Chap 4

Properties of Regular Languages

4.1 Closure Properties of Regular Languages

(1) Closure under Simple Set Operations

Theorem 4.1 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, $\overline{L_1}$, and L_1^*. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof: If L_1 and L_2 are regular, then there exist regular expressions r_1 and r_2 such that $L_1 = L(r_1)$ and $L_2 = L(r_2)$. By definition, $r_1 + r_2$, r_1r_2, and r_1^* are regular expressions denoting the languages $L_1 \cup L_2$, L_1L_2, and L_1^*, respectively. Thus, closure under union, concatenation, and star-closure is immediate.

To show closure under complementation, let $M = (Q, \Sigma, \delta, q_0, F)$ be dfa that accepts L_1. Then the dfa

$$\widehat{M} = (Q, \Sigma, \delta, q_0, Q - F)$$

accepts $\overline{L_1}$.

Demonstrating closure under intersection takes a little more work. Let $L_1 = L(M_1)$ and $L_2 = L(M_2)$, where $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ and $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$ are dfa's. We construct from M_1 and M_2 a combined automaton $\widehat{M} = \left(\widehat{Q}, \Sigma, \widehat{\delta}, (q_0, p_0), \widehat{F}\right)$, whose state set $\widehat{Q} = Q \times P$ consists of pairs (q_i, p_j), and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state (q_i, p_j) whenever M_1 is in state q_i and M_2 is in state p_j. This is achieved by taking

$$\widehat{\delta}((q_i, p_j), a) = (q_k, p_l),$$

whenever

$$\delta_1(q_i, a) = q_k$$

and

$$\delta_2(p_j, a) = p_l.$$

$\widehat{F}$ is defined as the set of all (q_i, p_j), such that $q_i \in F_1$ and $p_j \in F_2$. Then it is a simple matter to show that $w \in L_1 \cap L_2$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_1 \cap L_2$ is regular. ■

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this book; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, Equation (1.3), taking the complement of both sides. Then

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

for any languages L_1 and L_2. Now, if L_1 and L_2 are regular, then by closure under complementation, so are $\overline{L_1}$ and $\overline{L_2}$. Using closure under union, we next get that $\overline{L_1} \cup \overline{L_2}$ is regular. Using closure under complementation once more, we see that

$$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$$

is regular.

The following example is a variation on the same idea.

Example 4.1 Show that the family of regular languages is closed under difference. In other words, we want to show that if L_1 and L_2 are regular, then $L_1 - L_2$ is necessarily regular also.

The needed set identity is immediately obvious from the definition of a set difference, namely

$$L_1 - L_2 = L_1 \cap \overline{L_2}.$$

The fact that L_2 is regular implies that $\overline{L_2}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_1 \cap \overline{L_2}$ is regular, and the argument is complete. ■

Theorem 4.2 The family of regular languages is closed under reversal.

Proof: The proof of this theorem was suggested as an exercise in Section 2.3. Here are the details. Suppose that L is a regular language. We then construct an nfa with a single final state for it. By Exercise 7, Section 2.3, this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts w^R if and only if the original nfa accepts w. Therefore, the modified nfa accepts L^R, proving closure under reversal. ■

[2] Closure under Other Operations

Definition 4.1

Suppose Σ and Γ are alphabets. Then a function

$$h : \Sigma \to \Gamma^*$$

is called a **homomorphism**. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function h is extended to strings in an obvious fashion; if

$$w = a_1 a_2 \cdots a_n,$$

then

$$h(w) = h(a_1)\, h(a_2) \cdots h(a_n).$$

If L is a language on Σ, then its **homomorphic image** is defined as

$$h(L) = \{h(w) : w \in L\}.$$

Example 4.2 Let $\Sigma = \{a, b\}$ and $\Gamma = \{a, b, c\}$ and define h by

$$\begin{aligned} h(a) &= ab, \\ h(b) &= bbc. \end{aligned}$$

Then $h(aba) = abbbcab$. The homomorphic image of $L = \{aa, aba\}$ is the language $h(L) = \{abab, abbbcab\}$.

If we have a regular expression r for a language L, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each Σ symbol of r.

Example 4.3 Take $\Sigma = \{a, b\}$ and $\Gamma = \{b, c, d\}$. Define h by

$$h(a) = dbcc,$$
$$h(b) = bdc.$$

If L is the regular language denoted by

$$r = (a + b^*)(aa)^*,$$

then

$$r_1 = (dbcc + (bdc)^*)(dbccdbcc)^*$$

denotes the regular language $h(L)$. ■

Theorem 4.3 Let h be a homomorphism. If L is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

(Skip from here!)