DD2452 Formal Methods

Introductory Lecture

Lecture Outline

1. The lecturer
2. Introduction to Formal Methods
3. Course syllabus
4. Course objectives
5. Course organization

1. Lecturer

- Name: Dilian Gurov
- E-mail: dilian@csc.kth.se
- Phone: 08-790 81 98 (office)
- Visiting address:
  Osquars backe 2, floor 4, room 4417

- Research interests:
  – Analysis of program behaviour
  – Correctness: logics, compositionality

2. Formal Methods

- **Formal methods**:
  collection of formal notations and techniques (i.e. based on discrete mathematics and mathematical logic) for modelling and analysis of program behaviour.
- **Common goal**:
  The design of *correct* systems.

Why *Formal* Methods?

- Only formal methods can capture correctness *precisely*. Basis for *tools*.
- But: formal techniques are *expensive*
- Most needed for:
  – safety-critical systems
  – commercially-critical systems (security)
- Most succesful for: ”small” systems
  – embedded systems
  – communication protocols

Formal Verification

- **Two possibilities**:
  – correctness by design: transformation
  – establishing correctness: *verification*
- **Three ingredients**:

– model	M	M	ψ
– specification	S	ϕ	ϕ
– verification	$\models M \approx S$	$M \models \phi$	$\models \psi \rightarrow \phi$

Main Challenge

- Real systems are *large* and *complex*
- Only restricted problems are *decidable*, i.e. algorithmically solvable
- Scalablity requires efficient solutions, i.e. *tractability*

What do we do?

Abstraction

- Key technique for handling complexity! separation of concerns; buidling models
- Key questions:
  - What are the important properties?
  - What is the ”right” abstraction level?
  - How does the abstract model relate to the concrete system? (”modelling gap”)
  - How to interpret the results of the analysis?

Main Schools

- Hoare Logic
- Process Algebra
- Model Checking
- Theorem Proving

We will focus on the first and the third of these approaches.

3. Course Syllabus

Part I. Hoare Logic and Program Verification

Goal: Correctness of data manipulation
Models: Source code (Java)
Specs: Hoare Logic (JML)
Method: Proof tableaux; VCG + ATP
Tool: ESC/Java2

Course Syllabus

Part II. Temporal Logic and Model Checking

Goal: Correctness of state sequences
Models: Promela
Specs: Temporal logic formulas (LTL)
Method: Model checking
Tool: SPIN

4. Course Objectives

- **Aim**: provide working familiarity with main methods and tools, in theory and in practice.
- **Grading**: to pass the course, a student has to demonstrate the ability to apply the methods discussed in the course; for the highest grade he/she has also to be proficient in the theoretical foundations of these methods.

Course Objectives

After the course, you should be able to:

1. Independently select a suitable modelling approach for a given problem;
2. Argue informally and formally for the soundness and limitations of the chosen approach;
3. Identify, specify and verify important system properties using suitable automated tools;
4. Correctly interpret and evaluate the results of the analysis.

5. Course Organization

- 17 one-hour lectures/tutorials: mixed
- 2 lab sessions: for reporting only!
- 1 written exam: 5 hours, open-book
- Course **web page**:
  www.csc.kth.se/DD2452/form09/
- Course board: sv. ”kursnämnd”
  volunteers?

Course Literature

- Course **book**:
  ”*Logic in Computer Science*”
  by Huth and Ryan (see Kårbokhandeln)
- Additional material: on the web page
  don’t print without need!

Labs

- **ESC/Java2 lab**:
  correctness of an ADT implementation
- **SPIN lab**:
  correctness of a communication protocol

Tools

- **ESC/Java2**:
  ```
  module add escjava
  escjava2
  ```
- **SPIN lab**:
  ```
  module add spin
  xspin
  ```