

Software-Engineering Seminar, WS24 : Theorem Provers

WG Programming Languages

24.10.2024

Supervisors

- ▶ Cass Alexandru
- ▶ Alexander Dinges
- ▶ Sebastian Schloßer
- ▶ Michael Youssef

Goals

- ▶ Familiarize yourself with a Theorem Prover
- ▶ Gain a basic understanding of the theoretical framework it uses
- ▶ Learn how to formalize a proof / verify an algorithm in the system
- ▶ Gain experience in:
 - ▶ literature research
 - ▶ supervision meetings
 - ▶ writing an academic report
 - ▶ making and presenting slides

Tasks

- ▶ Teams of two participants
- ▶ Pick a Theorem Prover, then:
- ▶ Look into the theoretical framework / history of the theorem prover by doing literature research
- ▶ formalize a proof / verify an algorithm, preferably one you have encountered somewhere in your studies before
- ▶ write a report
 - ▶ Language: English (Bachelor: may be in German)
 - ▶ Use our Latex template
 - ▶ 10-15 pages (Bachelor: 7-15 pages), both team members contribute equally
- ▶ make slides and give a presentation
 - ▶ include demo of interacting with the tool
 - ▶ 25 minutes presentation (~12 minutes per team member)
 - ▶ about 10 minutes discussion and questions (be prepared to answer questions!)
 - ▶ each group will be assigned to ask questions / give feedback to another group!

Report Contents

- ▶ Theoretical framework
- ▶ Historical background
- ▶ Interaction with the tool
- ▶ How does the system allow for automation?
- ▶ Highlight unique features!
- ▶ Highlight interesting aspects of your formalization of a proof / program
- ▶ Highlight difficulties you had and how you overcame or worked around them

Theorem Provers

- ▶ Isabelle/HOL: 16.01.2025
 - ▶ A general purpose, semi-automated theorem prover based on Higher-Order Logic
- ▶ Liquid Haskell: 16.01.2025
 - ▶ An addition on top of Haskell that allows specification and verification of programs with the help of SMT solving
- ▶ Dependently typed functional programming languages / theorem provers based on dependent type theory:
 - ▶ Idris: 23.01.2025
 - ▶ Agda: 23.01.2025
 - ▶ Coq (Rocq): 30.01.2025
 - ▶ Lean: 30.01.2025
 - ▶ By the end of this seminar, and after attending the other groups' presentations, you will know how these differ!

Additional Rules

- ▶ Bachelors students:
 - ▶ Report may contain fewer details of your system's theoretical framework
 - ▶ Formalized proof / program may be simpler
- ▶ System choice:
 - ▶ Agda is not available to choose if you have heard / are hearing VFP
 - ▶ Coq is not available if you heard "Verification with the Coq Proof Assistant"

Schedule

- ▶ First draft of report: seven weeks before presentation (28.11/05.12./12.12.2024)
- ▶ Presentations
 - ▶ Thursday, 16.01.2025, 08:30 - 09:45 in 32-439
 - ▶ Thursday, 23.01.2025, 08:30 - 09:45 in 32-439
 - ▶ Thursday, 30.01.2025, 08:30 - 09:45 in 32-439
- ▶ Final report: 07.02.2025

All deadlines: End of the day 23:59.

Submissions: As source code + pdfs by email to your supervisor

How to Pass the Seminar

- ▶ Original text
- ▶ Stick to deadlines
- ▶ Attend final presentations
- ▶ Well written report
 - ▶ Conveys the key features
 - ▶ Comprehensible English
- ▶ Good presentation
 - ▶ Comprehensible story
 - ▶ Sticks to time limit
 - ▶ Able to answer questions
 - ▶ Not too much text on the slides
- ▶ Meet with your supervisor
- ▶ Use a spellchecker!