Module Introduction

5002CEM — Theory of Computation

Dr Kamal Bentahar

School of Computing, Electronics and Mathematics
Coventry University

The team

- Ian Dunwell
- Kamal Bentahar — Module Leader
- Paul Lunn
- Xingang WANG

Teaching and Learning

Lecture	12×2 hours	12%
Laboratory	12×3 hours	18%
Self guided	145 hours	70%
Total	200 hours	100%

- **Pen and paper**. Supporting tools: JFLAP, Programming (Python).
- **Lectures:** Tuesdays 11am–1pm, in ECG-24.
- **Tutorials/exercises:** Check your timetable.
  - 2-hour lab: for you to work on the exercises.
  - 1-hour lab: present model solutions.
- Formative tests.

Assessment

Assessment Components:

Component	Type	Credits	Learning Outcomes
Cw	Applied Core	10	1, 2
Ex	Applied Core	10	1, 3, 4

Pass requirements:

- Coursework $\geq 40\%$ (Mid-term In-Class 1-hour online test.)
- **and** Exam $\geq 40\%$ (2 hours in December)

Resits: Next opportunity (Semester) — Capped at 40%.

So what is 5002CEM about?

Understand the **theoretical foundations** of Computer Science, and from this an appreciation of the **limitations of computation** and the important questions that remain open to this day.

The module covers:

1. Formal specification of languages.
2. The main models of computation
3. What these models tell us about issues of computability and complexity

It's fun, cool, intellectually challenging, insightful, . . .

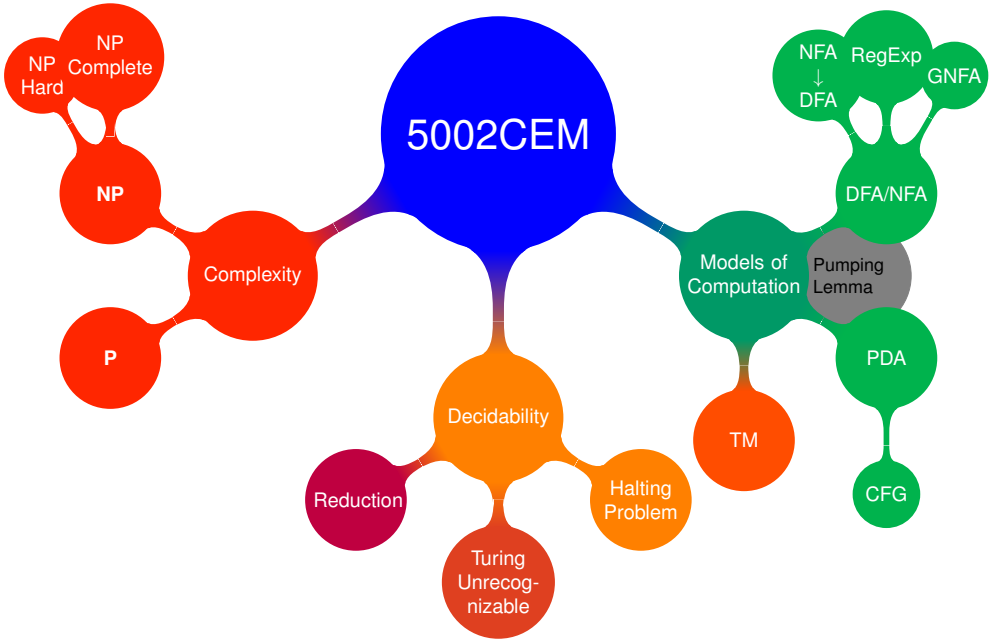
. . . it is! :-)

1. What is an “algorithm”? How “hard” is a problem? Can we “compute/solve” anything? If not then what are the limits.
2. For example:
   - $a^*b^*, a^nb^n, a^ib^jc^k$
   - $\{w \in \{0,1\}^* \mid w \text{ has equal number of 0s and 1s}\}$
   - L recognized by a given automaton
3.
   - Deterministic/Non-Deterministic Automoata (DFA/NFA)
   - Push Down Automata (PDA)
   - Turing Machines (TM).
4.
   - Complexity classes: P, NP, NP-complete, NP-hard, etc.
   - Algorithms to solve or heuristics to try...

Intended Module Learning Outcomes

On completion of this module the student should be able to:

1. Demonstrate the ability to use **formal notation** to specify patterns and **languages**.
2. Specify and be able to simulate various types of **automata**.
3. Demonstrate the ability to explain the connection between **algorithms**, **models of computation**, and **language classes**.
4. Classify the **computability** and **complexity** of problems.



Indicative Content

- **Mathematical background (Review):** Sets, functions, relations, propositional logic, and predicate calculus.
- **Formal Languages:** Regular languages and expressions; Context-free grammars. Applicatons to solve practical problems.
- **Models of Computation:** Finite State Automata (Deterministic and Non-deterministic); Push-down Automata; Turing machines. The relationships between models and classes of languages. The limits of models (Pumping Lemma). Practically use via a simulation package such as JFLAP.
- **Computability:** The Church-Turing Thesis, Reduction, Undecidability, and Unrecognisability.
- **Complexity:** Review of O-notation. The P versus NP question, NP-completeness, Polynomial time verification, Polynomial time reduction. Search problems and NP-hardness. Overview of further complexity classes (e.g. PSPACE, EXPTIME).

Books

Essential Reading

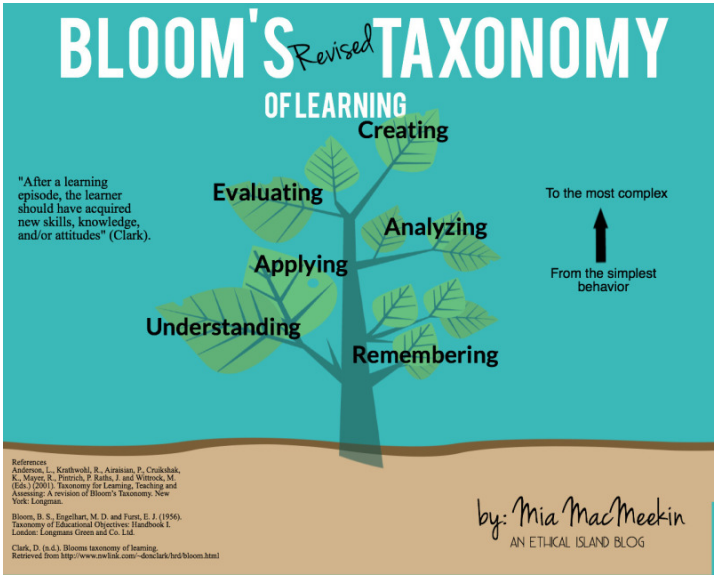
Sipser, M. (2013). **Introduction to the theory of computation** (3rd international ed.). Cengage Learning.

Recommended Reading

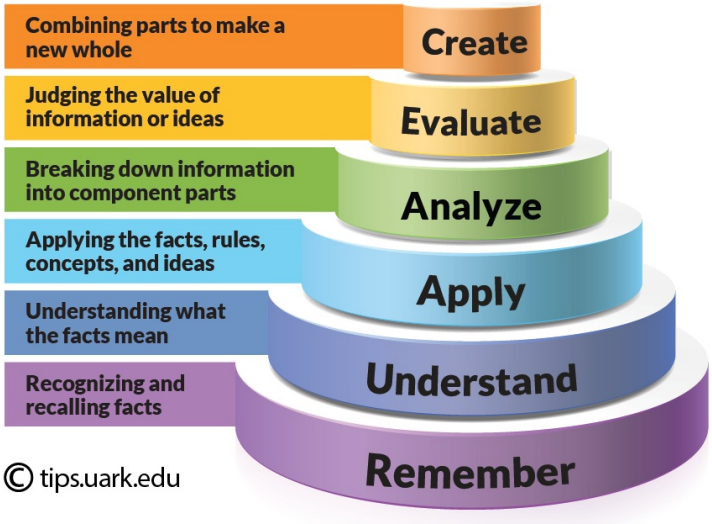
Garey, S. and Johnson, D. (1979) **Computers and Intractability: A Guide to the Theory of NP-Completeness.** Freeman

Dean, N. (1996) **The Essence of Discrete Mathematics.** Prentice Hall

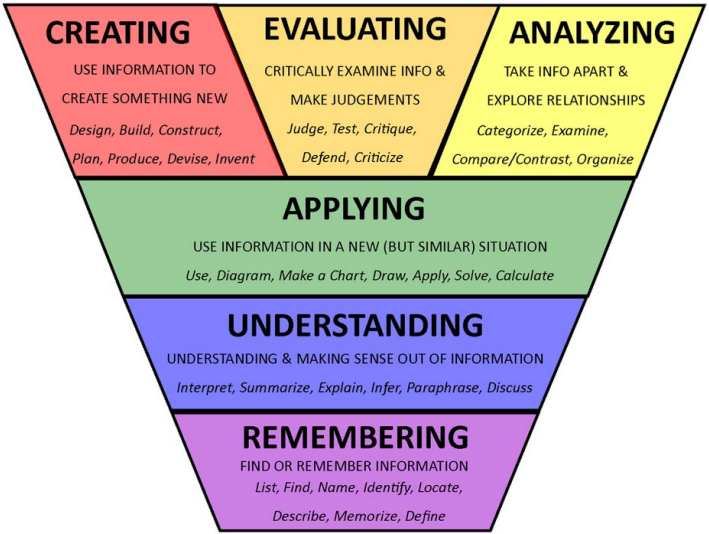
How to learn!


BLOOM'S Revised TAXONOMY
OF LEARNING
Creating
Evaluating
Analyzing
Applying
Understanding
Remembering
"After a learning episode, the learner should have acquired new skills, knowledge, and/or attitudes" (Clark).
To the most complex
From the simplest behavior
References
Anderson, L., Krathwohl, R., Airaisian, P., Cruikshak, K., Mayer, R., Pintrich, P. Raths, J. and Wittrock, M. (Eds.) (2001). Taxonomy for Learning, Teaching and Assessing: A revision of Bloom's Taxonomy. New York: Longman.
Bloom, B. S., Engelhart, M. D. and Furst, E. J. (1956). Taxonomy of Educational Objectives: Handbook I. London: Longmans Green and Co. Ltd.
Clark, D. (n.d.). Blooms taxonomy of learning. Retrieved from http://www.nwlink.com/~donclark/hrd/bloom.html
by: Mia MacMeekin
AN ETHICAL ISLAND BLOG

How to learn!



How to learn!


CREATING
USE INFORMATION TO CREATE SOMETHING NEW
Design, Build, Construct, Plan, Produce, Devise, Invent
EVALUATING
CRITICALLY EXAMINE INFO & MAKE JUDGEMENTS
Judge, Test, Critique, Defend, Criticize
ANALYZING
TAKE INFO APART & EXPLORE RELATIONSHIPS
Categorize, Examine, Compare/Contrast, Organize
APPLYING
USE INFORMATION IN A NEW (BUT SIMILAR) SITUATION
Use, Diagram, Make a Chart, Draw, Apply, Solve, Calculate
UNDERSTANDING
UNDERSTANDING & MAKING SENSE OUT OF INFORMATION
Interpret, Summarize, Explain, Infer, Paraphrase, Discuss
REMEMBERING
FIND OR REMEMBER INFORMATION
List, Find, Name, Identify, Locate, Describe, Memorize, Define