2EL1540 – Theoretical computing

Instructors: Marc Aiguier, Pascale Le Gall
Department: DÉPARTEMENT INFORMATIQUE
Language of instruction: FRANCAIS
Campus: CAMPUS DE PARIS - SACLAY
Workload (HEE): 60
On-site hours (HPE): 35,00
Elective Category : Fundamental Sciences
Advanced level : Yes

Description
This teaching gives a part of the fundamentals of computer science on its two paradigms of computation, which are reduction (calculation step by step) and resolution (logical inference / automatic reasoning).

Quarter number
SG8

Prerequisites (in terms of CS courses)
Algorithms and Complexity course (ST Modelling)
An interest in mathematical abstraction and reasoning

Syllabus

The course addresses the following fundamental notions:

- Induction and recurrence (well-founded sets ...).
- Computability (Gödel/Herbrand's recursive functions, Turing machines and all the associated indecidability results).
- Mathematical logic (syntax, semantics and proof systems). Propositional and first-order logics will be detailed.

The course is composed as follows:

- Induction and recurrence.

The following notions will be presented: set theory (ordering and preordering, upper and lower bounds, well-founded sets and induction, formal systems, proofs, correctness and completeness).

- Propostional logic.

The following notions will be presented: syntax, semantics and proof systems, binary decision tree, tableaux method, DPLL algorithm, satisfiability, SAT-solvers, and proof systems (resolution, sequent calculus and natural deduction).

- Computability and complexity.

The following notions will be presented: primitive recursive and recursive functions, computable and non-computable problems, Turing machine, equivalence theorems, Church thesis, and complexity theory.

- First-order logic.

The first-order logic is an extension of propositional logic, and is the privileged logic for describing data type structures.

Class components (lecture, labs, etc.)

The course will be divided into 15 hours of lectures and 15 hours of tutorials.
One or more personal works (project with computer implementation or problem to be solved) will be proposed and will constitute the continuous control mark.

Grading

The evaluation will be done by means of a project and a written exam of 2 hours.
The final grade will be divided into 50% for the continuous assessment and 50% for the written exam.
For this exam, only the handout and personal notes are allowed. Electronic devices (laptops, mobile phones and tablets) are not allowed.

Course support, bibliography

Students will be provided with a handout in French, as well as TD subjects with correction elements.

Resources

- Teaching team (names of lecturers): Marc Aiguier and Pascale Le Ga
- ll
- Size of the classes: at most 35 students

- Software tools and number of licences required: the only software used (prolog, solvers, proof assistants) will be free software that students will install on their personal machines

Learning outcomes covered on the course

Understand the fundamental principles and the formal (i.e. mathematically based) tools that underlie all methods of designing, verifying and implementing computer systems.

Formalizing computing problems and mastering the fundamental theoretical tools necessary to reason about these formalizations.

These theoretical tools are based on computation models classically used for the complexity analysis of the algorithms (see the course “Algorithms and Complexity”), as well as on reasoning methods based on mathematical logic.

Description of the skills acquired at the end of the course

Formalizing computing problems and mastering the fundamental theoretical tools necessary to reason about these formalizations. These theoretical tools are based on computation models classically used for the complexity analysis of the algorithms (see the course “Algorithms and Complexity”), as well as on reasoning methods based on mathematical logic.