Lecture 3 - Regular Languages, Operations, Expressions

Formal Definitions

A language over alphabet $\Sigma$ is any subset of $\Sigma^*$.

Deterministic Finite Automata A finite automaton is $5$-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. $Q$ is a finite set called the states.
2. $\Sigma$ is a finite set called the alphabet.
3. $\delta : Q \times \Sigma \rightarrow Q$ is the transition function.
4. $q_0$ is the start state.
5. $F \subseteq Q$ is the set of accept states.

Give the formal definition of some of these machines, $M_1 \cdots M_8$.

Formal Definition of Computation

A finite automaton $M = (Q, \Sigma, \delta, q_0, F)$, accepts a string $w \in \Sigma^*$, $w = w_1w_2 \ldots w_n$ if there is a sequence of states $r_0, r_1, \ldots, r_n$ in $Q$ such that

1. $r_0 = q_0$.
2. $\delta(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, \ldots, n - 1$, and
3. $r_n \in F$.

A language is a regular language if some finite automata recognizes it.
Try to design DFAs that recognize the languages described below.

alphabet $\Sigma = \{m, n, o\}$, $L_1 = \{mom\}$

alphabet $\Sigma = \{0, 1\}$ $L_2 = \{s \in \Sigma^* \mid 000$ is a substring of $s\}$

alphabet $\Sigma = \{a, b\}$ $L_3 = \{w \in \Sigma^* \mid w$ does not end with $bb\}$

alphabet $\Sigma = \{a, b\}$ $L_4 = \{s \in \Sigma^* \mid |s| > 4\}$

alphabet $\Sigma = \{0, 1\}$ $L_5 = \{s \in \Sigma^* \mid 01101$ is a substring of $s\}$

alphabet $\Sigma = \{0, 1\}$ $L_6 = \{s \in \Sigma^* \mid 01101$ is NOT a substring of $s\}$

**Regular Operations**

Let $A$ and $B$ be languages.

- **Union** $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- **Concatenation** $A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$
- **Star** $A^* = \{x_1x_2\ldots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$

What about the alphabets?

**Examples**

$A = \{\text{red, green, blue}\} \ B = \{\text{cat, dog}\}$

- **Union** $A \cup B = \{\text{red, green, blue, cat, dog}\}$
- **Concatenation** $A \circ B = \{\text{redcat, reddog, greencat, greendog, bluecat, bluedog}\}$
- **Star** $A^* = \{\varepsilon, \text{ red, green, blue, redred, redgreen, } \ldots \text{, greenredredbluegreen, } \ldots\}$

**Theorem:** The class of regular languages is closed under the union operation.

**Proof:** Let $L_1$ and $L_2$ be regular languages, and let

$M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize $L_1.$

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize $L_2.$

Construct $M = (Q, \Sigma, \delta, q_0, F)$ to recognize $L_1 \cup L_2.$
1. \( Q = \{ (r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2 \} \)

2. \( \Sigma \) is the same or \( \Sigma = \Sigma_1 \cup \Sigma_2 \) and change proof.

3. \( \delta((r_1, r_2), a) = (\delta(r_1, a), \delta(r_2, a)) \).

4. \( q_0 = (q_1, q_2) \)

5. \( F = \{ (r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2 \} \)

Example: \( M_1 \cup M_6 \) but change the state names first.

**Theorem:** The class of regular languages is closed under the intersection operation.