Revision Proposal for COMP2111 - System Modelling and Design

s.sequoiahgrayson

June 2025

1 Introduction and Background

This Revision Proposal has been made following encouragement and suggestions from HoS and Director of Education. The brief was to complete the review of Formal Methods (and Theory) courses in the School started by Carroll Morgan, and to then revise COMP2111 in light of this review. This Revision Proposal has been made following discussions with various stakeholders including Raveen De Silva, Gernot Heiser, Paul Hunter, Ron Van der Mayden, Craig McLaughlin, Carroll Morgan, Sushmita Ruj, Thomas Sewell, Rob Sison, Miki Tanaka, and Wayne Wobcke - many more than once. I would like to thank everyone for their time.

1.1 Summary of difference with existing version of COMP2111

- At the beginning of the course there is a new emphasis on model-theoretic semantics and tableaux proofs.
- Towards the middle of the course there is a new emphasis on the relationships between logical reasoning and computer science.
- In the second half of the course there is a renewed emphasis on formal methods, comprising Constraint Programming, Automated Theorem Proving, Software Verification, and System Implementation in the imperative/functional programming language Dafny.

Please see see Section 2 below for the details of all changes proposed.

1.2 Connections

- SENG2011 - This Revision Proposal provides both conceptual and technical support for the more focused study of program behaviour analysis undertaken in SENG2011.

- COMP3153 - The study of the logico-mathematical foundations of programming languages in the middle weeks of this Revision Proposal, along their support for formal verification in the later weeks, supports the study of fully automatic/algorithmic verification techniques in COMP3153.
- COMP3161 - The strong focus on syntax and semantics in the early weeks of this Revision Proposal support their more advanced study in COMP3161.
- COMP4141 - The detailed study of formal logics in early and middle weeks of this Revision Proposal supports the further study of formal languages in COMP4141.
- COMP4161 - The introduction to software verification in the later weeks this Revision Proposal supports the study of advanced approaches to software verification in COMP4161.

1.3 Proposed Textbooks

I have looked for many weeks, and I cannot find anything for the formal logico-mathematical part of the Revision Proposal that I think is suitable. Below are what I think to be the best candidates, along with my thoughts. There is one text below (Sitnikovski, 2022), that I believe may be suitable for the second half of the course.

- Brachman, R. J., and Levesque, H. J. (2004): Knowledge Representation and Reasoning. *Not bad for resolution, but the earlier sections are much too brief and too thin on details.*
- Gallier, J. H. (2003): Logic For Computer Science Foundations of Automatic Theorem Proving. *This book is great on its own terms, but is descends into categories in Chapter 2 and it does not look back. It is not suitable for our students in a 2000-level course as it is much too difficult.*
- Mordechai, B-A. (2012): Mathematical Logic for Computer Science - Third Edition. *This moves much too quickly for a 2000-level course. It is rigorous as far as it goes, but it is the lecture notes of the author's essentially. Too many details are missing.*
- Sitnikovski, B. (2022): Introducing Software Verification with Dafny Language: Proving Program Correctness. *This does not look too bad. I believe that it would be usable for the later weeks (7-10) of the Revision Proposal perhaps.*
- Turner, R. (2009): Computable Models. *This is a good book and it covers a lot of sensible topics. However, it is rather advanced and the author presents everything via types. It is not suitable for our students' level of logico-mathematical sophistication.*

As such, I propose to write up notes covering Weeks 1-5, and to look to get them published.

2 Proposed Revisions to ECLIPS Record for COMP2111 - System Modelling and Design

Please note - Below are all and only changes to the ECLIPS Record for COMP2111. Anything not mentioned is not proposed for changes.

2.1 Course Description for Handbook

2.1.1 Current Course Description

This course introduces rigorous and formal methods for modelling system behaviour. These methods support the modelling of abstract specifications and the refinement of abstract specifications through to concrete implementations. Consistency of formal development is verified by proof obligations and formal proof. The course will cover: specification, refinement, implementation, proof obligations, and proof. It reinforces, and builds on, prerequisite knowledge from MATH1081, especially set theory and predicate logic. The course will use case-studies and assignments to develop competence. The methods developed in this course are used in the SENG2011 workshops and in safety-critical industrial contexts.

2.1.2 Proposed Course Description

COMP2111 introduces students to theoretical computer science and formal methods in computing. It reinforces and builds upon prerequisite knowledge from MATH1081. Students will learn the foundations of logical reasoning, as well as the syntactic and semantic details of the formal languages that encode such reasoning. We will then explore the relationship between these languages and computer programs in detail. In the second half of the course, students will use everything thus far to explore Formal Methods in computer science in detail. Students will learn the principles of Constraint Programming, Automated Theorem Proving, Software Verification, and System Implementation in the imperative/functional programming language Dafny. No previous knowledge of Dafny is assumed.

2.2 Proposed Delivery Structure

36 hours of Lectures + Weekly two-hour Tutorials (throughout term)/Weekly two-hour Labs (occurring from Week 7 onwards).

Please note that 'Delivery Structure' does not occur in ECLIPS. It is included here for the sake of clarity.

2.3 Delivery Mode

2.3.1 Current Delivery Mode

Multimodal

2.3.2 Proposed Delivery Mode

In Person

2.4 Learning Outcomes

2.4.1 Current Learning Outcomes

- CLO1 - appreciate the relevance of discrete mathematics to computing
- CLO2 - make effective use of discrete mathematics concepts
- CLO3 - carry out rigorous reasoning about computing artefacts
- CLO4 - effectively employ a toolkit of formal modelling approaches frequently used in computing

2.4.2 Proposed Learning Outcomes

- CLO1 - Students will understand the relevance of logical reasoning to computer science.
- CLO2 - Students will learn how to build and check logically rigorous mathematical models.
- CLO3 - Students will understand the relationship between proofs in formal logics and computer programs.
- CLO4 - Students will learn how to carry out rigorous reasoning about computing artefacts.
- CLO5 - Students will learn how to effectively employ a toolkit of formal methods approaches to both software verification and system design.

2.5 Assessments

2.5.1 Current Assessments

- $3 \times$ Assignments roughly equally weighted 50%
- Final Exam weighted 50%

2.5.2 Proposed Assessments

- 4 x assignments equally weighted 40% (10% apiece) = 2 × paper-based + 2 × machine-based.
- Final Exam (invigilated in CSE labs) weighted 60% (hurdle at 50% of exam score).

2.6 Prerequisites

2.6.1 Current Prerequisites

MATH1081 AND (COMP1511 OR DPST1091 OR COMP1917 OR COMP1921)

2.6.2 Proposed Prerequisites

MATH1081 AND (COMP1511 OR DPST1091)

2.7 Exclusions

2.7.1 Current Exclusions

COMP2110 Software System Specification 2025.01IR

2.7.2 Proposed Exclusions

No exclusions

2.8 Key Search Terms

2.8.1 Current Key Search Terms

Computer Science, formal methods, formal reasoning about programs

2.8.2 Proposed Key Search Terms

Computer science, theoretical computer science, formal methods, formal reasoning about programs

3 Proposed Content

- **Week 1** - Propositional Logic.

  First Lecture - Syntax (wffs etc.), Semantics (truth-tables).

  Second Lecture - Validity, Tableaux Proofs, and Counterexamples.

- **Week 2** - General Predicate Logic with Identity.

First Lecture - Syntax (predicates, constantes, quantifiers, variables, wffs etc.), Semantics (Model-theoretic).

Second Lecture - Validity, Tableaux Proofs, and Countermodels.

- **Week 3** - Induction and Recursion

  First Lecture - Induction

  Second Lecture - Recursion

- **Week 4** - Resolution

  First Lecture - Resolution and Proof (Skolemization etc.).

  Second Lecture - Resolution and Computation (Herbrand's Theorem etc.).

- **Week 5** - Curry Howard Isomorphism

  First Lecture - Formulas as Types

  Second Lecture - Proofs as Programs

- **Week 6** - Flex Week

- **Week 7** - Specifications

  First Lecture - Hoare Logic (Hoare Triples and Inferences)

  Second Lecture - Proofs in Dafny and the Z3 Automated Theorem Prover (pre/postconditions, invariants, arrays, and termination).

- Week 8 - Induction in Dafny

  First Lecture - Induction in Dafny I

  Second Lecture - Induction in Dafny II

- Week 9 - Verification in Dafny

  First Lecture - Verification in Dafny I

  Second Lecture - Verification in Dafny II

- Week 10 - Implementation in Dafny

  First Lecture - Implementation in Dafny I

  Second Lecture - Implementation in Dafny II