Theory of Algorithms

ian.mcloughlin@atu.ie

January 18, 2023

An introduction to computational theory, computational complexity, and the evolution of computation from its historical origins to modern day paradigms.

Learning Outcomes

On completion of this module the learner should be able to[1]:

1. Identify difficult computational problems in everyday computing.
2. Define the common models of computation.
3. Design computer programs using a variety of computational paradigms.
4. Analyse the complexity of an algorithm.

[1] We recently updated these learning outcomes. They are making their way through our quality assurance procedures.

Assessment

Assessment is through a final written exam and a project[2]:

30% Project

70% Final exam

[2] The project will be in the form of a Jupyter notebook in a GitHub repository.

Delivery

- This is a semester-long module. Realistically, we will have ten uninterrupted teaching weeks[3].
- There are many ideas about how lecturers should deliver modules. Some suggest a top-down, structured approach where topics are clearly defined ahead of time. Others suggest involving students in decisions, letting content evolve during the semester. Let's not be idealistic about it: we'll have an initial plan and tailor it during the semester.
- It is worth discussing what you as a class would like to work on and giving me feedback on it. Just keep in mind that everyone might want something different. Also, remember that there is one lecturer and dozens of students in each of several modules. Time is limited, we will have to be careful about scope creep.

[3] Each semester typically has thirteen teaching weeks but some of those weeks have public holidays and other interruptions.

Lectures and Practicals

- Traditionally, lectures covered theory and students applied the theory in practicals. That can feel a bit contrived and artificial, especially in computing where practice often comes before theory.

- We won't make a clear distinction between lectures and practicals where possible. Rather, we will focus on topics, projects, and problems.

Topics

We will start with a plan to cover these four topics.

Structures and Operations: Sets, tuples, binary operations.

Formal Languages: Strings, languages, computability.

Computation: Automata, complexity classes.

Difficult Problems: NP Completeness, exponential problems.

Advice

- Everyone procrastinates, you need a strategy to compensate. You will be less stressed if you work regularly, a bit every week.
- Review the project marking scheme regularly and work to it.
- Be able to demonstrate your work. This is easier for practical work, you often have code and the like. Theoretical work can be demonstrated through writing, images, plots, and diagrams.

Policies

- In April 2022, GMIT merged with IT Sligo and LyIT to become ATU, the Atlantic Technological University.
- Although the merger has happened, it will take a couple of years for our systems and policies to fully merge.
- During this time, we will continue to use GMIT's policies where an ATU policy has not yet superseded them.
- That means the GMIT Quality Assurance Framework.[4]

GMIT is now ATU.

[4] GMIT. *Quality Assurance Framework*. `https://www.gmit.ie/general/quality-assurance-framework`.