universität innsbruck

SS 2021

Program Verification

Part 7 – Summary and Outlook

René Thiemann

Department of Computer Science

Summary of Course

Summary by Parts

- part 2: extend first-order logic (of Logic course) by types
- part 3: define standard model for functional programs;
  derive axioms for induction, equality of constructors, etc.
- part 4: methods to ensure well-definedness of functional programs, including automated termination analysis
- part 5: framework for induction proofs and equational reasoning; specifications can be given via functional programs
- part 6: verification of imperative programs via Hoare-calculus;
  includes formal semantics and proof of soundness of calculus

Summary by Methodology

- inductively defined sets
- proofs by induction in various settings
  (by algorithm, by data-structure, by inductively defined set, ...)
- proofs by invariants
- verification by refinement
  - prove soundness of (abstract) pseudo-code against specification
  - prove that concrete code is valid implementation pseudo-code
- integrating external tools and certification
  termination proofs via SMT-solver, logic-solver for Hoare-calculus
- development of paper-verified interpreter for functional programs written in Haskell
  - checks well-definedness of input (missing: termination analyser)
  - algorithms for these checks have been verified
  - verified implementation of one-step evaluation $\hookrightarrow$

Summary on Organisation

- relatively new course
- feedback is highly welcome
  (via mail, anonymous via PV-website, via evaluation, etc.)
  - content + structure
  - feasibility
  - typos
  - ...

Outlook

Related Courses

- backend-solvers: decision procedures, automated theorem proving
- core evaluation mechanism: (selected topics in) term rewriting
- program verification with tool support: interactive theorem proving
- more automation: program analysis

Related Bachelor Thesis Topics

- currently running
  - A Verified Decision Procedure for Termination of Right-Ground Term Rewrite Systems
  - Fast (and Verified) Multiset-Comparisons
  - Formalizing NP-hardness of SAT
- new topics will appear, or you can contact me with your own ideas on program verification related topics

Thank you for Watching!