

Programming Paradigms 2022

Session 9 : Functors

Preparing for the session

Hans Hüttel

8 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 8 November 2022 at **2:30 pm** – Functors

Please read

- Sections 12.1 and 12.2 of *Programming in Haskell*.
- A simply typed λ -calculus in one page (available from the course page on Moodle)

Learning goals for the session

- To understand the notion of a functor
- To understand how familiar type constructs such as Maybe types and lists can be instances of the type class Functor
- To understand the notion of applicative functions
- To understand the [pure](#) and `<*>` operations and how they can be used in the applicative style of programming in Haskell
- To be able to apply the notions of functors and applicative functors for writing well-structured Haskell programs

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. An onion consists of a finite number of layers surrounding a core. In this problem, we let the core be a value. Figure 0.1 shows an onion with six layers and core "bingo". Below is a declaration of an algebraic datatype [Onion a](#) parameterized by the type [a](#).

```
data Onion a = Core a | Layer (Onion a)
```



Figure 0.1: An onion whose core is the value x_a

Define `Onion` as an instance of `Functor`. *Hint:* Be inspired by how the book shows how one can let the `Tree` type become an instance of `Functor`.

2. Check that the first two applicative laws at the top of page 163 hold for the `Maybe` type. *Hint:* Use the definitions of `pure` and `<*>` on page 160.

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.