Functional Programming and Verification revision course

Jonas Hübotter

June 13, 2020

Organization

Wednesday, June 17th	recursion, list comprehensions, higher-order functions
Wednesday, June 24th	algebraic data types, type classes, abstract data types, type inference
Friday, June 26th	automated theorem proving
Wednesday, July 1st	IO, evaluation/reduction

Schedule

Session 1

- Haskell fundamentals
- recursion, guards, pattern matching
- list comprehensions
- QuickCheck
- polymorphism
- currying, partial application
- higher-order functions (incl. `fold`)

Session 2

- type classes
- algebraic data types (incl. `Maybe`)
- modules, abstract data types
- type inference

Schedule

Session 3

- structural induction
- case analysis
- extensionality
- computation induction

Session 4

- correctness
- I/O
- lazy evaluation, infinite data structures
- complexity and optimization

Structure

1. I give a brief introduction to a topic
2. we go over an example problem together
3. you work on problems
4. we compare results
5. I provide additional practice problems and further references

Structure

1. I give a brief introduction to a topic
2. we go over an example problem together
3. you work on problems
4. we compare results
5. I provide additional practice problems and further references

Important for you:

- ask questions!
- let me know when you want to spend more time on a topic

Structure

1. I give a brief introduction to a topic
2. we go over an example problem together
3. you work on problems
4. we compare results
5. I provide additional practice problems and further references

Important for you:

- ask questions!
- let me know when you want to spend more time on a topic

Slides, problems, and solutions can be found on GitHub:
`https://github.com/jonhue/teaching-fpv-rev`