MAP-i 2007/08

Program Semantics, Verification, and Construction
Approaches to Correct Software

DCCFCUP / DIUM

Motivation…

"Despite 50 years of progress, the software industry remains years – perhaps decades – short of the mature engineering discipline required to meet the needs of an information-age society."

W. Gibbs, in Trends in Computing: Software's Chronic Crisis
Scientific American, 1994

Formal Methods

- Specification and Models:
  algebraic; abstract state machines; automata; set-theoretical;
  declarative

- Theorem Proving and Proof Assistants

- Model Checking

- Hoare Logic / Weakest-precondition Calculi

- Integrated Design Methods (B)

- Refinement and Derivation of Implementations

Goals: Not a Formal Methods Course!

- To cover the fundamental theoretical bases of Programming Languages (Part I)

- To give an overview of some modern rigorous methods for the design of reliable software systems

  - The Logic-based approach:
    Program Verification Methods; Design by Contract (Part II)

  - The Algebraic approach:
    Program Construction and Calculation (Part III)

- NOT COVERED: Complexity theory, Computability

Recent Changes

- ISO-15408 (Common Criteria). Highest Evaluation Assurance levels (EAL5 – EAL7) require public certification and the use of formal methods
- Advances in the theory: correct-by-construction school of programming
- New safety requirements: code mobility ; memory confinement ; component outsourcing. GLOBAL COMPUTING
- New open standards: the JML effort; new practical and easy-to-use tools bring certification to the common programmer: Spec#, ESC/Java
- New pragmatic techniques and architectures for the certification of code: design-by-contract; proof-carrying code; certifying compilation
- Many recent success stories in industry

The Team and Context

- José Bacelar Almeida (DIUM)
- Sabine Broda (DCCFCUP)
- Luís Damas (DCCFCUP)
- Mário Florido (DCCFCUP)
- Maria João Frade (DIUM)
- Nelma Moreira (DCCFCUP)
- José Nuno Oliveira (DIUM)
- Jorge Sousa Pinto (DIUM)

- Ongoing collaboration between DCCFCUP and DIUM
- Recently approved joint project "RESCUE" (REliable and Safe Code execUtion for Embedded systems)
- Team members are or have been members of major EC-funded networks in the area: APPSEM II, TYPES

Part I: Overview of Foundations (15 hours)

1. Intuitionistic logic

2. Natural deduction

3. λ-calculus (terms, reduction, the Church-Rosser Theorem)

4. Simple Types (Church versus Curry typing, normalization, extensions)

5. The Curry-Howard isomorphism

6. Introduction to operational semantics

7. Domain theory (complete partial orders, continuous functions)

8. Denotational semantics

Part II: Program Verification (18 hours)

1. Dependent Types

2. Type-based proof assistants and interactive proof development

3. Verification of the correctness of functional programs

4. Axiomatic semantics of imperative programs: Hoare Logic

5. Weakest-precondition calculi and Design-by-contract: the JML effort

6. Tool support: verification condition generators; extended static checkers

7. Survey of alternative approaches to program verification:
   Certifying compilation and proof-carrying code

Part III: Program Construction (15 hours)

1. Introduction to the mathematics of program construction

2. Description versus calculation

3. The Point-free (PF) transform

4. Universal properties and Galois connections;
   Reynolds' relation and the free-theorem of polymorphism

5. Reasoning by PF-calculation
   Data-level calculation: representing and abstracting data models.

6. Inductive program calculation

7. Open issues and hot topics in the mathematics of program construction

Methods and Assessment

- Lectures
- Tool Demonstrations / Case studies
- Exams
- Research Assignments
  - talks (read papers)
  - practical assignments