IS1330
Theoretical computer science and discrete mathematics: formal languages and computability

Professor: Marc Aiguier

Language of instruction: French – **Number of hours**: 36 – **ECTS**: 3

Prerequisites: None

Period: S8 Elective 09 February to March IN28IE2, SEP8IE2

Course Objectives

Understanding the basic principles and formal tools (ie based on mathematics) at the base of all the methods of design and implementation of information systems. This course will adress the fundamentals of:

- induction and recurrence (lattice theory, well-founded sets and their equivalence with the mathematical induction). The objective is to formalize the basic notions of induction and recursion underlying all discrete mathematics.
- algorithmic (Gödel / Herbrand's recursive functions, Turing machine and lambda calculus and all the theorems of fixed point and the results associated with undecidability). The idea is to define formally (ie mathematically) what a decision problem is and to give a formal denotation of the concept of algorithm (Church thesis).
- the theory of complexity (both complexity classes P and NP, and structuring the class NP - NP-complete and NP-hard problems).
- design systems (rational languages ■■and automata). The objective is to study the formal basis of all methods of modeling of computer systems.

On completion of the course, students should be able to

Knowing how to adress the "discrete" modelling of a given problem for its implementation (decidable problem), understand the basic formal tools for this modelization.

Course Contents

Foundations of induction and recursion

- Order and preorder
- Upper bounds and lower bounds
- Sets and well-founded induction
- Lattices and fixed points (complete lattices and continuous functions, fixed points and monotone functions)

Foundations of algorithms (computable functions and decidable problems and complexity theory)

- Recursive functions (FR): primitive recursive functions, recursive functions (Problem of the Ackermann function) definition of a decidable problem, undecidability of the halting problem, computability on lists and trees (coding functions) universal recursive function (interpreter).
- Step by step Computation (other calculation models): Turing Machines (MT), equivalence theorems (FR MT LC) Church thesis.
- Complexity theory (time complexity, classes P and NP, structuring the class NP - NP-complete and NP-hard problems, first NP-complete problem)

Rational languages ■■and automata

- Free monoid
- Rational languages ■■(pumping lemma)

- ✧ Finite Automata (labeled transition systems, completion, determination, minimality, algebraic operations)

Teaching Material and Textbooks

An handout in French, plus tutorials for each pcs concerned

Evaluation

A 3-hr written examination at end of course. Course handout and tutorials for each pcs and their correction allowed. Laptops and other documents not allowed.