UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Logical Foundations of Computer Science

2627-1-F1802Q122

Aims

In this course we introduce the foundations of logic in order to explain the notion of computation. Our aim is to allow the student to manage the recent achievements of logic in computer science.

Knowledge and understanding: The student will understand the concept of the Curry-Howard isomorphism and how it relates to functional programming and intuitionistic logic in the context of program correctness. Modal logics will be introduced, and it will be shown how they relate to the representation of knowledge.

Applying knowledge and understanding: The student will be able to write functions in lambda calculus, carry out type-checking exercises for functions in the context of program correctness, determine the intuitionistic validity of a formula representing a specification, and extract a lambda calculus program from the proof. The student will also be able to determine the satisfiability of modal formulas and model problems by writing modal formulas.

Making judgements: Ability to choose between different representations of the same function or of the same typing; ability to evaluate different formulas that represent the same knowledge.

communication skills: Ability to correctly define the introduced terms and to distinguish what each given definition captures; ability to state the main theorems, even in one's own words, while maintaining mathematical accuracy and highlighting their meaning.

learning skills: Ability to recognize that the introduced mechanisms can also be instantiated in non-functional programming languages; ability to generalize the use of logical tools to modal logics not covered in the course, and to select the most appropriate logical framework for addressing a new problem.

Contents

1. Historical overview and connections with the courses *Programming Languages* and

Languages and Computability

2. The Curry-Howard Isomorphism or propositions as types paradigm;

3. Foundations of Classical and Intuitionistic logics;

4. Modal and Description Logics.

Detailed program

1. Starting from functional programming and recursive theory, we present the intimate relationships between logic and computation. This motivates a brief introduction of the functional language known as Lambda Calculus and the notion of Type: both have an important role in the framework of program correctness;
2. previous point motivates the introduction of the Curry-Howard Isomorphism, also known as Propositions as types paradigm that is theoretically important both in typed functional languages and in program correctness;
3. previous points are the main motivations to study logic in computer science. It will be clear the importance of Intuitionistic logic in computer science. Thus we start to study Classical and Intuitionistic logic by introducing the key points: syntax, semantics and proof systems;
4. Modal logics are widely used in computer science for applications ranging from formal verification, to game analysis, to multi-agent system representation. We will study the basic properties of these logics.
   .

Prerequisites

Basic notions from first and second year of Bachelor Degree in Informatica;

Teaching form

Teaching format: 24 lessons of 2 hours each in-person.

Textbook and teaching resource

Open Logic Project is a collection of teaching materials on mathematical logic used in logic courses as taught in many philosophy departments.
Open Logic contains the topics of the course and it can be a good reference.

As regards standard books, there is no a single book covering all points of the course.
The following books and papers are owned by our library and/or are freely downloadable:

1. Dirk van Dalen. Logic and structure (3. ed.). Universitext. Springer, 1994. It contains material related to Classical and Intuitionistic logics (point 3 of the program);

2. Philip Wadler.Propositions as types. Commun. ACM, 58(12):75–84, 2015. non-technical paper;
3. Morten Heine Sörensen and Pawel Urzyczyn. Lectures on the Curry-Howard isomorphism. Elsevier, 2006. Advanced book, technical, contains points 1-3 of the program.

Semester

Second semester.

Assessment method

- Final exam (no intermediate exam);
- oral exam. Two possible choices:
    1. oral exam on the contents of the course;
    2. oral exam on deeper topics not considered in the program of the course.
- In both cases we evaluate the knowledge of the topics, included the ability to answer to questions and solve exercises, clarity of presentation and mathematical precision.

Office hours

On request, scheduled via email.

Sustainable Development Goals