Theory of Computation

ChanBong

February 25, 2023

Contents

January 6, 2023

Lecture 1: Introduction

1.1 Basic Defination

To get started, we will define what is computation and various jargons we'll be using throughout this course

Definition 1.1 Strings: set of alphabets. Denoted by Σ

Example 1.1: Some examples of strings are:

- $u = aabbca$
- $v = bac$
- $uv = aabbcca$

Definition 1.2 Length of a string: Number of alphbets in the string

Empty strings are denoted by λ.

- $|\lambda| = 0$.
- $\lambda w = w\lambda = w$.

If $w = uv$ then u is called **prefix** of w and v is called **suffix** of w.

Example 1.2: Consider $w = abcab$. Then $\{\lambda, a, ab, abc, abca, abcab\}$ are all prefixes of w and $\{\lambda, b, bc, bca, bcab\}$ are all suffixes of w.

January 9, 2023

Lecture 2: Grammer

Theorem 2.1: $|uv| = |u| + |v|$.

Proof: We will use mathematical induction to prove this.
Base case: $|\lambda v| = |v| = |\lambda| + |v|$.
Inductive step: Assume $|uv| = |u| + |v|$. Then $|u'v| = |u'| + |v| = |u| + |v| + 1$.
Thus, $|uv| = |u'v| = |u| + |v| + 1 = |u| + |v|$.
Thus, $|uv| = |u| + |v|$.
□

2.1 Grammer

Definition 2.1 Grammer: A grammer $\mathbb{G} = \{\mathbb{V}, \mathbb{T}, \mathbb{S}, \mathbb{P}\}$

- $\mathbb{V}$ is finite set of objects called variables.
- $\mathbb{T}$ is finite st of objects called terminals.
- $\mathbb{S} \in \mathbb{V}$: special symbol called start variable
- $\mathbb{P}$ is a finite set of rules

January 10, 2023

Lecture 3: Deterministic Finite Acceptors

3.1 Language from grammer

Example 3.1: Consider the grammer $\mathbb{G} = (\{s\}, \{a, b\}, \mathbb{S}, \mathbb{P})$; $\mathbb{P}_1\mathbb{S} \to a\mathbb{S}b/\lambda$ then language generated by this grammer is

$$\mathbb{L}(G_1) = \{a^m b^m : m \geq 0\}$$

Example 3.2: Consider the grammer $\mathbb{G} = (\{s, \mathbb{A}\}, \{a, b\}, \mathbb{S}, \mathbb{P})$; $\mathbb{P}_2\mathbb{S} \to a\mathbb{S}b/\lambda, \mathbb{A} \to a\mathbb{A}b/\lambda$ then language generated by this grammer is

$$\mathbb{L}(G_2) = \{a^m b^m : m \geq 0\}$$

Definition 3.1 : Two grammers are equivalent iff $L(G_1) = L(G_2)$

3.2 Deterministic Finite Acceptors (DFA)

Definition 3.2 : A DFA is defined by $\mathbb{M} = (\mathbb{Q}, \Sigma, \mathbb{D}, \mathbb{Q}_0, \mathbb{F})$ where

- $\mathbb{Q}$ is a finite set of states
- Σ is a finite set of input symbols
- $\mathbb{D} : \mathbb{Q} \times \mathbb{E} \to \mathbb{Q}$ is a transition function
- $\mathbb{Q}_0 \in \mathbb{Q}$ is the start state
- $\mathbb{F} \subseteq \mathbb{Q}$ is the set of final states

Example 3.3: Consider the DFA $\mathbb{M} = (\{q_0, q_1, q_2\}, \{a, b\}, \mathbb{D}, q_0, \{q_1\})$. $\mathbb{D}(q_0, a) = q_1, \mathbb{D}(q_0, b) = q_0, \mathbb{D}(q_1, a) = q_2, \mathbb{D}(q_1, b) = q_0, \mathbb{D}(q_2, a) = q_2, \mathbb{D}(q_2, b) = q_2$

3.3 Graphical Representation of DFA

Definition 3.3 : A DFA is represented by a directed graph $G = (V, E)$ where

- $V = \mathbb{Q}$
- $E = \{(q, \mathbb{D}(q, \mathbb{E}), \mathbb{E}) : q \in \mathbb{Q}, \mathbb{E} \in \mathbb{E}\}$
- an open $\rightarrow$ indicates the start state
- $\bigcirc$ indicates the final state

Example 3.4: Consider the DFA $\mathbb{M} = (\{q_0, q_1, q_2\}, \{a, b\}, \mathbb{D}, q_0, \{q_1\})$. $\mathbb{D}(q_0, a) = q_1, \mathbb{D}(q_0, b) = q_0, \mathbb{D}(q_1, a) = q_2, \mathbb{D}(q_1, b) = q_0, \mathbb{D}(q_2, a) = q_2, \mathbb{D}(q_2, b) = q_2$

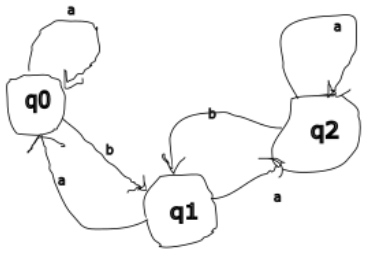

Figure 3.1: DFA

January 12, 2023

Lecture 4: DFA Acceptance

4.1 Walk in DFA

Definition 4.1 Trap State: A state q is a trap state iff $\mathbb{D}(q, \mathbb{E}) = q$ for all $\mathbb{E} \in \mathbb{E}$

Example 4.1: $\mathbb{M} = (\{q_0, q_1, q_2\}, \{a, b\}, \mathbb{D}, q_0, \{q_1\})$. $L(M) = \{a^n b : n \geq 0\}$

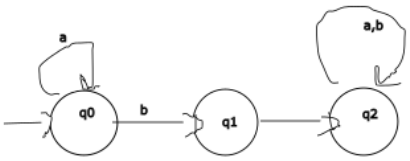

Figure 4.1: DFA

Here q_2 is a trap state.

Theorem 4.2: Let $\mathbb{M}$ be a DFA and let G_m be its graphical representation. Then for every $q_i \in Q$ and $w \in \Sigma^+$ we have $D^+(q_i, w) = q_j$ iff there is a walk from q_i to q_j with label w.

Proof: We will use mathematical induction on the length of w.
Base case: $w = a$. Then $D^+(q_i, a) = q_j$ iff there is a edge from q_i to q_j with label a.
Inductive step: Assume for all $|u| = n < m$, if $D^+(q_i, u) = q_k$ then there is a walk form q_i to q_j with label u.
Now if $w \in \Sigma^+$, with $|w| = n + 1 = m$, say $w = ua$.
Then $D^+(q_i, w) = D^+(D^+(q_i, v), a) = D^+(q_k, a) = q_j \implies$ there is a walk from q_i to q_j with label w.

Backward direction: Assume for all $|u| = n < m$, if there is a walk from q_i to q_j with label u then $D^+(q_i, u) = q_j$.
□

Remark: If there is a walk from q_i to q_j with label w, then $D^+(q_i, w) = q_j$

4.2 DFA Acceptance

Definition 4.2 : A DFA accepts a string w iff w is accepted by $\mathbb{M}$.

Definition 4.3 Regular Language: A language L is called regular iff there is a DFA $\mathbb{M}$ such that $L(M) = L$.

Example 4.3: Find a DFA for the language $L = \{abw : w \in \{a, b\}\}$.

Figure 4.2: Solution to the above example

Example 4.4: Show that the language $L = \{awa : w \in \{a, b\}^+\}$ is regular.

Figure 4.3: Solution to the above example

$\oint$ How to prove that a language is not regular?

Definition 4.4 Concatanation of two languages:

January 16, 2023

Lecture 5: Non Deterministic Fnite Automata

Continuing Example 4.4

Figure 5.1: Solution to the above example

Define $L^2 = \{; w_1, w_2 \in \{a, b\}^+\}$

Figure 5.2: Solution to the L^2 example

5.1 Non Deterministic Fnite Automata (NFA)

Inconvinences in DFA :

- We needed to introduce all the states wheather or not they serve any meaningful purpose.
- in DFA, transition function is a complete function.

- We can't leave arrows hanging.

Definition 5.1 NFA: An NFA is defined by $\mathbb{M} = (\mathbb{Q}, \Sigma, \mathbb{D}, \mathbb{Q}_0, \mathbb{F})$ where $\mathbb{Q}, \Sigma, \mathbb{Q}_0, \mathbb{F}$ are same as in DFA, but

$$\mathbb{D} : \mathbb{Q} * (\Sigma \cap \{ \lambda\}) \rightarrow 2^Q \text{ Power set of Q}$$

Example 5.1: $\mathbb{D}(q_1, a) = \{q_2, q_3\}$ and $\mathbb{D}(q_i, \lambda) = \{q_i, q_j\}$

- can have empty strings now
- machine can move from one state to another without processing any input
- any state with a path label w can end up in a subset of final states

5.2 NFA Acceptance

Example 5.2: Consider the following NFA

Figure 5.3: Solution to the above example

Definition 5.2 NFA Acceptance: The Language accepted by NFA M is $L(M) = \{w \in \Sigma^*, \mathbb{D}^*(q_0, w) \bigcap F \neq \phi\}$

Remark It may end up in a non-final state but it must have atleast one element from final state.

Lanuguage accepted by Example 5.2 is $L(M) = \{a^3\} \cap \{a^{2n} : n \geq 1\}$

Definition 5.3 Extended Transition function: $\mathbb{D}^* is defined \mathbb{D}^*(q_i, w)$ contains q_j iff there is a walk from q_i to q_j laveled w in transition graph.

Example 5.3: Consider this NFA
$L(M) = \{(10)^n : n \geq 0\}$

Figure 5.4: Solution to the above example

Figure 5.5: Solution to the above example

Definition 5.4 Dead Configuration: A state from which you can't go anywhere. It has no outgoing arrow

Definition 5.5 Equivalent Automata: If two automation accept the same language.

$$M_1 \approx M_2 \text{ if } L(M_1) = L(M_2)$$

5.3 Convert from DFA to NFA

$\oint$ Find a DFA that accepts $L(M_2)$?

19th January

Lecture 7: NFA to DFA Convergance

Example 6.1: Convert the following NFA to DFA

Figure 6.1: DFA

23rd January

Lecture 8: NFA to DFA Convergance

Theorem 7.1: The procedure **Mark** applicated to any DFA $M = \{\mathbb{Q}, \Sigma, \mathbb{D}, q_0, \mathbb{F}\}$ terminates and determines all pairs of distinguishable states.

7.1 Procedure "Reduce"

Given DFA $M = \{\mathbb{Q}, \Sigma, \mathbb{D}, q_0, \mathbb{F}\}$ construct a reduces DFA $\hat{M} = \{\hat{\mathbb{Q}}, \Sigma, \hat{\mathbb{D}}, \hat{q_o}, \hat{\mathbb{F}}\}$ as follows

- Use procedure "Mark" to generate equivalance classes.
- For each equivalance class $\{q_1 \dots q_k\}$ of indistinguishable states create a state with label $i_1 \dots i_k$ for $\hat{\mathbb{M}}$
- For each transition rule of $\mathbb{M}$ fo the form $\mathbb{D}(q_r, a) = q_s$, find the label $r \in \{i_1 \dots i_k\}$ add rule $\hat{\mathbb{D}}(i_1 \dots i_k, u) = j$ and j_t in $\hat{\mathbb{M}}$.
- The initial state $\hat{q_0}$ is that state which contains 0, $q_0 \in \{q_1 \dots q_k\}$
- The $\hat{\mathbb{F}}$ will contain all those states, whose label contains the $q_i \in \mathbb{F}$

Example 7.2: Convert this using Procedure "Reduce"

Figure 7.1: DFA

Theorem 7.3: Given any DFA $M = \{\mathbb{Q}, \Sigma, \mathbb{D}, q_0, \mathbb{F}\}$ the procedure "Reduce" produces another DFA $\hat{M}$ such that $L(M) = L(\hat{M})$. Furthermore $\hat{\mathbb{M}}$ is minimal in the sense that there is no DFA with less number of states than in $\hat{M}$, which accepts $L(M)$

February 25, 2023

Revision

8.1 Regular Expressions

Finite Langugaes

- Are Regular
- have a finite automata
- have a regular expression