CS3202: Logic, Specification and Verification

CS3202-LSV 2006–07

cs3202.lec@cs.st-andrews.ac.uk

Dr. James McKinna, Rm 1.03
Dr. Stéphane Lengrand, Rm. 1.02
Lecture 0 (06/02/2007):
Module overview
Module Overview, 0

• This is a module about
  
  – *logic*: the ‘science’ of reasoning
  
  – *specification*: the ‘art’ of modelling systems
  
  – and *verification*: proving properties of (models of) systems
Logic

- the ‘science’ of reasoning
  - from hypotheses to conclusions: consequence relations
  - language: syntax, including rules of inference
  - models: semantics
  - do the models support the language?: interpretations and soundness
  - does the language capture the models: expressivity and completeness
  - can we mechanize reasoning: effectiveness (computability and complexity)
Specification

- the ‘art’ of modelling systems

- **language**:
  - datatypes
  - operations
  - properties (static)
  - behaviour (dynamic)

- **abstraction**:
  - capture only those of interest
  - avoid implementation details

- have we got it right? **validation**
Verification

• proving properties of (models of) systems

• build a model $\mathcal{M}$

• satisfaction: does the property, $\phi$, hold in the model?

$\mathcal{M} \models \phi$
Verification, II

- factor the problem

- does the property follow from (simpler) hypotheses?
  \[ \phi_1, \ldots, \phi_n \vdash \phi \]

- does the model support the hypotheses?
  \[ M \models \phi_1, \ldots M \models \phi_n \]

- such steps might in general be interleaved

- forwards or backwards reasoning in proving \( \phi_1, \ldots, \phi_n \vdash \phi \)

- automatic checking of simple instances of \( M \models \phi_i \)
phenomenon

world
cap.

jane, blah, class

Testing
observation

systematisation

jane, blah, java

Verification

Prediction

Logic
+ modelling

PLDI
Module Overview, I

- Part I: (review of) logic
  - propositional logic: truth table methods
  - logical consequence
  - natural deduction
  - soundness (and completeness)
  - predicate logic (non-finiteness of domains of interest)
  - beyond first-order logic
    * induction and recursion as distinctive features
    * typed logic
    * higher-order logic
    * logic *via* types: mechanisation in an interactive theorem prover
Module Overview, II

- Part II: specification
  - models and specification languages
  - propositional logic and first-order logic as specification languages
  - typed higher-order logic as a... 
  - models of computation: the lambda calculus
  - Hoare logic (if time)
Module Overview, III

- Part III: verification
  - use of a specific tool: COQ (the national symbol of France/its rugby team)
  - propositional logic and first-order logic exercises
  - reasoning about simple functional programs
  - more advanced material (if time): inductive definitions of behaviour
  - emphasis on mechanised reasoning
  - some model checking and other tools (if time)
Timetable

- Mondays (W2, W4, W6, W8, W10): 10am here; regular lecture slot

- Tuesdays (each week): 10am–12noon
  - an hour of “lecture” material
  - an hour of “tutorial” material in the JH lab
  - working with the proof tool
  - log-in after this lecture, and fire up `coqide` at a terminal prompt
  - some flexibility: probably 5 such sessions

- TBA: there is a clash with CS4203 graphics, so we need to co-ordinate
  can we shift the “lab/tutorial” hour to Tuesday afternoons?
Textbooks and other resources

- In the Library, long loan (1 copy), 4-hour loan (1 copy), 3-day loan (2 copies) (check?):
  - *Coq’Art: Interactive Theorem Proving and Program Development*, Bertot and Castéran

- Huth and Ryan: good for the introductory logic, excellent for model checking

- Bertot and Castéran: more extensive use after a while

- [http://coq.inria.fr/](http://coq.inria.fr/) reference manual, tutorial, IDE guide, libraries...
Requirements, Assessment, PtP

- Attendance at the Tuesday sessions is **REQUIRED**
- Reward is 10% of coursework assessment
- 3 practical assignments
  - **end Week 3 (TBC)** 20% survey essay: “Why does specification and verification matter?”
  - **end Week 7 (TBC)** 30%: basic proofs and verification in CoQ
  - **end Week 11 (TBC)** 40%: substantial verification exercise
Questions?