Programming Paradigms 2022
Session 12 : Lazy evaluation

Preparing for the session

Hans Hüttel

29 November 2022

Where nothing else is mentioned, chapters and page numbers refer to *Programming in Haskell*.

The video podcast

You can watch the podcast on YouTube via the course page on Moodle.

Tuesday 29 November 2022 – Lazy evaluation

Please read

- Chapter 15 of *Programming in Haskell*.

Learning goals for the session

- To be able to precisely explain the two main evaluation strategies: Call-by-value and call-by-name and the notion of a redex
- To be able to precisely explain the notion of lazy evaluation
- To understand how lazy evaluation allows Haskell to express infinite structures and finite computations on such structures
- To understand how strict application can be used in Haskell.

How you should prepare before we meet on Tuesday

Before we meet, watch the podcast and read the text. You can do this in any order you like. Also see if you can solve the following two small discussion problems. We will talk about them in class.

1. Give two different definitions (one using recursion, one not using recursion) of a function nsonly that takes as input a number n and gives us the infinite list consisting of $0n$, $1n$, $2n$, $3n$, ...
2. We can define the following:

   ```
   x = 1 : (map (1+) x)
   ```

   and then evaluate take 5 x.

   Oone might think that in fact the following happens:

   ```
   take 5 x
   = 1:2:map (+1) x
   = 1:2:map (+1) [1, 2]
   = 1:2:2:3:map (+1) x
   = 1:2:2:3:map (+1) [1, 2, 2, 3]
   = 1:2:2:3:2:3:3:4:map (+1) x
   ...
   ```

   Explain precisely why this is wrong. Saying that ”That is because the Haskell interpreter gives a different result” is not a valid answer – you have to provide an evaluation sequence as the ones presented in the text for today.

What happens on Tuesday?

When we meet, students that have been contacted by me who will present the solutions to the small discussion problems above.

Problems for Tuesday

For the plenary session we will solve and discuss a collection of problems that can be found on a separate page, available on the day of the session.