Revision

Dr. Yuan Yao

University of Nottingham Ningbo China (UNNC)

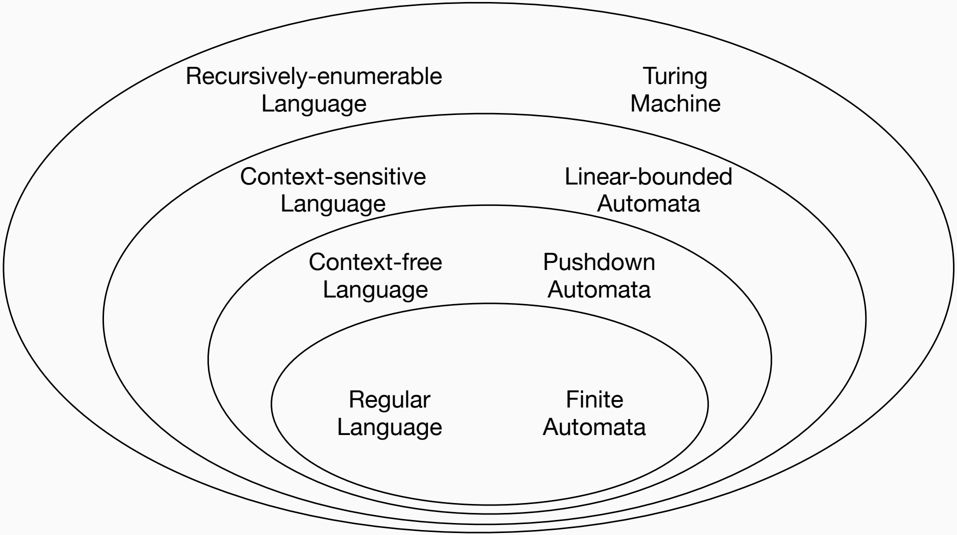
Exam Preparation

Exam Information

- Time allowed: 2 hours
- Total Marks: 100
- 75% of the final marks.
- Four Questions.
- Question Types:
  - True of False.
  - Knowledge: Concepts and definitions.
  - Proof.
  - Draw transition graphs.
  - ...

Content

Language and Computation



Language and Computation

Languages:

- Regular Languages.
- Context-Free Languages.
- Unrestricted Languages.

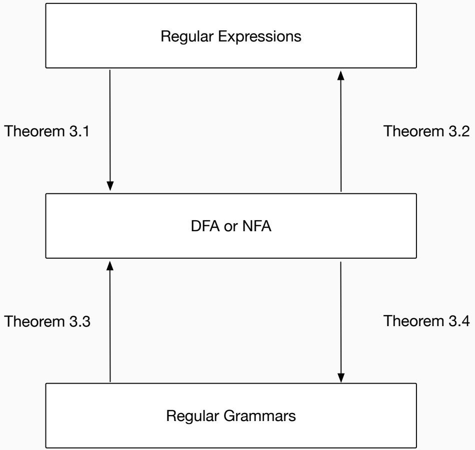
Models of Computation (Automaton):

- Finite Automata.
- Pushdown Automata.
- Turing Machine.

Regular Languages

Three ways of representing the regular languages:

- Finite Automaton:
  - DFA and NFA.
  - E.g. $M = \{Q, \Sigma, \delta, q_0, F\}$
- Regular Expression.
  - It is unique to RL.
  - E.g., $a^* + b$
- Regular Grammar.
  - $G = (V, T, S, P)$, all productions are right-linear or all productions are left-linear.



Deterministic Finite Automata

- A **deterministic finite automata (DFA)** is defined by a 5-tuple:

$$M = (Q, \Sigma, \delta, q_0, F)$$

  where

  - **Q** is a finite set of **internal states**.
  - Σ is a finite set of symbols called the **input alphabet**.
  - $\delta : Q \times \Sigma \to Q$ is a total function called the **transition function**.
  - $q_0 \in Q$ is the **initial state**.
  - $F \subseteq Q$ is the set of **final states**.

Nondeterministic Finite Automaton (NFA)

- Nondeterminism means a choice of moves.
- Formally, a **nondeterministic finite automaton** is defined as a 5-tuple:

$$M = (Q, \Sigma, \delta, q_0, F)$$

  where Q, Σ, q_0 and F are defined as for DFA, but

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \to 2^Q$$

- Major differences between NFA and DFA:
  - In an NFA, the transition function returns **a subset** of Q rather than a single element in Q, e.g.,

$$\delta(q_1, a) = \{q_0, q_2\}$$

  - δ is can be a **partial function**.
  - δ accepts λ as input, with which an NFA may change states without consuming input.

Properties of Regular Languages

- **Clousure:** If L_1 and L_2 are regular languages, then so are $L_1 \cup L_2$, $L_1 \cap L_2$, L_1L_2, $\overline{L_1}$, L_1^*. We say that the family of regular languages is closed under union, intersection, concatenation, complementation and star-closure.

- **Membership:** Given a standard representation of any regular language L on L and $w \in \Sigma^*$, there exists an algorithm to determine whether or not $w \in L$.

- **Empty, Finite, Infinite:** There exists an algorithm for determining whether a regular language, given in the standard representation, is empty, finite or infinite.

- **Non-Regular:** Pigeonhole Principle, Pumping Lemma.

- Definition: We call $G = (V, T, S, P)$ a **context-free grammar (CFG)** if all the productions in P have the form:

$$A \to x$$

  in which $A \in V$, and $x \in (V \cup T)^*$.

- The left-hand side of each production is a single variable, where there is no restrictions on the right-hand side.
- We say that L is a **context-free language (CFL)** if and only if there is a context-free grammar G such that $L = L(G)$, that is, L is generated by G.

Context-Free Grammars

- What is a derivation?
- How to write a derivation for a given string?
- Leftmost and rightmost derivation?
- What is a derivation tree?
- The parsing problem.
- The membership problem.
    - Exhaustive parsing.
- What is ambiguity?

Nondeterministic Pushdown Automata

- A *nondeterministic pushdown automaton (NPDA)* $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ is defined by:
  - Q : the finite set of internal states of the control unit.
  - Σ : the finite set of input alphabet.
  - Γ : the finite set of stack alphabet.
  - δ : the transition function with the type signature:

    $$Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to P_f(Q \times \Gamma^*)$$

    where $P_f(Q \times \Gamma^*)$ is the set of *finite subsets* of $(Q \times \Gamma^*)$
  - $q_0 \in Q$: the initial state of the control unit.
  - $z \in \Gamma$: the stack start symbol.
  - $F \subseteq Q$: the set of final states.

Instantaneous Descriptions

- While transition graphs are convenient for describing NPDAs, they are not so suitable for formal reasoning.
- To trace the operation of an NPDA, we must keep track of:
    1. the current state of the control unit
    2. the unread part of the input string
    3. and the stack contents
- **Instantaneous Description**: The triplet (q, w, u) in which:
    1. q is the state of the control unit
    2. w is the unread part of the input string
    3. and u is the stack contents, with the leftmost symbol indicating the top of the stack

  is called an instantaneous description of a pushdown automaton.

- How to convert a CFG to a corresponding NPDA?
    - Greibach Normal Form.
- What is a Deterministic Pushdown Automaton?
- The relationship between NPDA and DPDA.

Definition of a Turing Machine

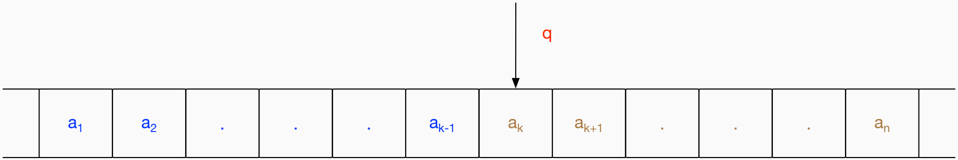
- **Definition 9.1:** A Turing Machine $M = \{Q, \Sigma, \Gamma, \delta, q_0, \square, F\}$ is defined by:
  - Q : a finite set of internal states.
  - Σ : the input alphabet.
  - Γ : the tap alphabet.
  - $\delta : Q \times T \to Q \times \Gamma \times \{L, R\}$: the transition function.
  - $\square \in \Gamma$: a special symbol called the blank.
  - $q_0 \in Q$: the initial state.
  - $F \subseteq$ the set of final states.
- In the definition of a Turing machine, we assume that:

$$\Sigma \subseteq \Gamma - \{\square\}$$

Instantaneous Description

- An instantaneous description of a machine in state q with the tape depicted in the figure below is as follows:

$$a_1 a_2 \ldots a_{k-1} q a_k a_{k+1} \ldots a_n$$



Turing Machine as Transducers

- Transducers: transforms input into output.
- A Turing machine transducer implements a function that treats the original contents of the tape as its input and the final contents of the tape as its output.
- Turing machines are the most powerful model of computation as transducers as well:
  - Arithmetic operators, Exponentiation, Integer logarithm;
  - Comparison;
  - String manipulation;
  - ...

Turing Machine

- Combining Turing Machine.
- Universal Turing Machine.
- What is the Church-Turing Thesis?
- Computability and Decidability.
- The halting problem.