Lecture 4: Properties of Regular Language

Instructor: Ketan Mulmuley
Scriber: Yuan Li

January 15, 2015

1 Properties of Regular Language

Recall in the last lecture, we have shown that if L_1, L_2 are regular languages, then $L_1 \cap L_2$ is also regular, using the following identity

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

Let us give an other proof here. Let $M_1 = (Q, \Sigma, \delta_1, q_0, F), M_2 = (P, \Sigma, \delta_2, p_0, F')$ be DFAs accepting L_1, L_2 respectively. Define a new DFA

$$M = (Q \times P, \Sigma, \delta, q_0 \times p_0, F \times F'),$$

where

$$\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a)).$$

It is easy to see $L(M) = L_1 \cap L_2$.

Given $L_1, L_2 \subseteq \Sigma^*$, define

$$L_1/L_2 = \{x \in \Sigma^* : \exists y \in L_2 \text{ such that } xy \in L_1\}.$$

Proposition 1.1. *If $R \subseteq \Sigma^*$ is regular and L is any set (probably non-regular), then R/L is regular.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA such that $L(M) = R$. Define a new DFA $M' = (Q, \Sigma, \delta, q_0, F')$, where $q' \in F'$ if and only if there exists $y \in L$ such that $\hat{\delta}(q, y) \in F$. Then $L(M') = R/L$. □

Let Δ an alphabet set. A *substitution* $f : \Sigma \to \mathcal{P}(\Delta)$ such that for all $a \in \Sigma$, $f(a) \subseteq \Delta^*$ is a regular language. Note that, if $x \in \Sigma^*$, then $f(\epsilon) = \epsilon$ and $f(xa) = f(x)f(a)$ for any $x \in \Sigma^*$ and $a \in \Sigma$. Let

$$f(L) = \bigcup_{x \in L} f(x) \subseteq \Delta^*.$$

Theorem 1.2. *If L is regular, then $f(L)$ is regular for any substitution f.*

Proof. Since L is regular, it has a regular expression R, which is a string in alphabet Σ and $*, +, (,)$.

Since $f(a)$ is regular for all $a \in \Sigma$, there is a regular expression $S(a)$ (a string in alphabet and $*, +, (,)$ for each $a \in \Sigma$.

Let T be the regular expression obtained by replacing each a in R by $S(a)$. Then $f(L) = L(T)$. □

Definition 1.3. *A string homomorphism is a function on strings that works by substituting a particular string for each symbol, that is, $g : \Sigma \to \Delta^*$.*

Corollary 1.4. *If L is regular then $f(L)$ is regular for any homomorphism f.*

Let $h : \Sigma \to \Delta^*$ be a homomorphism, where $L \subseteq \Delta^*$. Let

$$h^{-1}(L) = \{x \in \Sigma : h(x) \in L\}.$$

Proposition 1.5. *If L is regular then $h^{-1}(L)$ is regular for any homomorphism h.*

Proof. Let M be the DFA accepting L, that is, $L = L(M)$, where $M = (Q, \Delta, \delta, q_0, F)$. Let $M' = (Q, \Sigma, \delta', q_0, F)$, where $\delta'(q, x) = \hat{\delta}(q, h(x))$ for all $x \in \Sigma$, and $\hat{\delta}$ is the transitive closure of δ. □

In the last class, we have proved $L' = \{0^n 1^n : n \geq 1\}$ is not regular.

Proposition 1.6. $L = \{a^n b a^n : n \geq 1\}$ *is not regular.*

Proof. Define homomorphism $h_1 : \{a, b, c\} \to \{a, b\}^*$ as

$$\begin{aligned} h_1(a) &= a, \\ h_1(b) &= ba, \\ h_1(c) &= a, \end{aligned}$$

and homomorphism $h_2 : \{a, b, c\} \to \{0, 1\}^*$ as

$$\begin{aligned} h_2(a) &= 0, \\ h_2(b) &= 1, \\ h_2(c) &= 1. \end{aligned}$$

We claim $h_2(h_1^{-1}(L) \cap a^*bc^*) = L'$ (observe $h_1^{-1}(L) \cap a^*bc^* = \{a^n bc^{n-1}\}$), which should not be regular because L' is not regular as we proved in the last lecture. □

2 Some Decision Problems

Consider the following three problems:

- Given a DFA M, is $L(M)$ empty?
- Given a DFA M, is $L(M)$ finite?
- Given two DFAs M_1 and M_2, is $L(M_1) = L(M_2)$?

The first problem is relatively easy. Perform a Breath First Search (BFS) on the (state transition) graph of M, and find which states are reachable from q_0. $L(M)$ is nonempty if and only if some final state is reachable from q_0. Note that the running time of BFS is linear in the size of graph, and therefore testing whether $L(M)$ is empty can be done in linear time.

For the second problem, first decide if $L(M)$ is nonempty; otherwise, stop. Then decide if there is a loop in the graph of M reachable from q_0 and going through some final state.

For the third problem, construct a DFA M to accept

$$L = (L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1}).$$

$L(M_1) = L(M_2)$ if and only if $L(M) = \emptyset$, which reduces this problem to the first one. For the running time, recall the Cartesian product construction. Let $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ and $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$, and $M = (Q \times P, \Sigma, \delta, q_0 \times p_0, F')$, where

$$\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$$

and $F' = \{(p, q) : p \in F_1 \text{ or } q \in F_2\}$. Then, $L(M) = L(M_1) \cup L(M_2)$. Similar construction works for $L(M_1) \cap L(M_2)$. Apply this, the running time would be $O(n^4)$, where n is the input size.

On one side of the building (Eckhart), we don't care about time; on the other side of the building (Ryerson), we need to do things fast. We are at the transition (Ryerson 251), we'd better worry about it.

3 Myhill - Nerode Theorem

Given a regular language L, what is the smallest DFA accepting L? This problem looks very difficult. However, it can be solved efficiently.

Definition 3.1. *An equivalence relation on $\Sigma^* \times \Sigma^*$ is called* right-invariant *if*

$$x \sim y \Rightarrow xw \sim yw \;\; \forall w \in \Sigma^*.$$

Theorem 3.2. *The following statements are equivalent:*

(1) L is accepted by some DFA (i.e., L is regular).

(2) L is the union of equivalence classes of a right-invariant equivalence relation of finite index, that is, L has finite number of equivalence classes.

(3) Define $\sim_L$ by $x \sim_L y$ if and only if for all $z \in \Sigma^*$ $xz \in L$ if and only if $yz \in L$. Then $\sim_L$ has finite index.

In fact, (3) implies the smallest DFA accepting L, where each equivalence class corresponds to each state. Please refer to the textbook for complete proof.