Chapter 1

Regular Languages

1.1 Languages

Definition 1.1. An **alphabet** is a finite nonempty set of symbols.

Definition 1.2. Let Σ be an alphabet.

- A **string** over Σ is a finite sequence of symbols from Σ. The collection of all strings over Σ is denoted by Σ^*.
- The **length** of a string w, denoted by $|w|$, is the number of symbols it contains.
- The string containing no symbols is called the **empty string**, denoted by ϵ.

Definition 1.3. A subset of Σ^* is called a **language** over Σ.

1.2 Deterministic Finite State Automata

Definition 1.4. A **deterministic finite state automaton** (DFA) is a tuple

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

where each component is as follows.

- Q is a finite set of **states**.
- Σ is a finite set of input symbols.
- $\delta : Q \times \Sigma \to Q$ is a function, called the **transition function**.
- $q_0 \in Q$ is called the **start state**.
- $F \subseteq Q$ is called the **accepting states**.

Definition 1.5. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

(a) For each symbol $a \in \Sigma$, we define $\delta_a : Q \to Q$ to be the function such that $\delta_a(p) = \delta(p, a)$ for any states $p, q \in Q$.

(b) For each string $w \in \Sigma^*$, we define $\delta_w : Q \to Q$ as follows.

  - δ_ϵ is the identity function.
  - For any strings $x \in \Sigma^*$ and any symbol $a \in \Sigma$, the function δ_{xa} satisfies $\delta_{xa}(p) = \delta_a(\delta_x(p))$ for any $p \in Q$.

Definition 1.6. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

- We say that A accepts a string $w \in \Sigma^*$ if $\delta_w(q_0) \in F$.
- The **language** of A, denoted $L(A)$, is defined as the set of strings that are accepted by A.

Definition 1.7. A language L is **regular** if there exists a DFA A such that $L(A) = L$.

1.3 Nondeterministic Finite State Automata

Definition 1.8. A **nondeterministic finite state automaton** (NFA) is a tuple

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

where each component is as follows.

- Q is a finite set of **states**.
- Σ is a finite set of input symbols.
- $\delta : Q \times \Sigma \times Q$ is a relation, called the **transition relation**.
- $q_0 \in Q$ is called the **start state**.
- $F \subseteq Q$ is called the **accepting states**.

Definition 1.9. Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

(a) For each symbol $a \in \Sigma$, we define $\delta_a \subseteq Q \times Q$ to be the relation such that $(p, q) \in \delta_a$ if and only if $(p, a, q) \in \delta$ for any states $p, q \in Q$.

(b) For each string $w \in \Sigma^*$, we define $\delta_w \subseteq Q \times Q$ as follows.

- δ_ϵ is the identity relation.
- For any strings $x \in \Sigma^*$, any symbol $a \in \Sigma$ and any states $p, q \in Q$,

$$(p, q) \in \delta_{xa}$$

if and only if there exists a state $r \in Q$ such that

$$(p, r) \in \delta_x \quad \text{and} \quad (r, q) \in \delta_a.$$

Definition 1.10. Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

- We say that A accepts a string $w \in \Sigma^*$ if there exists $q \in F$ such that $(q_0, q) \in \delta_w$.
- The **language** of A, denoted $L(A)$, is defined as the set of strings that are accepted by A.

Theorem 1.11. For every NFA A, there is a DFA A' with $L(A') = L(A)$.

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

- $\Delta : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$ is the function with

$$\Delta_a(P) = \bigcup_{p \in P} \{q \in Q : (p, q) \in \delta_a\}$$

for any $P \subseteq Q$ and $a \in \Sigma$.

- $\Phi = \{P \subseteq Q : P \cap F \neq \varnothing\}$.

Now we prove that for any $w \in \Sigma^*$, for any $q \in Q$ and for any $P \subseteq Q$, we have $q \in \Delta_w(P)$ if and only if $(p, q) \in \delta_w$ for some $p \in P$. For the induction basis, let $w = \epsilon$, and we have

$$q \in \Delta_\epsilon(P) \quad \Leftrightarrow \quad q \in P \quad \Leftrightarrow \quad (p, q) \in \delta_\epsilon \text{ for some } p \in P.$$

For the induction step, let $w = xa$, where a is the last symbol of w. Note that by the construction of Δ, we have $q \in \Delta_a(P)$ if and only if $(p, q) \in \delta_a$ for some $p \in P$. Thus, we can conclude that

$$\begin{aligned}
q \in \Delta_{xa}(P) \quad & \Leftrightarrow \quad q \in \Delta_a(\Delta_x(P)) \\
& \Leftrightarrow \quad (r, q) \in \delta_a \text{ for some } r \in \Delta_x(P) \\
& \quad \text{and } (p, r) \in \delta_x \text{ for some } p \in P \\
& \Leftrightarrow \quad (p, q) \in \delta_{xa} \text{ for some } p \in P.
\end{aligned}$$

Finally we prove that $L(A') = L(A)$, which is given by

$$\begin{aligned}
w \in L(A') \quad & \Leftrightarrow \quad \Delta_w(\{q_0\}) \in \Phi \\
& \Leftrightarrow \quad \Delta_w(\{q_0\}) \cap F \neq \varnothing \\
& \Leftrightarrow \quad q \in \Delta_w(\{q_0\}) \text{ for some } q \in F \\
& \Leftrightarrow \quad (p, q) \in \delta_w \text{ for some } q \in F \text{ and } p \in \{q_0\} \\
& \Leftrightarrow \quad (q_0, q) \in \delta_w \text{ for some } q \in F \\
& \Leftrightarrow \quad w \in L(A).
\end{aligned}$$

□

1.4 Regular Expressions

Definition 1.12. Let Σ be an alphabet. A **regular expression** over Σ is a string in the minimal language over $\Sigma \cup \{\varnothing, \epsilon, {}^*, \cup, (,)\}$ 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 $(e_1 e_2)$.
5. If e_1 and e_2 are regular expressions, then so is $(e_1 + e_2)$.
6. If e is a regular expression, then so is $(e)^*$.

Definition 1.13. 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. $L((e_1 e_2)) = L(e_1)L(e_2)$ for each regular expressions e_1 and e_2.
5. $L((e_1 + e_2)) = L(e_1) \cup L(e_2)$ for each regular expressions e_1 and e_2.
6. $L((e)^*) = L(e^*)$ for each regular expression e.

Remark. We may omit parentheses if there is no ambiguity.

Lemma 1.14. If L is a regular language over an alphabet Σ, then there is a regular expression e over Σ such that $L(e) = L$.

Proof. Since L is regular, there exists a DFA $A = (Q, \Sigma, \delta, q_0, F)$ with $L(A) = L$. Suppose that $Q = \{p_1, p_2, \ldots, p_n\}$ with $p_1 = q_0$. For any $1 \le i \le n$, for any $1 \le j \le n$ and for any $0 \le k \le n$, let $L_{ij}^{(k)}$ be the language of strings w satisfying the following conditions (a) and (b).

(a) $\delta_w(p_i) = p_j$.

(b) For any nonempty prefix x of w, $\delta_x(p_i) = p_\ell$ for some $\ell \le k$. (A nonempty proper prefix x of w is a string x such that $w = xy$ with $x \ne \epsilon$ and $y \ne \epsilon$.)

We are going to prove that for all $1 \le i \le n$, $1 \le j \le n$ and $0 \le k \le n$, there exists a regular expression $e_{ij}^{(k)}$ such that

$$L\big(e_{ij}^{(k)}\big) = L_{ij}^{(k)}.$$

The proof is by induction on k. For the induction basis, let $k = 0$. Let $\Pi_{ij} \subseteq \Sigma$ be the collection of symbols a with $\delta_a(p_i) = p_j$. If $i \ne j$, we have

$$L_{ij}^{(0)} = \bigcup_{a \in \Pi_{ij}} \{a\},$$

and thus we can construct $e_{ij}^{(0)}$ by

$$e_{ij}^{(0)} = \sum_{a \in \Pi_{ij}} a.$$

(If $\Pi_{ij} = \varnothing$, then the summation is defined as $\varnothing$.) If $i = j$, we have

$$L_{ii}^{(0)} = \{\epsilon\} \cup \bigcup_{a \in \Pi_{ii}} \{a\},$$

and thus we can construct $e_{ii}^{(0)}$ by

$$e_{ii}^{(0)} = \epsilon + \sum_{a \in \Pi_{ii}} a.$$

Now for the induction step, let $k \geq 1$. Suppose that $w \in L_{ij}^{(k)}$.

- If there is no nonempty proper prefix x of w such that $\delta_x(p_i) = p_k$, then we have $w \in L_{ij}^{(k-1)}$.
- Otherwise, let $x_0, x_1, \ldots, x_\ell$ be all nonempty proper prefixes of w such that

  $$\delta_{x_0}(p_i) = \delta_{x_1}(p_i) = \cdots = \delta_{x_\ell}(p_i) = p_k,$$

  where x_{h-1} is a proper prefix of x_h for $1 \leq h \leq \ell$. Then there exist $u_0, u_1, \ldots, u_{\ell+1}$ such that $w = u_0 u_1 \cdots u_{\ell+1}$, $x_0 = u_0$, and $x_h = x_{h-1} u_h$ for $1 \leq h \leq \ell$. Note that we have $u_0 \in L_{ik}^{(k-1)}$, $u_{\ell+1} \in L_{kj}^{(k-1)}$, and $u_h \in L_{kk}^{(k-1)}$ for $1 \leq h \leq \ell$. Thus, we can conclude that

  $$w \in L_{ik}^{(k-1)} \left(L_{kk}^{(k-1)} \right)^* L_{kj}^{(k-1)}.$$

As a result, we have

$$L_{ij}^{(k)} \subseteq L_{ij}^{(k-1)} \cup L_{ik}^{(k-1)} \left(L_{kk}^{(k-1)} \right)^* L_{kj}^{(k-1)},$$

implying

$$L_{ij}^{(k)} = L_{ij}^{(k-1)} \cup L_{ik}^{(k-1)} \left(L_{kk}^{(k-1)} \right)^* L_{kj}^{(k-1)}.$$

Therefore, we can construct $e_{ij}^{(k)}$ by

$$e_{ij}^{(k)} = e_{ij}^{(k-1)} + e_{ik}^{(k-1)} \left(e_{kk}^{(k-1)} \right)^* e_{kj}^{(k-1)}.$$

Now we construct the regular expression e with $L(e) = L$. Let Φ be the set of integers $j \in \{1, \ldots, n\}$ such that $p_j \in F$. Note that we have

$$L = \bigcup_{j \in \Phi} L_{1j}^{(n)},$$

and thus e can be constructed by

$$e = \sum_{j \in \Phi} e_{1j}^{(n)},$$

which completes the proof. $\square$