Programming Paradigms 2024
Session 2 : First steps

Hans Hüttel

24 September 2024

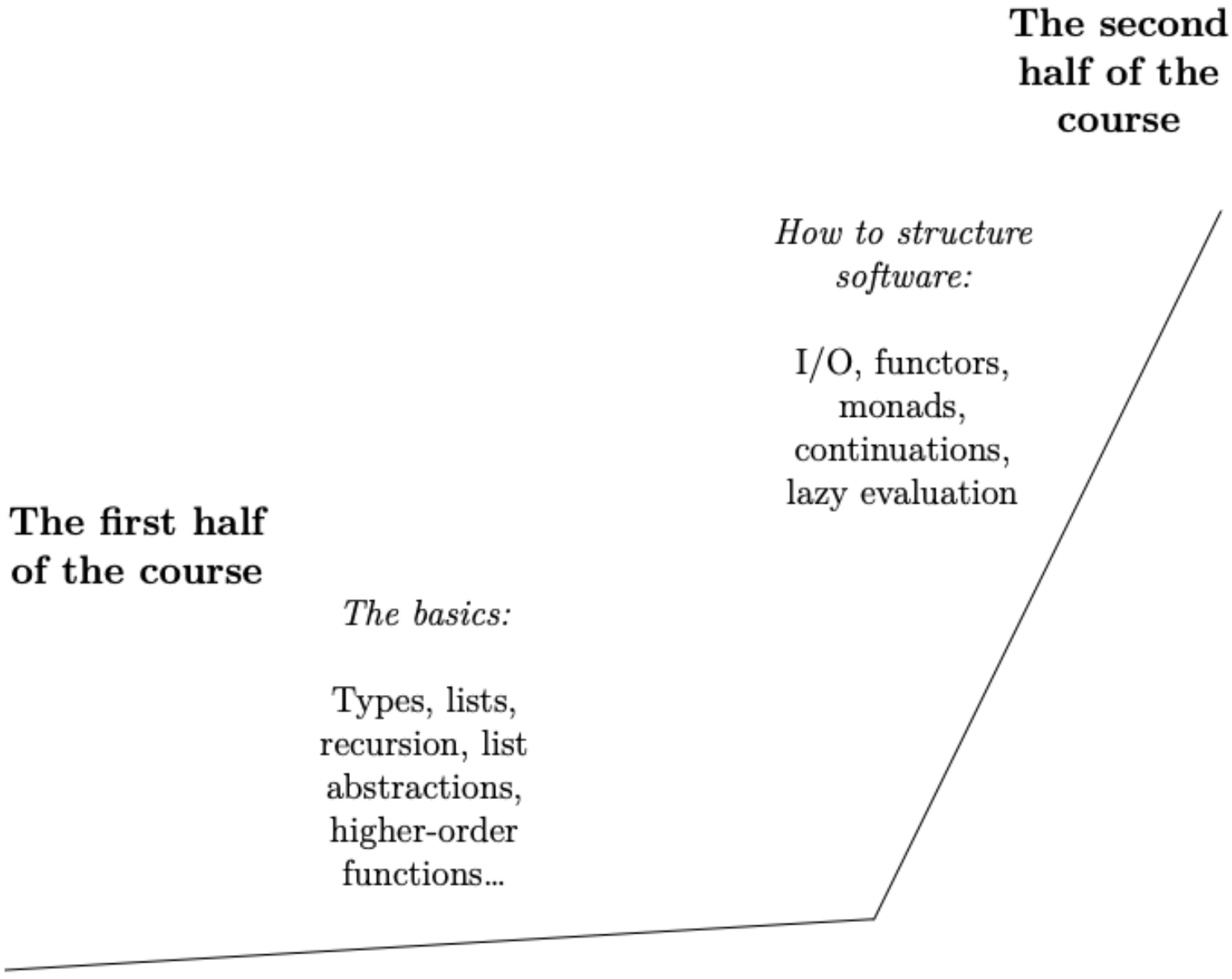
Our plan for today

1. How we make use of our time here.
2. The learning goals
3. The preparation problems
4. Something about libraries
5. A discussion
6. Problem no. 1
7. Break
8. Problem no. 2
9. Another discussion
10. Problem no. 3
11. If time allows: More problems at your own pace.
12. We evaluate today's session – please stay until the end!

A word of warning

This session is not very ambitious. That is completely deliberate – we need to make sure that the set-up will work such that the next sessions can make use of it in a good way. The sessions that follow will be more demanding.

The learning curve of this course



How we make use of our time here

We have four main activities:

- Talking about the preparation problems
- Discussions of typical mistakes and misunderstandings: Here is a piece of "interesting code". What has gone wrong in it? I will use the blackboard for this.
- Writing new code: Problem solving in pairs.
- A discussion of obstacles we encountered today.

Please take notes as we go along. You are participants, not spectators. The teaching assistants and I are always on hand.

Sharing code from solutions

There is a common folder for this session where you are allowed to upload solutions such that we can present them.
There will a folder of this kind for each of the sessions that follow.
Please remember to write your name(s) in the file.

Learning goals

The learning goals are

- To be able to explain the notion of a function in the functional programming paradigm in a precise way
- To understand the central points in the history of functional programming
- To be able to install the Glasgow Haskell compiler and use it with a simple programming environment
- To be able to write simple definitions in Haskell
- To be able to edit, load and use Haskell programs
- To understand and be able to apply simple aspects of Haskell syntax: Definitions, comments, the layout rule and **where** declarations in definitions.

Preparation problem – Trying out Haskell (5 minutes)

Load the program simple.hs available from the Moodle section about this session.

- Try to evaluate laengde myList.
- What do you think the result of sumtree myBigOak will be? Try to explain why.
- Then check your answer by asking Haskell.

Preparation problem – The second element (5 minutes)

Use the functions in Section 2.4 to define a function second that will, when given a list xs, return the second element of xs if it exists. As examples of what this function should do, we expect that

```
second [1,4,5,6]
```

will give us 4, and that

```
second ["some","bizarre","mango"]
```

will give us "bizarre". Show that your definition of second works for these examples of arguments. Then find two more examples of arguments and see what happens. Is your function a total function?

Something about libraries

Every problem that you will be asked to solve in this course can be solved by elementary means by using only the functions defined in the Haskell prelude. Do not waste your time looking for functions in obscure Haskell libraries that can solve the problem.

Remember what the time that we spend here is meant for: It is there to help you get experience such that you can reach the learning goals.

The problems are a setting in which you can get experience – they are not problems arising out of a need, so the "by any means necessary" slogan does not apply here.

A discussion (10 minutes)

A famous influencer posted a code snippet in Haskell to Instagram. Here it is:

```
x = 4
x = x + 4
```

The influencer had a lot of experience with C and was unable to figure out why such a simple piece of code would not work here. "It works fine in C – this does not make sense!" the influencer wrote.
Can you help the influencer without asking Haskell?

Problem 1 (15 minutes)

Define a function allbutsecond that will, when given a list list , return the list consists of all but the second element of the list. As examples of what this function should do, we expect that

```
allbutsecond [1,4,5,6]
```

should give us [1,5,6] and that

```
allbutsecond ["some","bizarre","mango"]
```

will give us ["some","mango"].

How can you become more certain that your solution is correct?

A break

Problem 2 (20 minutes)

Define a function midtover that will, when given a list xs of length n, return a pair of two lists (xs1,xs2) such that xs1 consists of the first $\lfloor \frac{n}{2} \rfloor$ elements from xs and such that xs2 consists of the remaining elements.

```
midtover [1,4,5,6]
```

should give us ([1,4],[5,6]) and that

```
midtover ["this","is","actually","a","
    fairly","long","list"]
```

will give us

```
(["this","is","actually"],["a","fairly","long
    ","list"])
```

How can you become more certain that your solution is correct?

Another discussion (10 minutes)

In the text for today we see that nothing is done to check if the argument of a function on integers actually is an integer.

Is this something we should worry about?

Problem 3 (15 minutes)

Something is wrong in the following tiny piece of code. But what is wrong? Explain. Then repair the code such that it works.

```
N = a `div` length xs
    where
      a = 10
     xs = [1,2,3,4,5]
```

Another discussion

Suppose you are asked to define functions iseven and isodd with types

```
iseven :: Integral a => a -> Bool

isodd :: Integral a => a -> Bool
```

such that iseven will take an integer and tell us if it is an even number and such that isodd will take an integer and tell us if it is an odd number. Somebody came up with the following solution.

```
iseven n = if n `div` 2 == 0 then True else False

isodd n = if iseven n then False else True
```

Do these functions work as intended? Is this an example of good coding style in Haskell?

Working at your own pace (pair programming)

There are further problems in the problem set, that you can work on during the rest of the session. See if you can solve one or more of them.

Evaluation

- What did you find difficult?
- What surprised you?
- What went well?
- What could we improve? (Such as how we can organize activities in the room.) Please bear in mind that is realistic for us all.
- Is there a particular problem that we ought to follow up on with a short video?