Specification and Verification in Higher-order Logic
Lecture SS 2010

Prof. Dr. A. Poetzsch-Heffter

AG Softwaretechnik
TU Kaiserslautern
Overall structure

1. Introduction
2. Functional programming and specification
3. Language and semantical aspects of higher-order logic
4. Proof system for higher-order logic
5. Sets, functions, relations, and fixpoints
6. Verifying functions
7. Inductively defined sets
8. Specification of programming language semantics
9. Program verification and programming logic
Chapter 1: Introduction

1. Terminology: Specification, verification, logic
2. Language: Syntax and semantics
3. Proof systems
   3.1 Hilbert style proof systems
   3.2 Proof system for natural deduction

» notes on black board
» slides_01: 1-24
Chapter 2: Functional programming and specification

1. Functional programming in ML
2. A simple theorem prover: Structure and unification
3. Functional specification in HOL

» slides_02: 1-65
» slides_02: 77-101
» Chapter 2 and 3 of Isabelle/HOL Tutorial
Chapter 3: Language and semantical aspects of HOL

1. Introduction to higher-order logic
2. Foundation of higher-order logic
3. Conservative extension of theories

» hol_introSllind: 3-12
» hol_foundations06: 1-30
» hol_conservative_extensions06: 1-25
Chapter 4: Proof system for HOL

1. Formulas, sequents, and rules revisited
2. Application of rules
3. Fundamental rules of Isabelle/HOL
4. An overview of theory Main
   4.1 The structure of theory Main
   4.2 Set construction in Isabelle/HOL
   4.3 Natural numbers in Isabelle/HOL
5. Rewriting and simplification
6. Case analysis and structural induction
7. Proof automation
8. More proof methods

» lecture script for Chapter 4
» slides of Sessions 2, 3.1, 3.2, and 4 & 5 by T. Nipkow
» Chapter 5 of Isabelle/HOL Tutorial til page 99
Chapter 5: Sets, functions, relations, and fixpoints

1. Sets
2. Functions
3. Relations
4. Well-founded relations
5. Fixpoints

» Chapter 6 of Isabelle/HOL Tutorial til page 118
» lecture script for Chapter 5
Chapter 6: Verifying functions

1. Conceptual aspects
2. Case study: Gcd
3. Case study: Quicksort – Shallow embedding of algorithms

» lecture script for Chapter 6
» theories for Gcd and Quicksort
Chapter 7: Inductively defined sets

1. Defining sets inductively
2. Specification of transitions systems
   2.1 Transition systems
   2.2 Modeling: Case study Elevator
   2.3 Reasoning about finite transition systems

» Section 7.1 of Isabelle/HOL Tutorial
» slides of Sessions 6.1 T. Nipkow
» lecture script for Chapter 7
» theory for Elevator
Chapter 8: Specification of programming language semantics

1. Introduction to programming language semantics
2. Techniques to express semantics
   2.1 Natural semantics / big step semantics
   2.2 Structured operational semantics / small step semantics
   2.3 Denotational semantics
3. Formalizing semantics in HOL

» slides about operational semantics by P. Müller
» lecture script for Chapter 8
» theory for while-language
Chapter 9: Program verification and programming logic

1. Hoare logic
2. Program verification based on language semantics
3. Program verification with Hoare logic
4. Soundness of Hoare logic

» lecture script for Chapter 9
» theory for while-language
» theory for Hoare logic