Informatik and Mathematical Modelling
DTU

15/11 07

Michael R. Hansen

02153 Declarative Programming
Programming Exercises: Lazy Lists

These exercises relates to slides and programs available on the homepage for the course.

Exercise 1

In this exercise you should use sequences of reals for approximating the square-root function using the Newton-Raphson method:

1. Declare a function `iterate`, where `iterate` $f\ x$ gives the infinite sequence
$$x, f(x), f(f(x)), \dots, f^k(x), \dots$$
2. Declare a function `inTollerance` *eps seq*, which gives the first occurrence x_k, where $|x_k - x_{k-1}| < eps$, in the sequence *seq* of real numbers $x_0, x_1, ..., x_{k-1}, x_k, ...$
3. The Newton-Raphson approximation method for computing the square root of a can be described as follows:
   - Generate a sequence of approximations by starting with a positive real number x_0 (you can choose 1.0), and generate further approximations according to the rule: $x_{i+1} = (a/x_i + x_i)/2$.
   - The approximation stops when the two successive approximations are close to each other within a given tolerance, where you, for example, can choose 10^{-6} (`1E~6`).
4. Declare a function which computes square roots according to Newton-Raphson's method.
5. How does it compare with the built-in square-root function?

Exercise 2

The exponential function can be approximated using Taylor's series:

$$e^x = \frac{1}{0!} + \frac{x^1}{1!} + \cdots + \frac{x^k}{k!} + \cdots$$

1. Declare a function for computing the exponential function according to this method. (Try first computations with small tolerance, e.g. 0.1.)
2. Solve the problem without use of factorial and power functions.