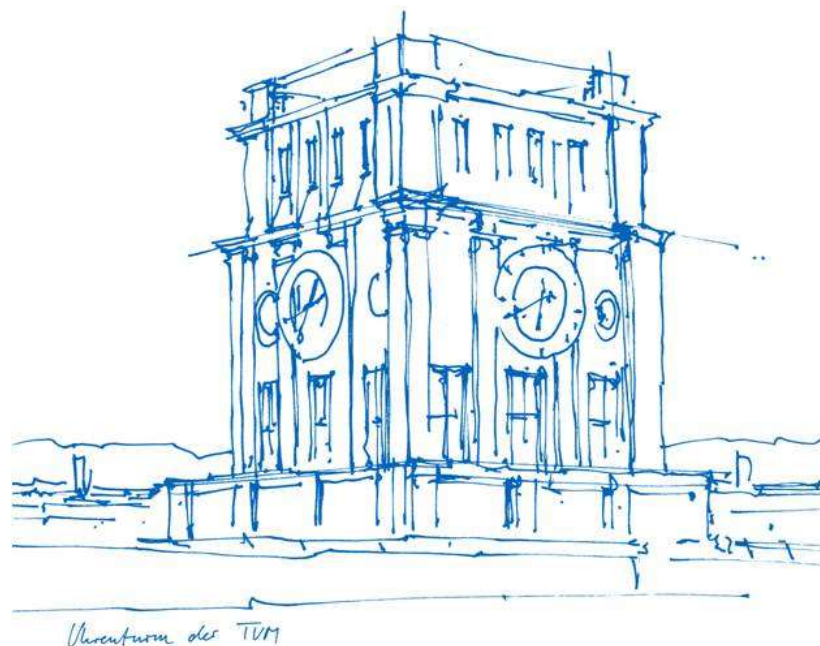


Revision course

Functional Programming and Verification

Jonas Hübotter



Organization

Wednesday, June 17th
2 pm - 6 pm

recursion,
list comprehensions,
higher-order functions

Wednesday, June 24th
2 pm – 6 pm

automated theorem proving

Friday, June 26th
3 pm – 7 pm

algebraic data types,
type classes,
type inference

Wednesday, July 1st
2 pm – 6 pm

IO,
evaluation/reduction

Schedule

Session 1 – functional programming and Haskell

- Haskell fundamentals
- QuickCheck
- lists, list comprehensions
- recursion and pattern matching
- currying/partial application
- higher-order functions (incl. `fold`)
- generic functions (incl. `Maybe`)

Session 2 – automated theorem proving

- structural induction
- case analysis
- extensionality
- computation induction

Schedule

Session 3 – algebraic data types, type classes, type inference

- algebraic data types
- modules and abstract data types
- type classes
- type inference

Session 4 – IO, evaluation/reduction

- correctness
- IO (monads)
- evaluation strategy (incl. infinite data structures)
- complexity and optimization

Structure

1. I give a brief introduction to the topic and underlying theory
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>

Setup

Does everyone have a laptop with a functional installation of Haskell?