COMP4011/8011
Advanced Topics in
Formal Methods and Programming Languages

– Software Verification with Isabelle/HOL –

Peter Höfner

October 6, 2024

Section 20

Summary

The Big Picture

- Foundations & Principles
  - Intro, Lambda calculus, natural deduction
  - Higher Order Logic, Isar (part 1)
  - Term rewriting
- Proof & Specification Techniques
  - Inductively defined sets, rule induction
  - Datatype induction, primitive recursion
  - General recursive functions, termination proofs
  - Proof automation, Isar (part 2)
  - Hoare logic, proofs about programs, invariants

Lambda Calculus

- λ calculus syntax
- free variables, substitution
- β reduction
- α and η conversion
- β reduction is confluent
- λ calculus is very expressive (Turing complete)
- λ calculus results in an inconsistent logic

Simple Typed Lambda Calculus

- Simply typed lambda calculus: $\lambda^{\rightarrow}$
- Typing rules for $\lambda^{\rightarrow}$, type variables, type contexts
- β-reduction in $\lambda^{\rightarrow}$ satisfies subject reduction
- β-reduction in $\lambda^{\rightarrow}$ always terminates
- Types and terms in Isabelle

Proofs in Isabelle

- natural deduction rules for $\wedge$, $\vee$, $\longrightarrow$, $\neg$, iff...
- proof by assumption, by intro rule, elim rule
- safe and unsafe rules
- indent your proofs! (one space per subgoal)
- prefer implicit backtracking (chaining) or *rule_tac*, instead of *back*
- *prefer* and *defer*
- *oops* and *sorry*

Isar

- Isar style proofs
- proof, qed
- assumes, shows
- fix, obtain
- moreover, ultimately
- forward, backward
- mixing proof styles

Higher Order Logic (HOL)

- Defining HOL
- Higher Order Abstract Syntax
- Deriving proof rules
- More automation
- Equations and Term Rewriting

Termrewriting

- Equations and Term Rewriting
- Confluence and Termination of reduction systems
- Term Rewriting in Isabelle
- Conditional term rewriting
- Congruence rules
- AC rules
- More on confluence

Sets and Datatypes

- Sets
- Type Definitions
- Inductive Definitions

Induction

- Formal background of inductive definitions
- Definition by intersection
- Computation by iteration
- Formalisation in Isabelle
- Datatypes
- Primitive recursion
- Structural Induction

General Recursion

- General recursion with **fun**/**function**
- Induction over recursive functions
- How **fun** works
- Termination, partial functions, congruence rules

Sledgehammer and Co.

- sledgehammer
- nitpick
- quickcheck

IMP and Hoare Logic

- Syntax of a simple imperative language
- Operational semantics
- Program proof on operational semantics
- Hoare logic rules
- Soundness of Hoare logic
- Weakest precondition
- Verification conditions

State Monads

- Deep and shallow embeddings
- Isabelle records
- Nondeterministic State Monad with Failure
- Monadic Weakest Precondition Rules

AutoCorres and C Verification

- The automated proof method **wp**
- The C Parser and translating C into Simpl
- AutoCorres and translating Simpl into monadic form
- The option and exception monads

Exam

- 24h take-home exam

- Open book: can use any passive resource (books, slides, google, etc)
- **Not** allowed to ask for help from anyone
- **Not** allowed AI assistance for technical support (e.g. ChatGPT).
- starts 9am AEDT, Friday 1st Nov 2024, ends 9:00am AEDT, Saturday 2nd Nov 2024

- Should be doable in about 4-6 hours.
  The 24h are for flexibility not for you to stay awake actual 24 hours.
- Recommend to start early, finish the easy questions first.
- Take breaks. Don't forget to eat :-)
- If there are clarification questions, make **private** threads on Ed.