THEORY OF COMPUTATION

Semester 1, 2020

Course Overview

Theory of Computation

- In theoretical computer science and mathematics, the theory of computation is the branch that deals with how efficiently **problems** can be solved on a model of computation, using an **algorithm**.

- The field is divided into three major branches:
  - automata theory and language,
  - computability theory, and
  - computational complexity theory,

which are linked by the question:

"What are the fundamental capabilities and limitations of computers?"

FORMAL LANGUAGE

- In mathematics, computer science, and linguistics, a formal language is **a set of strings** of symbols that may be constrained by rules that are specific to it.
- The alphabet of a formal language is the set of symbols, letters, or tokens from which the strings of the language may be formed; frequently it is required to be finite.
- A formal language is often defined by means of a **formal grammar** such as a regular grammar or context-free grammar, also called its formation rule.

AUTOMATA

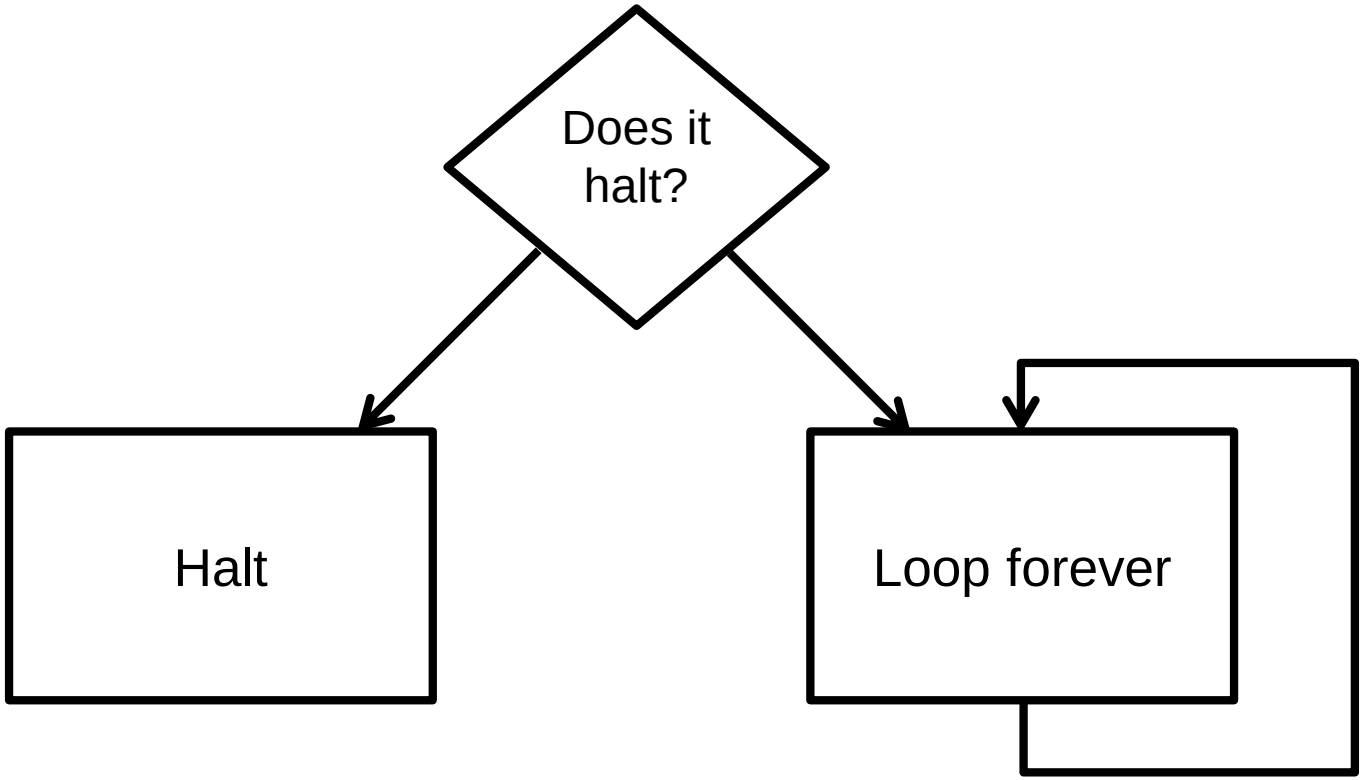
- An automaton is a finite representation of a formal language that may be an infinite set.
- Automata are often classified by the class of formal languages they are able to recognize.
- Automata play a major role in theory of computation, compiler design, artificial intelligence, parsing and formal verification.

Let’s make it simple

FORMAL LANGUAGE AND AUTOMATA

PROBLEMS AND PROGRAMS

What is the course about?



COURSE SUMMARY

Subject Code	COMP2270/COMP6270
Subject Title	Theory of Computation
Homepage	https://uonline.newcastle.edu.au/
Unit Value	10
Assumed knowledge	**SENG1120, MATH1510**

This is an advanced course in your program and the assumed knowledge will be needed to be fully prepared for it

COURSE CONTENT

MAIN TOPICS

(0) Preliminaries
(1) Formal languages
(2) Finite automata and regular languages
(3) Push-down automata and context-free languages
(4) Turing machines and phrase-structured languages; Church-Turing Thesis
(5) Decidability and reducibility
(6) Complexity theory

COURSE OVERVIEW

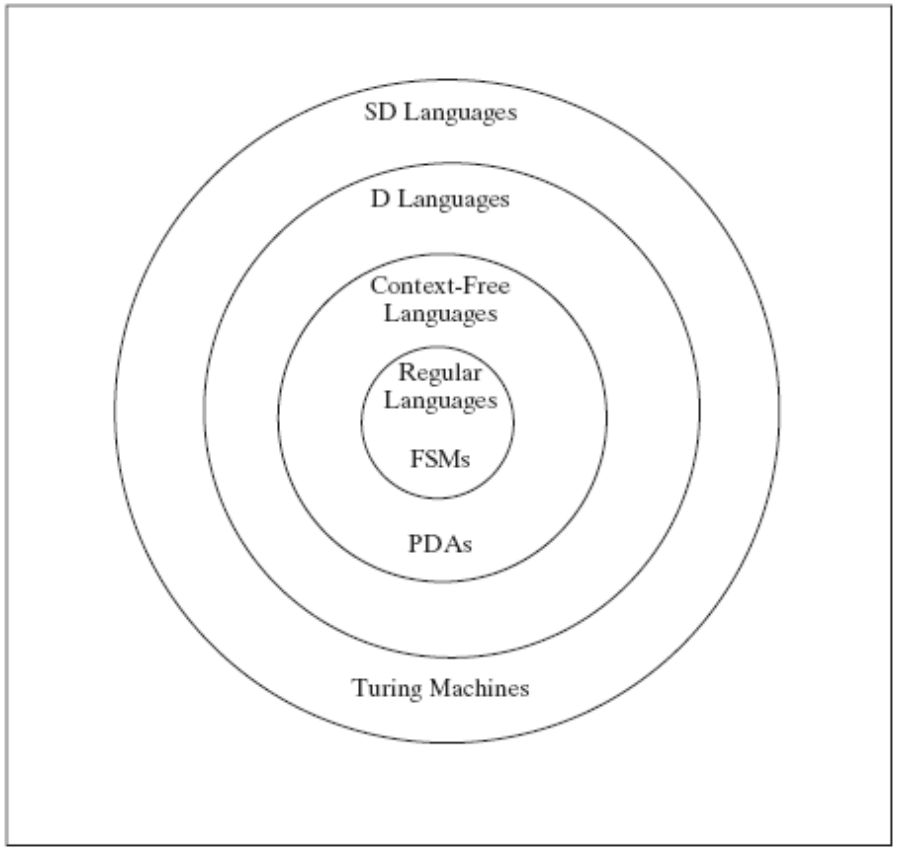
Although the subject matter of this course is **languages and automata, we need a framework**—some support infrastructure—in which to work

SUBJECT MATTER is Automata, Grammars, and Computability. We will see a general overview today.

FRAMEWORK is basic discrete mathematics, *i.e., some* set theory, some logic, and some proof techniques. We will devote most of this week to revise what you need to know for this course.

COURSE OVERVIEW

A little bit more about the subject matter: The hierarchy



- We will study this hierarchy of language classes from the inside out
- This will give us a full notion of what is *computable*

Lecture Plan

Weekly program (lectures)

- Week 1 – Background knowledge revision: logic, sets, proof techniques
- Week 2 – Languages and strings. Hierarchies. Computation. Closure properties
- Week 3 – Finite State Machines: non-determinism vs. determinism
- Week 4 – Regular languages: expressions and grammars
- Week 5 – Non regular languages: pumping lemma. Closure
- Week 6 – Context-free languages: grammars and parse trees
- Week 7 – Pushdown automata
- Week 8 – Non context-free languages: pumping lemma and decidability. Closure
- Week 9 – Decidable languages: Turing Machines
- Week 10 – Church-Turing thesis and the unsolvability of the Halting Problem
- Week 11 – Decidable, semi-decidable and undecidable languages (and proofs)
- Week 12 – Revision of the hierarchy and Complexity theory
- Week 13 – Extra revision (if needed)

Tutorial Plan

Weekly program (tutorials)

- ❑ Week 2 – Languages and strings. Hierarchies. Computation. Closure properties
- ❑ Week 3 – Finite State Machines
- ❑ Week 4 – non-determinism vs. Determinism
- ❑ Week 5 – Regular languages: expressions and grammars
- ❑ Week 6 – Non regular languages: pumping lemma. Closure
- ❑ Week 7 – Context-free languages: grammars and parse trees
- ❑ Week 8 – Pushdown automata
- ❑ Week 9 – Non context-free languages: pumping lemma and decidability. Closure
- ❑ Week 10 – Decidable languages: Turing Machines
- ❑ Week 11 – Decidable, semi-decidable and undecidable languages (and proofs)
- ❑ Week 12 – Review

Why do we study THOERY OF COMPUTATION?

- You are about to embark on the study of a fascinating and important subject: the **theory of computation**
  - Fundamental mathematical properties of computer hardware, software, and certain applications of these
  - Seek to determine
    - What can and cannot be computed
    - Which type of computational model can be used

There are obvious connections with engineering practice, but also purely philosophical aspects

Why do we study THOERY OF COMPUTATION?

- Computer technology changes quickly
- Specific technical knowledge becomes outdated in a few years
- Studying theory trains you in the abilities to think, express yourself clearly and precisely, solve problems, know when you have not solved a problem
  - These abilities have lasting value

Why do we study THOERY OF COMPUTATION?

- Provides conceptual tools that practitioners use in computer engineering

  - FSMs for parity checkers, vending machines, communication protocols, and building security devices.
  - Interactive games as nondeterministic FSMs.
  - Programming languages, compilers, and context-free grammars.
  - Natural languages are mostly context-free. Speech understanding systems use probabilistic FSMs.
  - Computational biology: DNA and proteins are strings.
  - The undecidability of a simple security model.
  - Artificial intelligence: the undecidability of first-order logic.

CONTACTS

Course Coordinator & Lecturer	Nasimul Noman Room: CIT361 Email: Nasimul.Noman@newcastle.edu.au
Contact Hours	Wednesday; 10:00~12:00 / by previous appointment
Tutor	Mohammad Chowdhury Email: Mohammad.Chowdhury@newcastle.edu.au
Contact Hours	By previous appointment

TIMETABLE

Callaghan Campus Timetable
COMP2270/COMP6270
Theory of Computation
Enquiries: School of Electrical Engineering and Computing
Semester 1 – 2020

Lecture	Monday	11:00 - 13:00	[V107]	Week1-12 (13)
Workshop	Wednesday	08:00 - 10:00	[AVG14]	Wks 2-12 only.
Or Workshop	Friday	12:00 - 14:00	[EA102]	Wks 2-12 only.

Delivery Mode: Semi-Flipped

- **PPP**
  - **P: Prepare**
    - Every week a couple of videos in the BB
    - **Watch those before you attend the lectures (workshop)**
  - **P: Participate**
    - Get engaged in the lectures/tutorials
  - **P: Practice**
    - Practice the problems in the tutorials
    - Additional videos to be released to supplement your learning

Test every week

- **Test your learning as you go**
  - Weekly quiz – starting from **week 1**!!
  - Complete the weekly quiz by the following week
  - Released after lecture (at 5PM)
  - 2 weeks to complete
  - Get 5% credit !!!

Blackboard Collaborate Ultra

- A real-time conferencing tool that provides a dynamic learning environment
    - **Based on requirement**
    - **Previous appointment/arrangement needed**

ASSESSMENT

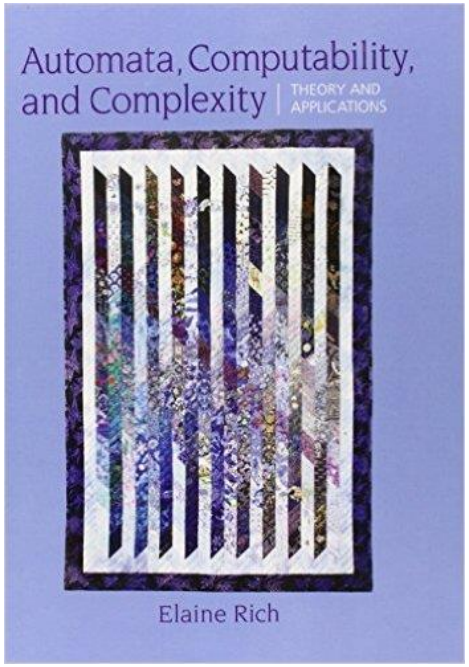
Assessment Item and Description	Method of submission	Due date	Weighting	Item Returnable (Y/N)
Online MCQ	Online through Blackboard	Every week	5%	**N**
Assign 1	Electronically through Blackboard	Week 5 **(29/03/2020)**	15%	**Y**
Assign 2	Electronically through Blackboard	Week 9 **(10/05/2020)**	15%	**Y**
Midterm test 1	In class test	Week 7 **(06/04/2020)**	10%	**Y**
Midterm test 2	In class test	Week 11 **(18/05/2020)**	15%	**Y**
Final exam	**Formal Examination process**	**Exam Period**	**40%**	**N**

TEXTBOOK

Automata, Computability and Complexity
Theory and Applications

By Elaine Rich

Pearson 1 edition



Questions/Queries?