Logic and Modeling (X_401015)

Robert Y. Lewis

1 Basic Info

- Period 5, Spring 2020
- Location: TBD
- Instructor: Dr. Robert Y. Lewis (r.y.lewis@vu.nl)
- Office: W&N S-414
- Office hours: after class, or by appointment
- Canvas page: TBD

2 Course Description

This course is an introduction to the use of logic in computer science. Logic is a fundamental tool with both theoretical and practical importance. Theoretically, the language of logic can be used to describe the foundations of our subject. There are deep ties between logic and computation, some of which we will see in this course. Practically, the logical languages taught in this course can be used to describe program behavior, database queries, and more; the proof calculi and proof assistant tools we will learn are used to rigorously check the correctness of programs. Safety-critical programming environments rely on the tools of formal logic.

This lecture primarily covers the languages of propositional logic and first-order predicate logic. We work with natural deduction as a proof system. A key distinction for this course is between *syntax* and *semantics*. The syntax of a formal language describes how to write well-formed formulas in that language and how to follow the rules of a corresponding proof system to produce well-formed proofs. The semantics of a language describe how to interpret formulas of that language in a particular model, to determine when they are true or false. We will see the relation between semantic and syntactic methods: the central keywords are *soundness*, *completeness*, and *consistency.* We will also pay attention to the expressive power of these languages, asking what properties of a model can and cannot be formulated in a particular logic. Alongside these theoretical concerns, we will discuss how to use these languages to model real-life situations, particularly those arising in computer science.

Proof assistants are computer-readable languages for writing and checking logical formulas and proofs. These tools are used in academia and industry for the logical verification of safety-critical programs. In this course, we will learn the basics of the Lean proof assistant. This tool gives instant feedback on your proofs, allowing you to easily test your understanding of deductions.

The study of decision procedures is an important topic in theoretical computer science, with close connections to logic. We will define what these problems are and look into the (un)decidability of different logics.

As a variation of the mentioned logics, we also consider modal logic with Kripke models as semantics. Logics based on this framework are widely used in program verification.

3 Course Objectives

The course objective is to obtain a good knowledge and understanding of the most important logical systems: propositional logic, predicate logic and modal logic. You will learn to use these systems to model data, knowledge and actions. An important aspect of the course is the ability to reason using these logics and reason about these logics: what can and what can not be expressed with a logic system, and what are the differences between the systems with respect to expressive power or the existence of decision procedures.

By the end of this course, you should be able to:

- Recognize, define, and use the fundamental terms of logical vocabulary. This includes, but is not limited to:
  - Terms related to the syntax and semantics of first-order logic, such as *valid*, *satisfiable*, *unsatisfiable*
  - Important meta-concepts of logic, such as *soundness*, *completeness*, *compactness*
  - Terms related to modal logic, such as *Kripke model*, *frame*
  - Concepts related to computation, such as *decidability*
- Determine whether logical sentences are valid, satisfiable, or unsatisfiable.
- Write pen-and-paper proofs in our natural deduction systems for propositional and first-order logic.
- Write proofs of propositional and first-order sentences in the Lean proof assistant.
- Produce counterexamples for sentences that are not valid.
- Translate between informal English sentences and sentences in propositional and first-order logic.

- Recognize modal logic formulas that correspond to certain frame properties and prove these correspondences.
- Give sketches of informal proofs for metatheorems, including definability and undefinability theorems.

4 Textbooks

- The primary reference for the first five weeks of this class is Avigad, Lewis, and Roux, *Logic and Proof*. This is a freely accessible text, available at `http://avigad.github.io/logic_and_proof/`. While this text has already been used to teach many courses (including this one), it is still being developed and expanded, and your feedback is very welcome!
- The primary reference for the last three weeks, and the secondary reference for the first five, is Huth and Ryan, *Logic in Computer Science*. This text contains many practice exercises, some of which are suggested on Canvas.

5 Assignments

- There is one 2.5 hour exam at the end of this course. The exam will ask you to demonstrate the skills listed in the previous section (except for writing proofs in Lean). You will be expected to define important concepts, write formal and informal proofs, and use logical languages to model situations.
- All students eligible for the exam will also be eligible for the retake exam, held approximately one month later. This exam will have the same structure as the first exam, but different questions. Students who sit both exams will have both grades submitted to the examination board regardless of which is higher.
- There are two homework assignments that ask you to write formal proofs in the Lean proof assistant.
  - Propositional logic: 20 problems, due three weeks into the period
  - First-order logic: 20 problems, due seven weeks into the period.

  These problems will be assigned and submitted through CoCalc (www.cocalc.com), an online editor and course management system that supports the Lean language. More details about CoCalc will be given during the period. The solutions submitted should be your own work: you are allowed to work on the problems in groups and in the exercise sessions, but should write your answers alone.

  Students are required to complete at least 10 of 20 problems on both assignments in order to qualify for the exam. (Since Lean is a proof checker, it will clearly tell you when you have correctly completed a problem!)

6 Grading

The course is graded on a scale from 0 to 10. The exam grade determines the final grade for the course. Students will earn up to half a bonus point for completing the Lean homework exercises. (A student who completes n Lean exercises will earn $n/80$ bonus points.)

7 Classroom Policies

This course consists of 15 lectures and 15 exercise sessions. The exercises will introduce concepts and techniques, and the exercise sessions are a chance for students to practice these techniques. Attendance is not required at either, but it is strongly encouraged. Priority at office hours will be given to students who have attended classes.

Lists of exercises to cover in the exercise sessions will be available on Canvas. It is encouraged to try some of these problems *before* the class, so you can come prepared to ask questions about topics you find difficult. Passively following the work of others during lectures and exercise sessions is unlikely to lead to success in this course.

Each class lasts 105 minutes, including a 15 minute break in the middle. Please feel free to ask questions during this break or after the lecture! For longer discussions, I encourage arranging a time to meet. I can also be contacted by email (r.y.lewis@vu.nl) or by Canvas message, and I check both regularly.

Course materials—lecture slides, practice problems, exam review topics, practice exams, and more—are posted on the course Canvas page. The schedule of lectures will be updated daily to reflect exactly what material was covered in each lecture.