- if $f''\ x_0 < 0$, then x_0 is a maximum
- if $f''\ x_0 > 0$, then x_0 is a minimum
- if $f''\ x_0 = 0$, then, if $f''\ (x_0 - \epsilon) * f''\ (x_0 + \epsilon) < 0$ (i.e., f'' changes its sign in the neighbourhood of x_0), x_0 is an inflection point (not an optimum)
- otherwise, we don't know

Use Newton's method to find the optima of the test functions from point 2. That is, implement a function

$$optim :: (Tri\ \mathbb{R} \to Tri\ \mathbb{R}) \to \mathbb{R} \to \mathbb{R} \to Result\ \mathbb{R}$$

so that $optim\ f\ \epsilon\ x$ uses Newton's method to find a zero of f' starting from x. If y is the result (i.e. $f'\ y$ is within ϵ of 0), then check the second derivative, returning *Maximum* y if $f''\ y < 0$, *Minimum* y if $f''\ y > 0$, and *Dunno* y if $f'' = 0$.

As before, use several starting points.

Hint: you might want to modify the code you've written for Newton's method at point 2.

Formalities

Submission: Assignments are to be submitted via Canvas

Deadline: 2019-03-05

Grading: Discussions with each of the teams during one of the slots 2019-03-08.

References

R. Bird and P. Wadler. *Introduction to Functional Programming, 1988.* Prentice-Hall, Englewood Cliffs, NJ, 1988.