Formale Methoden der Softwaretechnik
Formal methods of software engineering

Till Mossakowski, Christoph Lüth

SoSe 2011

Overview

- Why formal methods?
- Overview of the course
- Literature
- “Scheinkriterien”

Therac-25

- New *linear accelerator* in radiation therapy
  - Computer controlled (PDP-11, assembler)
- Three *letal* accidents (1985– 1987)
  - To high radiation dose (4000 – 20000 rad, letal: 1000 rad)
- Problem: *bug in the software*
  - One programmer (five years)
  - All in assembly language, no operating system
  - Programmer simultaneous tester (quality control)

Ariane-5

The Vasa

Formal specification and verification

- a formal model is based on a *logical system* with *mathematical semantics*
- *properties* (e.g. safety) of software and hardware can be *specified*
- correctness of software and hardware can be *proved* (with respect to the specification)
- generation of *test cases* from specifications
- Examples:
  - NASA specifies physical units (after loss of Mars probe)
  - Pentium 4 arithmetic completely specified and verified with higher-order logic (after pentium bug)
  - properties of telephone systems specified in ATL

Learning targets

- deeper knowledge of the methods of formal (i.e. logic-based) specification and verification of systems,
- comprehension of the therefore needed methods of analysis and proof, especially formal calculi and their algorithms,
- use of formal modeling and verification tools,
- ability to select appropriate methods and tools for practical purposes.

Topics

- propositional logics, SAT solvers, DPLL
- SMT solvers
- First-order logic, datatypes, first-order calculi and (automated) provers
- Higher-order logic, interactive provers (Isabelle/HOL)
- fragments of C and their specification languages
- Hoare logic (specification of pre and post conditions)
- embedded systems, UPAAL

Tools for formal methods

SAT solvers minisat, zChaff

first-order provers SPASS, Vampire, Darwin, MathServer

higher-order provers Isabelle, Leo-II

Verification of C SAMS prover, Frama-C

SMT solvers Z3, etc.

Tool integration Heterogeneous Tool Set (Hets)

We will provide these tools as Ubuntu package and as a virtual machine

Organisation

Christoph Lüth
christoph.lueth@dfki.de
Cartesium 2.43, Tel. 64223

Till Mossakowski
till.mossakowski@dfki.de
Cartesium 2.51, Tel. 64226

- Monday 14:00 - 16:00 MZH 1460
- Thursday 16:00 - 18:00 MZH 1380
- bring your Laptops with you!
- Web:
  www.informatik.uni-bremen.de/~cxl/lehre/fmst.ss11/

Scheinkriterien

- exercise sheets
- and: presentation of solutions to the class, and/or oral exam ("Fachgespräch")