Chapter 5

Verifying Functions

Overview of Chapter

Section 5.1

Introduction

Motivation

Verifying properties of functions

Verifying properties of functions is a fundamental task in theorem proving and software engineering:

- Functions allow to express recursive algorithms
- Functions can be used to model systems (e.g., a compiler is essentially a function)
- Functions are used to specify input/output behavior of procedures, so called IO-properties
- Verifying recursive functions is related to terminiation proofs

Specification

Kinds of specifications:

- specification = model + properties
  $\Longrightarrow$ verify that model has the properties

or

- specification = model_1 + model_2 + relationship
  $\Longrightarrow$ verify that models are in the relationship

Here:

specification = function definition + property of function

Basic proof techniques

Verify:

- well-definedness of function by:
  - structural induction according to parameter types
  - more general: well-founded ordering on parameter space: "show that parameters get smaller"
- property of defined function:
  - structural induction according to parameter types
  - in general, proof technique depends on properties

Discussion

Verification

- checks for consistency of models and properties
  - models may not reflect what designer/programmer had in mind
  - properties may not reflect what designer/programmer had in mind
- works for the full parameter space (in contrast to testing)
- discovers also "pathological" problems
- uses redundancy to find errors
- helps to improve the descriptions

Formal verification avoids misunderstanding, allows using tools, and avoids errors in proofs.

Section 5.2

Case study: Greatest common devisor