



MATH43011 - 2007/2008

General Information

- Title: Computational Complexity
- Unit code: MATH43011
- Credits: 15
- Prerequisites: A general mathematics background. Familiarity with the propositional calculus such as MT2151/ MATH20302 *Propositional Logic*, MATH31011 (2006-2007) *Mathematical Logic* or MATH43001 *Predicate Calculus* provide would be useful but is not absolutely essential. The course does not contain any practical computing.
- Co-requisite units: None
- School responsible: Mathematics
- Members of staff responsible: Prof. [Jeff Paris](#)

Specification

Aims

The aims of this course are:

- to introduce the main model of computation currently being employed in the theory of computation, Turing machines;
- to introduce the key parameters quantifying computational complexity (deterministic, non-deterministic, random, time, space) and the relationships between them.

Brief Description of the unit

It is now rather well known that the Clay Foundation is offering a million dollars for solving the *Travelling Salesman Problem*, in a nutshell the problem of determining how *hard* it is to find a salesman's shortest itinerary for visiting some set of towns. It turns out that this is just one of a multitude of similar questions coming from all areas of mathematics. The purpose of this course is to explain how this notion of the 'hardness of a problem' may be captured mathematically and to demonstrate some of the surprisingly rich (and still little understood) structure of mathematical problems themselves that follows from this.

Learning Outcomes

On successful completion of this course unit students should

- be familiar with the way Turing Machines work and their capabilities and limitations, and be able to construct examples to answer simple problems;
- be aware of the main parameters quantifying the computational complexity classes, and the relations between them, and be able to classify and compare the hardness of problems in simple cases.

Future topics requiring this course unit

None.

Syllabus

1. **Turing Machines:** Alphabets, languages, Turing machines, recursive functions, decidability. Multitape and nondeterministic Turing Machines. Universal machines and undecidability. [11 lectures]
2. **Time and Space Complexity Classes:** The time and space classification, speed up. Space and time hierarchy theorems. P and NP, NP-completeness. P= NP? (Travelling Salesman Problem). Savitch's Theorem and the Immerman-Szelepcsenyi Theorem.. [17]

3. Other Complexity Classes : Probabilistic complexity classes RP, ZPP. Circuit complexity. [5]

Textbooks

The course is self contained and you will not be required to consult any books. However the following (all available in the JRUL) cover some of the same material as the course and may provide interesting complementary reading.

- Minsky, *Computation, Finite and Infinite Machines*, 1967.
- Aho, Hopcroft and Ullman, *Design and Analysis of Computer Algorithms*, 1974.
- Garey and Johnson, *Computers and Intractability*, 1979.
- Papadimitriou, *Computational Complexity*, 1994.
- Sipser, *Introduction to the Theory of Computation*, 1997.
- Minsky and Papert, *An Introduction to Computational Geometry*, 1969.

Teaching and learning methods

33 lectures plus weekly office hours.

Assessment

Mid-semester coursework: two take home tests weighting 20%

End of semester examination: two and a half hours weighting 80%

Arrangements