COURSE TOPICS

- preliminaries natural deduction
- operational semantics ~~se~~ dynamic (small step)
- static semantics
- type soundness : preservation & progress
- type systems
- simply typed lambda calculus.
- type checking and inference
- polymorphism
- subtyping
- co-inductive types
- simulation & bisimulation
- conductive semantics of type safety.
- model checking
- automata and timed automata.
- temporal logics ~~model~~.
- model checking.
- specification of safety and liveness properties.
- tools: spin and upaal.

TEXT BOOKS

- Pierce et. al.
  Software Foundations

- Berard et. al.
  Systems & Software verification.

SOFTWARE FOUNDATIONS:

VOL1: LOGICAL FOUNDATIONS

classmate

Date 24.1.2020

Page 1

PREFACE

Sw foundations ⇒ mathematical underpinnings of reliable software.

- basic concepts of logic
- computer-assisted theorem proving
- Coq proof assistant
- functional programming
- operational semantics
- logics for reasoning about programs.
- static type systems.

- functional programming
- constructive logic
- Coq proof assistant

} this book does this.

techniques for improving sw reliability:

- extreme programming ... (project management)
- mvc, pub-sub, ... (design philosophy)
- OOP, AOP, FP, ... (prog. lang.)
- math. techniques to specify & reason about sw, + tools for helping.

(1) basic tools from Logic for making and justifying precise claims about programs.

(2) use of proof assistants to construct rigorous logical args.

(3) functional programming, both as a method of programming that simplifies reasoning about programs and as a bridge between programming and logic.

tools

① automated theorm provers SAT solvers, SMT solvers, modelchk.

② proof assistants.