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 the 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.
• Hoare Logic. Inductive Invariants. Well-Founded Relations and Termination Measures
• Modeling Hardware: Verilog to Sequential Circuits
• Linear Temporal Logic. System Verilog Assertions. Monitors
• SAT Solvers and Bounded Model Checking
• Model Checking using Binary Decision Diagrams
• Loop Invariants. Hoare Logic. Statically Checked Function Contracts. Relational Semantics and Fixed-Point Semantics
• Symbolic Execution. Satisfiability Modulo Theories
• Abstract Interpretation and Predicate Abstraction
• Information Flow and Taint Analysis
• Verification of Security Protocols
• Dependent and Refinement Types

Learning Prerequisites
Recommended courses
Computer Language Processing / Compilers

Important concepts to start the course
Discrete Mathematics

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 meaningless course description forms

Teaching methods
Instructors will present lectures, conduct whiteboard or blackboard exercises, and supervise labs on student laptops.

Expected student activities
Attend lectures (optional but highly recommended), solve exercises on whiteboard and continue at home as needed, complete computer labs.

Assessment methods
We will assign written exams and grade labs.

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
• Aaron Bradley and Zohar Manna: The Calculus of Computation - Decision Procedures with Applications to Verification
• Tobias Nipkow, Gerwin Klein: Concrete Semantics with Isabelle/HOL
• Peter B. Andrews: An Introduction to Mathematical Logic and Type Theory
• Nielson, Flemming, Nielson, Hanne R., Hankin, Chris: Principles of Program Analysis
• Michael Huth and Mark Rayan: Logic in Computer Science - Modelling and Reasoning about Systems

Websites
• https://lara.epfl.ch/w/fv

Moodle Link
• https://moodle.epfl.ch/course/view.php?id=13051

Videos
• https://youtu.be/mm6CCGSDmOw?t=39
• https://www.youtube.com/watch?v=oLS_y842fMc
• https://www.youtube.com/channel/UCP2eLEqI4tROYmlYm5mA27A