Module Introduction

Dr Kamal Bentahar

School of Science, Coventry University

So what is this 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.

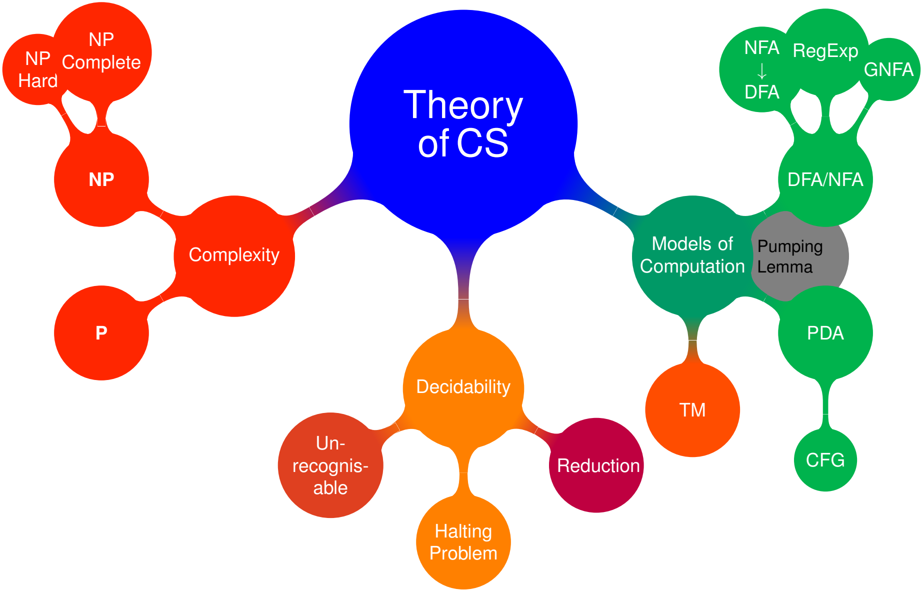
The module covers:

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 a "**problem**"? What is "**computation**"? What is an "**algorithm**"?
2. How "hard" is a problem? Can we "compute/solve" anything? If not then what are the limits?
3. Symbolic notation. For example:
   - $\{w \in \{0,1\}^* \mid w \text{ has equal number of 0s and 1s}\}$
   - $0^*1^*, a^nb^n, a^ib^jc^k$
   - L recognized by a given automaton
4. Models of computation.
   - Deterministic/Non-Deterministic Finite Automoata (DFA/NFA)
   - Push Down Automata (PDA)
   - Turing Machines (TM).
   - Regular expressions (RegEx).
   - Context-free grammars (CFG).
   - Grammars.
5. Classes: Regular, context-free, decidable, undecidable, recognisable, and un-recognisable!
6. Complexity.
   - Complexity classes: **P**, **NP**, **NP-complete**, **NP-hard**, **L**, **NL**, etc.
   - **Algorithms** to solve or **approximation/meta-heuristics** to try.


Theory of CS
Complexity
NP
NP Complete
NP Hard
P
Decidability
Un-recognis-able
Halting Problem
Reduction
Models of Computation
TM
Pumping Lemma
DFA/NFA
NFA ↓ DFA
RegExp
GNFA
PDA
CFG

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). Greedy search and meta-heuristics.

Learning Style

- **Pen and paper!**
  Supporting tools: JFLAP, Programming (Python).
- **Module Delivery**
  - Lectures: Wednesdays 11am-1pm.
  - Practical labs: Thursdays 9-11am. (Feedback)

Lecture	12$\times$2 hours	12%
Laboratory	12$\times$2 hours	12%
Self guided	152 hours	76%
Total	200 hours	100%

- **Assessments**
  - Formative (Feedback).
  - Summative (Test and Exam).

Books

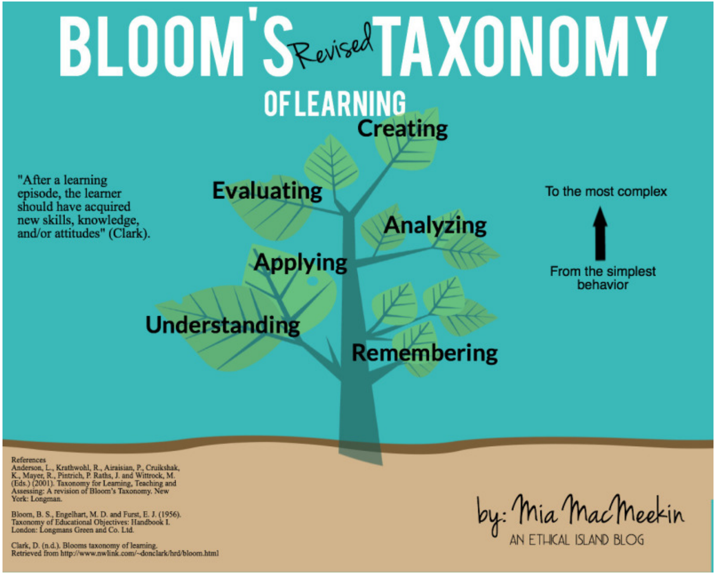
Essential Reading

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

Recommended Reading

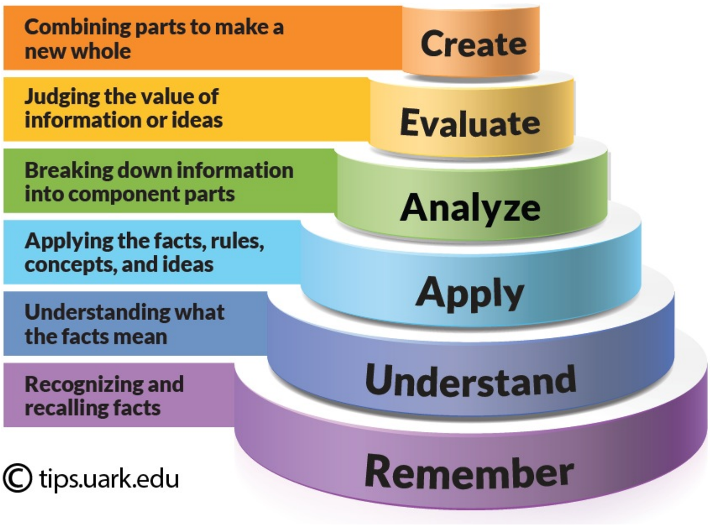
- Garey, S. and Johnson, D. (1979) **Computers and Intractability: A Guide to the Theory of NP-Completeness.** Freeman
- Dean, N. (1996) **The Essence of Discrete Mathematics.** Prentice Hall

How to learn!



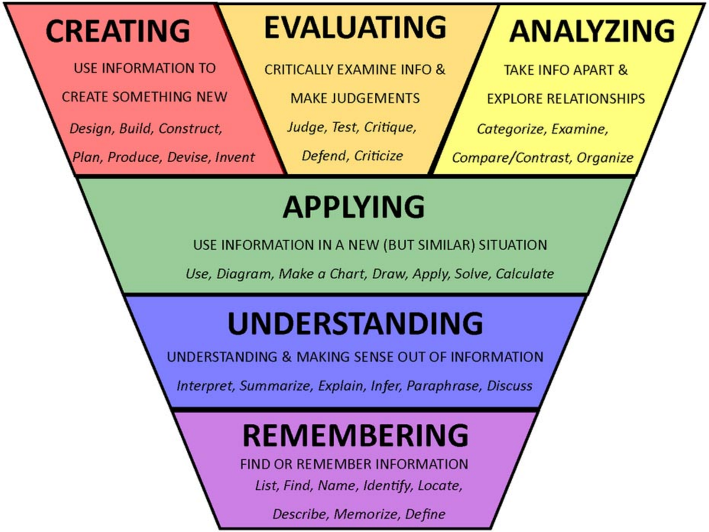
BLOOM'S Revised TAXONOMY
OF LEARNING
"After a learning episode, the learner should have acquired new skills, knowledge, and/or attitudes" (Clark).
Creating
Evaluating
Analyzing
Applying
Understanding
Remembering
To the most complex
From the simplest behavior
References
Anderson, L., Krathwohl, R., Airaisian, 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!



Combining parts to make a new whole
Create
Judging the value of information or ideas
Evaluate
Breaking down information into component parts
Analyze
Applying the facts, rules, concepts, and ideas
Apply
Understanding what the facts mean
Understand
Recognizing and recalling facts
Remember
© tips.uark.edu

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