Semantica dei linguaggi di programmazione

Introduzione al corso

Short introduction

Language semantics (and proof assistants)

- to describe the behaviour of programs in a formal and rigorous way,
- first step in formally in proving correctness, or other properties, of programs.

There exists several approaches

- operational semantics
- denotational semantics
- axiomatic semantics (Hoare logic)

Proof assistants

Problem: the complexity of the description,
Software tool are necessary for managing, analysing the formal descriptions of programs
There are several of these tools:

- proof automation
  - Interpretazione astratta, Ragionamento automatico
- proof assistants (our choice)
  - tools that help the development of formal proofs
  - guide in the construction of the proofs
  - check correctness
  - automatically builds parts of the proofs

Textbook

Software foundations

- Benjamin Pierce and others
- A series (5) textbooks, collectively written
- Available on line: https://softwarefoundations.cis.upenn.edu/
- A broad presentation of different approaches to obtain reliable software using formal methods.
- We present some parts of this material.

Coq

- Textbook based on the proof assistant Coq
- Every presented subject is also formally written in Coq
  - Many exercises.
- The first part of the course will be an introduction to proof Coq
  - Vol 1. Logical foundations of the series
- Still to decide which other material to present
  mainly from Vol 2 and 3.

Main objectives of the course

- Proof assistants and their underline theory:
  - a different formalisation of logic
  - a new class of programming languages, p. l. with dependent types
    - connection between logic and programming languages, Curry-Howard isomorphism
  - introduction and exercises in formal reasoning
- Semantics of programming languages
  - introduction to the subject
  - show how semantics can be used to obtain reliable software.
- Reliability of software,
  - the broad aim
  - a difficult task

Relation with other courses:

Triennale

- Logica Matematica, but with a different formalizaion of logic (predicate calculus and natural deduction)
- Linguaggi di programmazione, mainly for the functional programming part.

Magistrale

- Analisi e verifica mediante interpretazione astratta, different formalism, more emphasis on the automatic part.
- Metodi formali per l'informatica, operational semantics
- Logica per l'informatica, we treat different subjects
- Ragionamento automatico, different formalism, more emphasis on the automatic part.

Exams

- a set of simple exercises to solve at during the course
- a larger final exercises, small project,
- an oral exam, on appointment.the program

Tools

- Coq, available from the Coq home page
- An IDE for interacting with Coq
  - Proof General is an Emacs-based IDE.
  - CoqIDE is a simpler stand-alone IDE distributed with Coq

Proof assistant

Tools for constructing proofs

- Automated theorem provers
  - “push-button” operation:
  - capabilities limited to specific domains, improved in recent years, used in a multitude of settings
  - examples of such tools:
    - SAT solvers (proposition calculus),
    - SMT solvers (firs order theories ,
    - model checkers.
- Proof assistants
  - hybrid tools:
  - automate some aspects of building proofs
  - depending on human guidance for more difficult aspects
  - examples: Agda, F*, Isabelle, Twelf, ACL2, PVS, Coq
  - a variety of approaches,
    some of them can be seen also as programming languages: Agda, F*, Idris

Coq

It is based on a relatively simple calculus the Calculus of Constructions

- a sort of functional programming languages with
  - a richer set of types, dependent types
    one can specified that an expression is:
    - a natural number
    - a prime
    - a vector having length 5
    - an function ordering a vector
  - types can be seen a propositions, or predicate
    - an element of a proposition type, is also a proof that proposition is true
  - Curry-Howard correspondence, in some context:
    - types correspond to propositions
    - elements correspond to proofs.

Coq

Calculus of Constructions is the kernel of the systems, above it

- an environment for interactive development of formal reasoning
- a large library of
  - definitions
  - lemmas
  - tactics for automatically proof simple propositions
- a special-purpose programming language for defining new tactics

Coq is then use

- to formally proof the correctness of complex software
  - CompCert, a fully-verified optimised compiler for C
- to build environment for reasoning about the security of software
  - CertiCrypt cryptographic algorithms
- to formally check mathematical proofs
  - 4-colour theorem
  - Voevodsky