Summary
We introduce formal verification as an approach for developing highly reliable systems. Formal verification finds proofs that computer systems work under all relevant scenarios. We will learn how to use formal verification tools and explain the theory and the practice behind them.

Content
Topics may include (among others) some of the following:
- Review of Sets, Relations, Computability, Propositional and First-Order Logic Syntax, Semantics, Sequent Calculus.
- State Machines. Transition Formulas. Traces. Strongest Postconditions and Weakest Preconditions.
- SAT Solvers and Bounded Model Checking.
- Model Checking using Binary Decision Diagrams.
- Symbolic Execution. Satisfiability Modulo Theories.
- Abstract Interpretation.
- Set theory for verification.

Learning Prerequisites
Recommended courses
Computer Language Processing / Compilers

Important concepts to start the course
Discrete Mathematics (e.g. Kenneth Rosen: Discrete Mathematics and Its Applications)
Learning Outcomes
By the end of the course, the student must be able to:
• Formalize specifications
• Synthesize loop invariants
• Specify software functionality
• Generalize inductive hypothesis
• Critique current software development practices

Teaching methods
Instructors will present lectures and exercises and supervise labs on student laptops.

Expected student activities
Follow the course materials, take mid-term, and complete and explain projects during the semester.

Assessment methods
The grade is based on the written mid-term, as well as code, documentation, and explanation of projects during the semester. Specific percentages will be communicated in the first class.

Supervision
Office hours  Yes
Assistants  Yes
Forum  Yes

Resources

Bibliography
• Peter B. Andrews: *An Introduction to Mathematical Logic and Type Theory (To Truth Through Proof)*, Springer 2002.
• http://logitext.mit.edu/tutorial

Ressources en bibliothèque
• Handbook of model checking / Clarke
• Introduction to mathematical logic and type theory / Andrews
• Principles of Program Analysis / Flemming
• The Calculus of Computation / Bradley
• Logic in Computer Science / Huth
• Handbook of model checking : Model Checking Security Protocols / Bassin

Notes/Handbook
• https://lara.epfl.ch/w/fv

Websites
• https://lara.epfl.ch/w/fv

Moodle Link
• https://go.epfl.ch/CS-550

Videos
• https://tube.switch.ch/channels/f2d401d

Prerequisite for
MSc thesis in the LARA group