Chapter 1

Regular Languages

1.1 Deterministic Finite State Automata

Definition 1.1.1. An **alphabet** Σ is a finite set of symbols.

- A **string** over Σ is a finite sequence of symbols from Σ.
- The **length** of a string w, denoted by $|w|$, is the number of symbols it contains.
- The string of length 0 is called the **empty string**, denoted by ϵ.

Definition 1.1.2. Let Σ be an alphabet.

- For any nonnegative integer n, Σ^n denotes the set of words of length n.
- Σ^* denotes the set of all strings over Σ.
- A **language** over Σ is a subset of Σ^*.

Definition 1.1.3. A **deterministic finite state automaton** (DFA) is a system $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$, where each component is as follows.

- Σ is the alphabet.
- Q is a finite set of **states**.
- $q_0 \in Q$ is the **initial** state.
- $F \subseteq Q$ is the set of **accepting** states.
- δ is the **transition function** from $Q \times \Sigma$ to Q.

Definition 1.1.4. The **run** of DFA $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ on an input string $w = a_1 \cdots a_n$ over Σ is the sequence of states

$$r = (r_0, r_1, \dots, r_n)$$

where $r_0 = q_0$ and $\delta(r_{i-1}, a_i) = r_i$ for each $i \in \{1, \dots, n\}$.

- r is an **accepting** run if $r_n \in F$.
- We say that $\mathcal{A}$ **accepts** w if the run of $\mathcal{A}$ on w is an accepting run.

- The language of all strings accepted by $\mathcal{A}$ is denoted by $L(\mathcal{A})$.
- A language L is **regular** if there is a DFA $\mathcal{A}$ with $L = L(\mathcal{A})$.

Remark.

- For DFA $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$, the empty string ϵ is accepted by $\mathcal{A}$ if and only if $q_0 \in F$.

1.2 Nondeterministic Finite State Automata

Definition 1.2.1. A **nondeterministic finite state automaton** (NFA) is a system $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$, where each component is as follows.

- Σ is the alphabet.
- Q is a finite set of **states**.
- $q_0 \in Q$ is the **initial** state.
- $F \subseteq Q$ is the set of **accepting** states.
- $\delta \subseteq Q \times \Sigma \times Q$ is the **transition relation**.

Definition 1.2.2. A **run** of NFA $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ on an input string $w = a_1 \cdots a_n$ over Σ is the sequence of states

$$r = (r_0, r_1, \dots, r_n)$$

where $r_0 = q_0$ and $(r_{i-1}, a_i, r_i) \in \delta$ for each $i \in \{1, \dots, n\}$.

- r is an **accepting** run if $r_n \in F$.
- We say that $\mathcal{A}$ **accepts** w if there is an accepting run of $\mathcal{A}$ on w.
- The language of all strings accepted by $\mathcal{A}$ is denoted by $L(\mathcal{A})$.

Theorem 1.2.3. For every NFA $\mathcal{A}$, there is a DFA $\widehat{\mathcal{A}}$ with $L(\mathcal{A}) = L(\widehat{\mathcal{A}})$.

Proof. Let $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$. We construct $\widehat{\mathcal{A}} = (\Sigma, \widehat{Q}, \widehat{q_0}, \widehat{F}, \widehat{\delta})$ as follows.

- $\widehat{Q} = \mathcal{P}(Q)$.
- $\widehat{q_0} = \{q_0\}$.
- $\widehat{F} = \{\widehat{q} \in \widehat{Q} : q \in \widehat{q} \text{ for some } q \in F\}$.
- $\widehat{\delta} : \widehat{Q} \times \Sigma \to \widehat{Q}$ is the transition function such that

  $$\widehat{\delta}(\widehat{q}, a) = \{q \in Q : (p, a, q) \in \delta \text{ for some } p \in \widehat{q}\}.$$

  holds for each $\widehat{q} \in \widehat{Q}$ and $a \in \Sigma$.

Now we prove that $L(\mathcal{A}) = L(\widehat{\mathcal{A}})$. For $w \in \Sigma^*$, let $\widehat{r} = (\widehat{r}_0, \widehat{r}_1, \dots, \widehat{r}_n)$ be the run of $\widehat{\mathcal{A}}$ on w.

(i) Suppose that $r = (r_0, r_1, \dots, r_n)$ is an accepting run of $\mathcal{A}$ on w, and we prove that $\widehat{r}$ is an accepting run on w. It is obvious that $r_0 \in \widehat{r}_0$. If $r_{i-1} \in \widehat{r}_{i-1}$ for some $i \in \{1, \dots, n\}$, then we have $r_i \in \widehat{\delta}(\widehat{r}_{i-1}, a_i) = \widehat{r}_i$ since $(r_{i-1}, a_i, r_i) \in \delta$. Thus, $r_n \in \widehat{r}_n$, and it follows that $\widehat{r}_n \in \widehat{F}$. Therefore, we have $L(\mathcal{A}) \subseteq L(\widehat{\mathcal{A}})$.

(ii) Suppose that $\widehat{r}$ is an accepting run. Then due to the construction of $\widehat{F}$ and $\widehat{\delta}$, we can construct an accepting run $r = (r_0, r_1, \dots, r_n)$ of $\mathcal{A}$ on w as follows.

- Let r_n be a state in $\widehat{r}_n \cap F$.
- For $i \in \{0, \dots, n-1\}$, let r_i be a state in $\widehat{r}_i$ such that $(r_i, a_{i+1}, r_{i+1}) \in \delta$.

Thus, we have $L(\widehat{\mathcal{A}}) \subseteq L(\mathcal{A})$, which completes the proof. □

1.3 Regular Expressions

Definition 1.3.1. Let Σ be an alphabet. A **regular expression** over Σ is a string in the minimal language over $\Sigma \cup \{\varnothing, \epsilon, {}^*, \cdot, \cup, (,)\}$ that satisfies the following conditions.

1. $\varnothing$ is a regular expression.
2. ϵ is a regular expression.
3. If $a \in \Sigma$, then a is a regular expression.
4. If e_1 and e_2 are regular expressions, then so is $(e_1 \cdot e_2)$.
5. If e_1 and e_2 are regular expressions, then so is $(e_1 \cup e_2)$.
6. If e is a regular expression, then so is $(e)^*$.

Definition 1.3.2. A regular expression e over an alphabet Σ defines a language $L(e)$ as follows.

1. $L(\varnothing) = \varnothing$.
2. $L(\epsilon) = \{\epsilon\}$.
3. $L(a) = \{a\}$ for each $a \in \Sigma$.
4. $L((e_1 \cdot e_2)) = L(e_1) \cdot L(e_2)$ for each regular expressions e_1 and e_2.
5. $L((e_1 \cup e_2)) = L(e_1) \cup L(e_2)$ for each regular expressions e_1 and e_2.
6. $L((e)^*) = L(e^*)$ for each regular expression e.

Remark. We will write $(e_1 e_2)$ instead of $(e_1 \cdot e_2)$ for simplification. Furthermore, we may omit parentheses if there is no ambiguity.

Theorem 1.3.3. Let Σ be an alphabet. A language L over Σ is regular if and only if there is a regular expression e over Σ such that $L = L(e)$.

Proof. ($\Leftarrow$) It holds trivially since all finite languages are regular, and regular languages are closed under union, cancatenation and star operations.

($\Rightarrow$) Since L is regular, there is a DFA $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ with $L = L(\mathcal{A})$. For $p, q \in Q$ and $S \subseteq Q$, define $L(p, S, q)$ to be the language of strings $w \in \Sigma^*$ such that the run r of $w = a_1 \cdots a_n$ on $\mathcal{A}$ with

$$r = (r_0, r_1, \ldots, r_{n-1}, r_n)$$

satisfying $r_0 = p$, $r_n = q$ and $r_i \in S$ for each $i \in \{1, \ldots, n-1\}$. Then it suffices to prove that there exists a regular expression e with $L(e) = L(p, S, q)$ for each $p, q \in Q$ and $S \subseteq Q$ since

$$L = \bigcup_{q \in F} L(q_0, Q, q).$$

The proof is by induction on $|S|$. For the induction basis, let $|S| = 0$, i.e., $S = \varnothing$. Then we have

$$L(p, \varnothing, p) = \{\epsilon\} \cup \{a \in \Sigma : \delta(p, a) = p\}$$

and

$$L(p, \varnothing, q) = \{a \in \Sigma : \delta(p, a) = q\}$$

for all $p, q \in Q$ with $p \neq q$, and thus the induction basis is proved. Now assume the induction hypothesis that the statement holds for $|S| = k$, and we prove that it is true for $|S| = k+1$. Let s be an arbitrary state in S and let $S' = S \setminus \{s\}$. Then it suffices to prove that

$$L(p, S, q) \quad = \quad L(p, S', q) \quad \cup \quad L(p, S', s) \cdot (L(s, S', s))^* \cdot L(s, S', q).$$

since $|S'| = k$.

(i) It is obvious that

$$L(p, S, q) \quad \supseteq \quad L(p, S', q) \quad \cup \quad L(p, S', s) \cdot (L(s, S', s))^* \cdot L(s, S', q)$$

since $S = S' \cup \{s\}$.

(ii) Then we prove that

$$L(p, S, q) \quad \subseteq \quad L(p, S', q) \quad \cup \quad L(p, S', s) \cdot (L(s, S', s))^* \cdot L(s, S', q).$$

Suppose that $w \in L(p, S, q)$. Let $i_1, i_2, \ldots, i_\ell$ be all indices in $\{1, \ldots, n-1\}$ such that $r_{i_j} = s$ for each $j \in \{1, \ldots, \ell\}$, where $i_1 < i_2 < \cdots < i_\ell$.

If $\ell = 0$, then $w \in L(p, S', q)$ since $r_i \neq s$ for all $i \in \{1, \ldots, n-1\}$. Otherwise, we have

$$\begin{aligned} a_1 \cdots a_{i_1} &\in L(p, S', s) \\ a_{i_1+1} \cdots a_{i_2} &\in L(s, S', s) \\ &\vdots \\ a_{i_{\ell-1}+1} \cdots a_{i_\ell} &\in L(s, S', s) \\ a_{i_\ell+1} \cdots a_n &\in L(s, S', q), \end{aligned}$$

and thus $w \in L(p, S', s) \cdot (L(s, S', s))^* \cdot L(s, S', q)$. This completes the proof. $\square$