COSE215: Theory of Computation

Lecture 8 — Properties of Regular Languages (1)

Hakjoo Oh
2017 Spring

Properties of Regular Languages

- Closure properties
- "Pumping Lemma" for regular languages

Closure Properties

If one (or several) languages are regular, then certain related languages are also regualr. E.g.,

- Given regular languages L_1 and L_2, $L_1 \cup L_2$ is also regular.
- Given regular languages L_1 and L_2, $L_1 \cap L_2$ is also regular.

The family of regular languages is *closed* under union and intersection.

Closure Properties

Regular languages are closed under:

- union
- difference
- complementation
- intersection
- reversal
- homomorphism
- ...

Closure under Union

Theorem

If L and M are regular languages, then so is $L \cup M$.

Closure under Difference

Theorem

If L and M are regular languages, then so is $L - M$.

Closure under Complementation

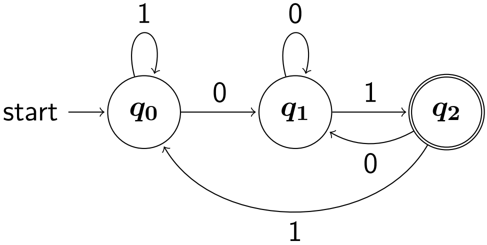
Theorem

If L is a regular language over alphabet Σ, then $\overline{L} = \Sigma^* - L$ is also a regular language.

Let A be a DFA that accepts L, i.e., $L = L(A)$ for DFA $A = (Q, \Sigma, \delta, q_0, F)$. Define a DFA B as follows:

$$B = (Q, \Sigma, \delta, q_0, Q - F)$$

ex)



Closure under Intersection

Theorem

If L and M are regular languages, then so is $L \cap M$.

- Non-constructive proof: $L \cap M = \overline{\overline{L} \cup \overline{M}}$
- Constructive proof: construct an automaton that accepts $L \cap M$.

Closure under Intersection

Theorem

If L and M are regular languages, then so is $L \cap M$.

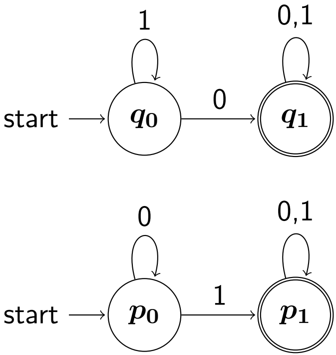
- Non-constructive proof: $L \cap M = \overline{\overline{L} \cup \overline{M}}$
- Constructive proof: construct an automaton that accepts $L \cap M$.

(Constructive proof) Let $A_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ and $A_2 = (P, \Sigma, \delta_2, p_0, F_2)$ be DFAs for L and M, respectively. Define the automaton A:

$$A = (Q \times P, \Sigma, \delta, (q_0, p_0), F_1 \times F_2)$$

where $\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$. Then, $L(A) = L(A_1) \cap L(A_2)$.

Example



Closure under Reversal

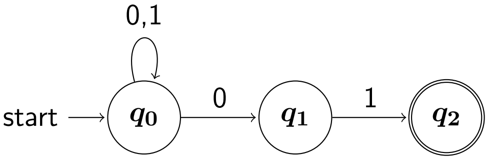
Theorem

If L is a regular language, then so is L^R.

Let A be a ϵ-NFA that accepts L, then we can construct an automaton that accepts L^R as follows:

1. Reverse all the arcs in the transition graph for A.
2. Make the start state of A be the only accepting state for the new automaton.
3. Create a new start state p_0 with transitions on ϵ to all the accepting states of A.

Example



Closure under Homomorphism

Definition (Homomorphism)

Suppose Σ and Γ are alphabets. Then a function

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

is called a homomorphism. For a given string $w = a_1 a_2 \cdots a_n$,

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

For a language L,

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

Theorem

If L is a regular language over Σ and h is a homomorphism on Σ, then $h(L)$ is also regular.

Example

Let $\Sigma = \{0, 1\}$ and $\Gamma = \{a, b\}$ and define h by

$$h(0) = ab, \qquad h(1) = \epsilon$$

Given any string of 0's and 1's, it replaces all 0's by the string ab and replaces all 1's by the empty string. For example,

$$h(0011) = abab.$$

If L is a language of regular expression 10^*1, i.e., any number of 0's surrounded by 1's. Then $h(L)$ is the language $(ab)^*$.