Theoretical Aspects of Computer Science
(380CT)

Dr Kamal Bentahar

School of Engineering, Environment and Computing
Coventry University

03/10/2016

The team

- Dr Kamal Bentahar (ML, ab3735)
- Dr Abdulrahman Altahhan (ab8556)
- Dr Matthew England (ab9797)
- Dr Rob Low (mtx014)

380CT?

380CT?

1. Foundations of CS: practical & theoretical understanding.

- What is an “algorithm”?
- How “hard” is a problem?
- Can we “compute/solve” anything?
- If not then what are the limits.

380CT?

1. Foundations of CS: practical & theoretical understanding.
2. Formal specification of patterns and “languages.”

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

380CT?

1. Foundations of CS: practical & theoretical understanding.
2. Formal specification of patterns and “languages.”
3. Models of computation and the issues of computability and complexity.

- Deterministic/Non-Deterministic Automoata (DFA/NFA)
- Push Down Automata (PDA)
- Turing Machines (TM).

380CT?

1. Foundations of CS: practical & theoretical understanding.
2. Formal specification of patterns and “languages.”
3. Models of computation and the issues of computability and complexity.
4. Algorithmic techniques used to tackle complex problems.

- Complexity classes: P, NP, NP-complete, NP-hard, etc.
- Algorithms to solve or heuristics to try...

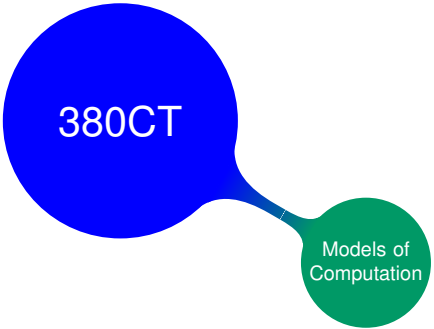
380CT?

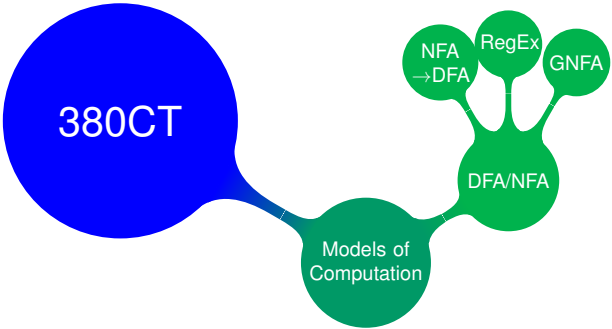
1. Foundations of CS: practical & theoretical understanding.
2. Formal specification of patterns and “languages.”
3. Models of computation and the issues of computability and complexity.
4. Algorithmic techniques used to tackle complex problems.

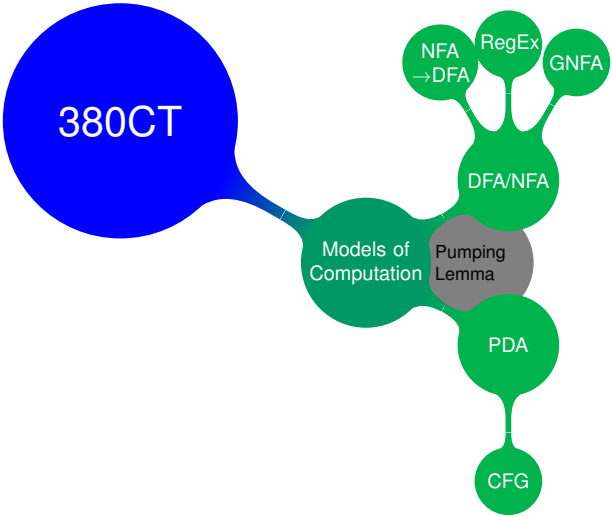
It’s fun, cool, intellectually challenging, insightful, . . .

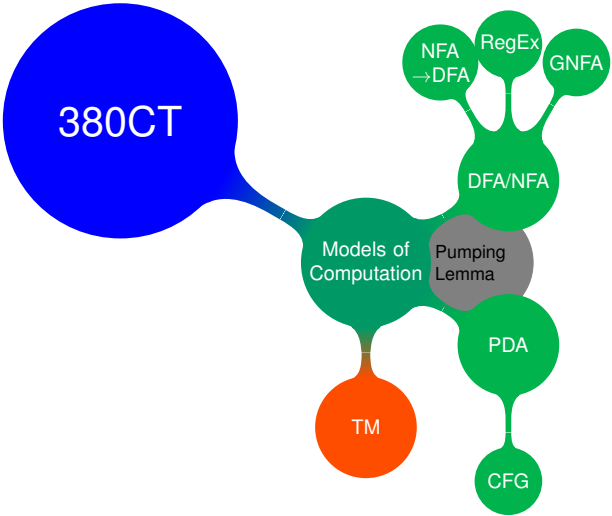
. . . it is! :-)

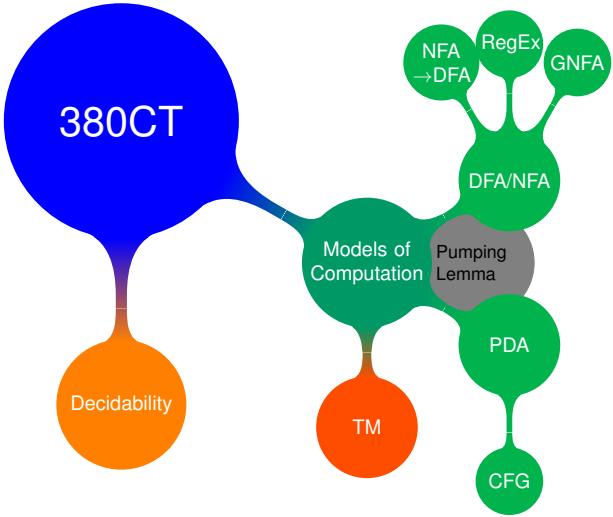

380CT

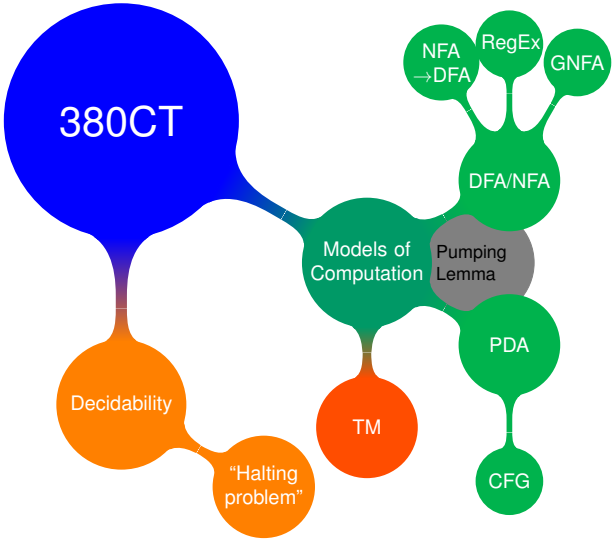

380CT
Models of Computation

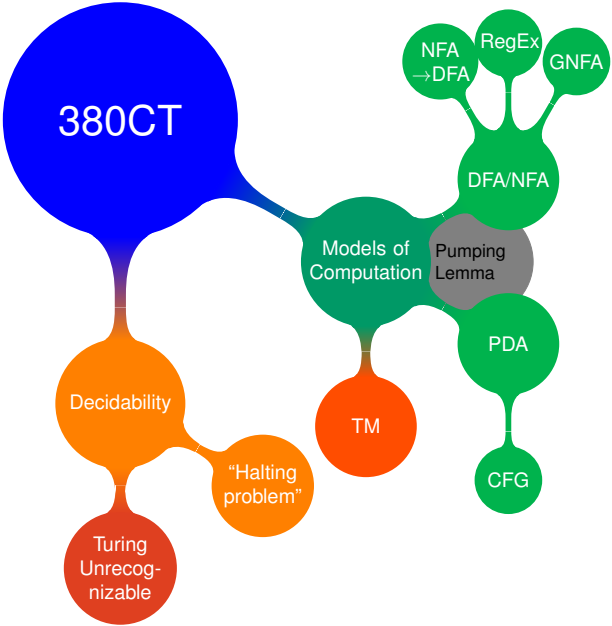

380CT
Models of Computation
DFA/NFA
NFA →DFA
RegEx
GNFA

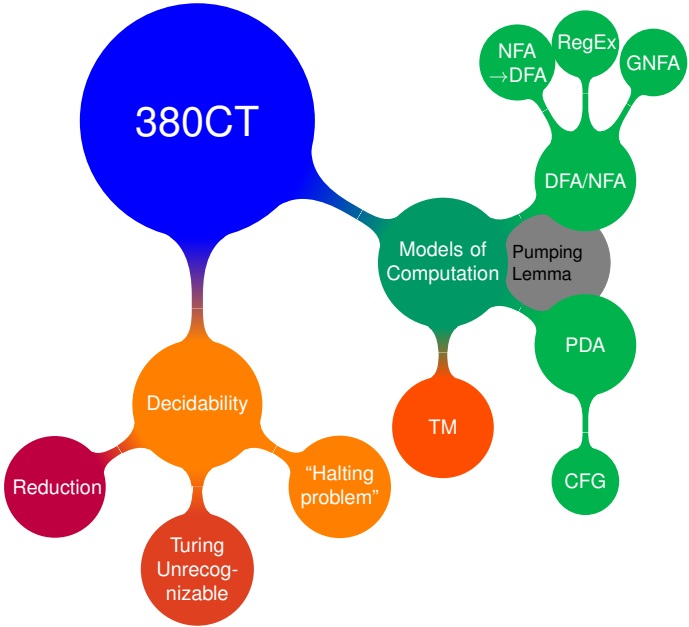

380CT
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG

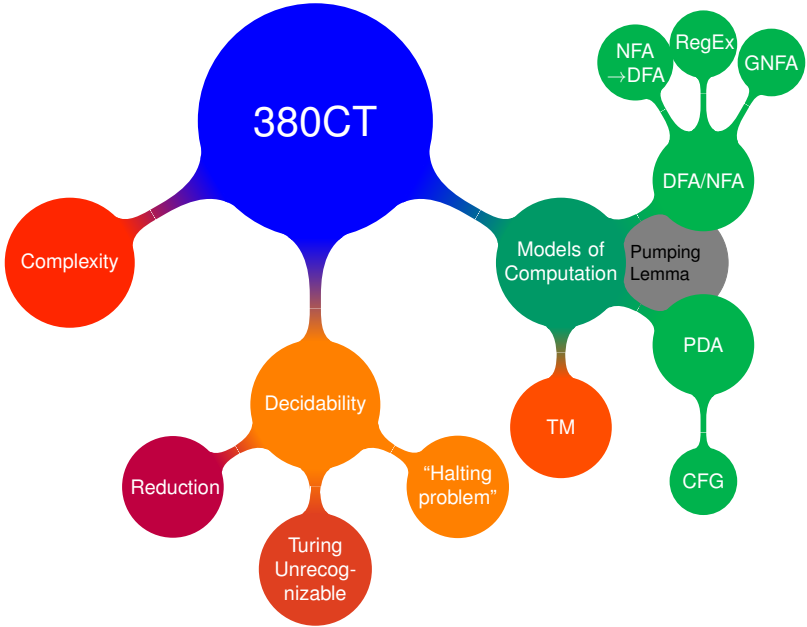

380CT
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM

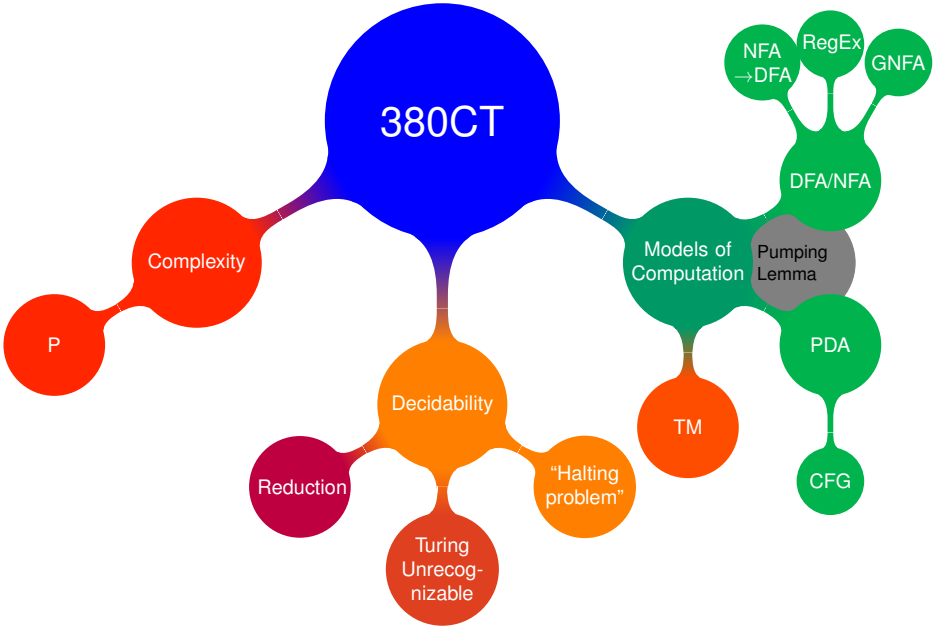

380CT
Decidability
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
TM
CFG

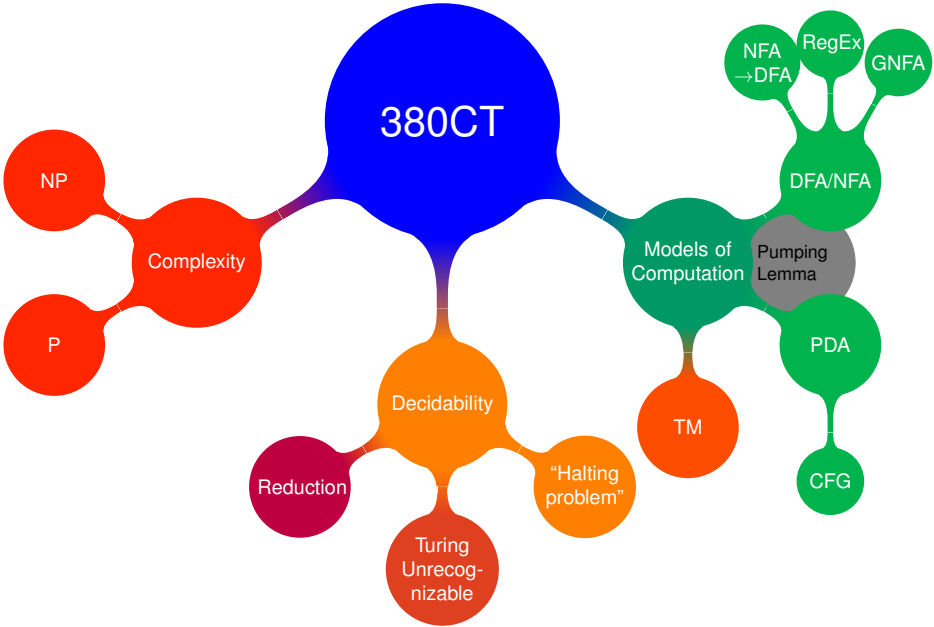

380CT
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
“Halting problem”

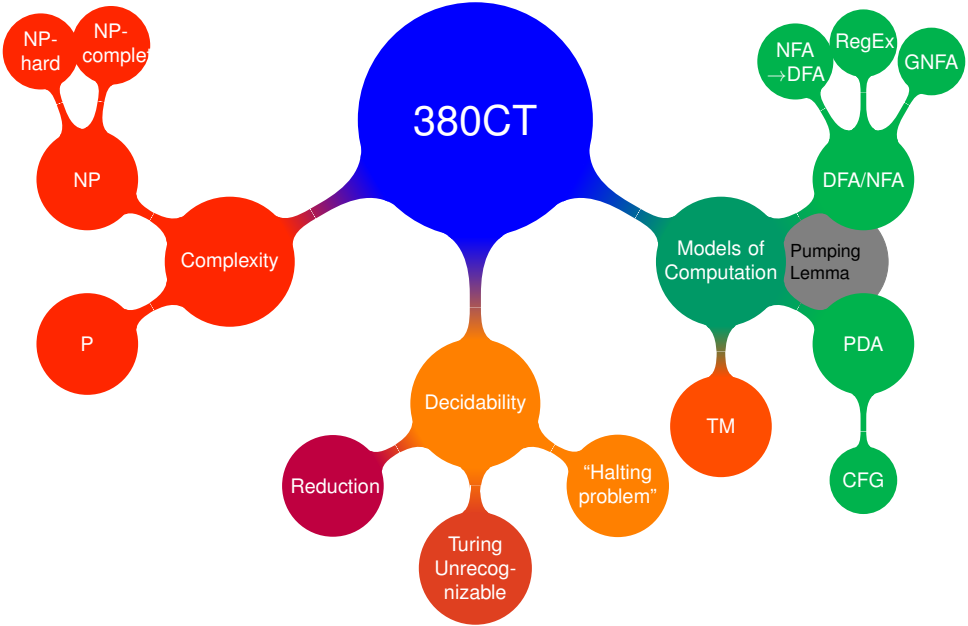

380CT
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM
Decidability
“Halting problem”
Turing Unrecog-nizable

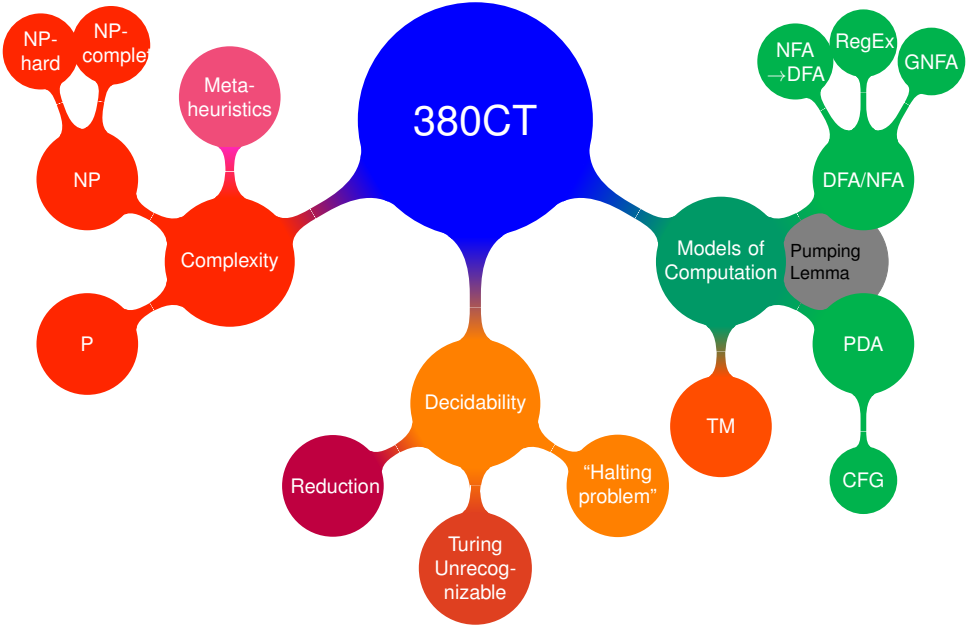

380CT
Models of Computation
DFA/NFA
NFA →DFA
RegEx
GNFA
Pumping Lemma
PDA
CFG
TM
Decidability
Reduction
"Halting problem"
Turing Unrecog-nizable


380CT
Complexity
Decidability
Reduction
Turing Unrecog-nizable
“Halting problem”
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM


380CT
Complexity
P
Decidability
Reduction
Turing Unrecog-nizable
"Halting problem"
Models of Computation
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG
TM


380CT
NP
Complexity
P
Decidability
Reduction
Turing Unrecog-nizable
"Halting problem"
Models of Computation
TM
Pumping Lemma
DFA/NFA
NFA →DFA
RegEx
GNFA
PDA
CFG


380CT
NP-hard
NP-complete
NP
Complexity
P
Decidability
Reduction
Turing Unrecog-nizable
"Halting problem"
Models of Computation
NFA →DFA
RegEx
GNFA
DFA/NFA
Pumping Lemma
PDA
TM
CFG


380CT
NP-hard
NP-complete
NP
Meta-heuristics
Complexity
P
Decidability
Reduction
"Halting problem"
Turing Unrecog-nizable
NFA →DFA
RegEx
GNFA
DFA/NFA
Models of Computation
Pumping Lemma
PDA
TM
CFG

Intended Module Learning Outcomes

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

1. Use **formal** notation to *specify* **patterns** and **languages**.

Intended Module Learning Outcomes

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

1. Use **formal** notation to *specify* **patterns** and **languages**.
2. *Specify* and *simulate* various **automata**.

Intended Module Learning Outcomes

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

1. Use **formal** notation to *specify* **patterns** and **languages**.
2. *Specify* and *simulate* various **automata**.
3. *Explain* the connection between classes of **languages**, **models of computation** and types of **algorithms**.

Intended Module Learning Outcomes

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

1. Use **formal** notation to *specify* **patterns** and **languages**.
2. *Specify* and *simulate* various **automata**.
3. *Explain* the connection between classes of **languages**, **models of computation** and types of **algorithms**.
4. *Classify* the **computability** and **complexity** of real world problems.

Intended Module Learning Outcomes

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

1. Use **formal** notation to *specify* **patterns** and **languages**.
2. *Specify* and *simulate* various **automata**.
3. *Explain* the connection between classes of **languages**, **models of computation** and types of **algorithms**.
4. *Classify* the **computability** and **complexity** of real world problems.
5. *Specify* and *implement* methods to **estimate** solutions to **intractable problems**.

Teaching and Learning

- Lectures: Mon 9am.
- Tutorials/exercises – check your timetable.
- Pen and paper, JFLAP, GeoGebra, Programming.
- Formative tests.

Develop a portfolio of practical exercises.

Laboratory	48 hours	24%
Lecture	24 hours	12%
Self guided	128 hours	64%
Total	200 hours	

Assessment

Assessement: 50% Coursework and 50% Exam.
Resits: second portfolio and exam
Pass requirements:

- Coursework $\geq 35\%$
- **and** Exam $\geq 35\%$
- **and** Module Mark $\geq 40\%$.

Indicative Content

- **Mathematical background** (Review) Sets, functions, relations, propositional logic and predicate calculus. O-notation.

Indicative Content

- **Mathematical background** (Review) Sets, functions, relations, propositional logic and predicate calculus. O-notation.
- **Automata** DFAs, NFAs, PDAs, TMs. Determinism and Nondeterminism. Relationship between automata and classes of languages. Limits of automata (pumping lemma, undecidability and unrecognisability). Simulation package (JFLAP).

Indicative Content

- **Mathematical background** (Review) Sets, functions, relations, propositional logic and predicate calculus. O-notation.
- **Automata** DFAs, NFAs, PDAs, TMs. Determinism and Nondeterminism. Relationship between automata and classes of languages. Limits of automata (pumping lemma, undecidability and unrecognisability). Simulation package (JFLAP).
- **Computability and Complexity** The Church-Turing Thesis, Reduction, P versus NP, NP-completeness, Polynomial time verification, Polynomial time reduction. Search problems and NP-hardness. Overview of further complexity classes (e.g. PSPACE, EXPTIME)

Indicative Content

- **Mathematical background** (Review) Sets, functions, relations, propositional logic and predicate calculus. O-notation.
- **Automata** DFAs, NFAs, PDAs, TMs. Determinism and Nondeterminism. Relationship between automata and classes of languages. Limits of automata (pumping lemma, undecidability and unrecognisability). Simulation package (JFLAP).
- **Computability and Complexity** The Church-Turing Thesis, Reduction, P versus NP, NP-completeness, Polynomial time verification, Polynomial time reduction. Search problems and NP-hardness. Overview of further complexity classes (e.g. PSPACE, EXPTIME)
- **Algorithms and Heuristics** Exhaustive search, Approximation Algorithms, Greedy Algorithms, Metaheuristics. Pseudocode and implementation (C++, Python, . . .).

Books

Essential Reading

Sipser, M. (1997) **Introduction to the Theory of Computation.** 2nd Edn. Thomson Course Technology Inc

Books

Essential Reading

 Sipser, M. (1997) **Introduction to the Theory of Computation.** 2nd Edn. Thomson Course Technology Inc

Recommended Reading

 Harel, D. (2004) **Algorithmics: The Spirit of Computing.** 3rd Edn. Addison Wesley

Books

Essential Reading


Sipser, M. (1997) **Introduction to the Theory of Computation.** 2nd Edn. Thomson Course Technology Inc

Recommended Reading


Harel, D. (2004) **Algorithmics: The Spirit of Computing.** 3rd Edn. Addison Wesley


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

Books

Essential Reading

- Sipser, M. (1997) **Introduction to the Theory of Computation.** 2nd Edn. Thomson Course Technology Inc

Recommended Reading

- Harel, D. (2004) **Algorithmics: The Spirit of Computing.** 3rd Edn. Addison Wesley
- 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

Books

Essential Reading

- Sipser, M. (1997) **Introduction to the Theory of Computation.** 2nd Edn. Thomson Course Technology Inc

Recommended Reading

- Harel, D. (2004) **Algorithmics: The Spirit of Computing.** 3rd Edn. Addison Wesley
- 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
- Hoos, H. and Stutzler, T. (2005) **Stochastic Local Search: Foundations and Applications.** Morgan Kaufmann

Pre-requisites

- **210CT**
  - Algorithms (Searching and sorting, Recursion, Divide and Conquer strategies, Greedy algorithms)
  - Complexity and efficiency (Time and space complexity, Big-O notation)

Pre-requisites

- **124MS**
  - Propositional Calculus (Statements, $\implies$, $\iff$, $\neg, \wedge, \vee, \subset, \supset, \in$, Truth tables. Formal proof.)
  - Predicate calculus (Predicates, $\exists, \forall$)
  - Sets and Functions (Subset, Cardinality, Venn diagrams, Functions, domain and codomain, Composition and inverse. Relations.)
  - Algebra (Congruences)
  - Graph Theory (connectivity, depth and breadth first search, shortest path, Trees)
  - Algorithms (Uncomputable problems. Asymptotic efficiency. Heuristic algorithms)

Polynomials vs Eponentials

Realtime Moodle quiz. . .