Computability and Complexity Theory (COMP0017)

Description

Aims:
The module addresses the theoretical and practical limitations of computation and provides a theoretical framework for modelling computation. The concepts of undecidability and intractability are introduced through a number of examples. The module will convey the proof techniques that are used to classify problems and it is intended that students learn how to apply them in order to classify unfamiliar problems for themselves.

Learning outcomes:
On successful completion of the module, a student will be able to:
1. analyse the complexity of a variety of problems and algorithms;
2. reduce one problem to another;
3. prove that a problem is undecidable;
4. find a polynomial time reduction from one problem to another;
5. determine the complexity class of a decidable problem;
6. categorise the complexity of a language.

Content:
Models of Computation:
- Deterministic Turing machines;
- Equivalent Turing machines;
- Register machines.

Languages:
- Language recognition;
- Language acceptance;
- Recursive languages;
- Recursively enumerable languages.

Undecidability:
- The Halting Problem;
- Problem reduction;
- Undecidability of the tiling problem;
- Undecidability of first-order logic;
- Other unsolvable problems.

Key information

Year: 2019/20
Credit value: 15 (150 study hours)
Delivery: UG L6, Campus-based
Reading List: View on UCL website
Tutor: Prof Robin Hirsch
Term: Term 1
Timetable: View on UCL website

Assessment

Find out more

For more information about the department, programmes, relevant open days and to browse other modules, visit ucl.ac.uk

Disclaimer: All information correct as of June 2019. Please note that aspects of the module may be subject to change. UCL will make best efforts to inform applicants of major changes.
Non-determinism:
- Non-deterministic Turing machines;
- Polynomial-time reduction;
- Elementary properties of polynomial time reduction;
- The complexity classes P, NP, NP-complete;
- Cook's theorem;
- How to prove NP-hardness of various problems.

Probabilistic Algorithms:
- Examples of probabilistic algorithms;
- How to make 'almost sure' your algorithm is correct;
- Complexity analysis of probabilistic algorithms;
- The complexity classes PP and BPP.

Other Complexity Classes:
- Space complexity;
- Savitch’s theorem;
- Exponential time;
- Non-elementary problems.

Requisites:
In order to be eligible to select this module, a student must be registered on a programme for which it is a formally-approved option or elective choice AND must have:
- passed Theory of Computation (COMP0003);
- passed Algorithms (COMP0005);
- passed Logic and Database Theory (COMP0009).