The Relationship between languages

$$All\ Languages(L \subseteq \Sigma^*)$$
$$(eg.\, Halting\ problem)$$
$$\supseteq$$
$$Decidable/Computable\ Languages$$
$$\supseteq$$
$$NP(All\ problems\ that\ are\ efficiently\ verifiable)$$
$$(eg.\, Ham.\ Cycle)$$
$$\supseteq$$
$$P(All\ problems\ that\ are\ efficiently\ solvable, i.e.\ solvable\ in\ n^{O(1)}\ time)$$
$$(eg.\, Max\ Flow)$$
$$\supseteq$$
$$Regular\ Languages$$

- **Regular Languages**: Problems that are solvable without memory, i.e. problems that are solvable by **finite automatons**.
- **Upper bound**: Given L, prove L is decidable in time $T(n)$.
- **Lower bound**: Given L, prove L is not decidable in time $T(n)$.

Finite Automaton

Def 2.1 **A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where**

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

Def 2.2 **Let $M = (Q, \Sigma, \delta, q_0, F)$ be a finite automaton. Let $w = w_1 w_2 ... w_n$ be a string, where each $w_i \in \Sigma$. Then M accepts w if there is a sequence of states $r_0, r_1, \dots, r_n \in Q$, s.t.**

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

Def 2.3 **If L is the set of strings that M accepts, we say L is the language of M, and write $L(M) = L$. We say M recognizes/decides/accepts L.**

- If M accepts no strings, it recognizes one language, namely, the empty language.

Def 2.4 **$L \subseteq \Sigma^*$ is a regular language if there is a finite automaton that accepts L. Let $A, B \subseteq \Sigma^*$. Define:**

- **(Union)** $A \cup B = \{x \in \Sigma^* | x \in A \ or \ x \in B\}$
- **(Concatenation)** $AB = \{xy | x \in A, y \in B\}$
- **(Star)** $A^* = \{x_1 x_2 \ldots x_k | k \geq 0, x_1, \ldots, x_k \in A\}$

eg.

If

$$\Sigma = \{0, 1\}, A = \{\epsilon, 0, 00, \ldots\}, B = \{\epsilon, 1, 11, \ldots\}$$

Then

$$\begin{gathered} AB = \{0^i 1^j | i, j \geq 0\}, \\ A^* = A, \\ B^* = B, \\ (AB)^* = \Sigma^* \end{gathered}$$

Thm 2.5 **If A_1, A_2 are regular languages, so is $A_1 \cup A_2$.**

Proof: Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ accepts A_1, and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ accepts A_2. Construct M to accept $A_1 \cup A_2$, where $M = (Q, \Sigma, \delta, q, F)$:

1. $Q = Q_1 \times Q_2 = \{(r_1, r_2) | r_1 \in Q_1, r_2 \in Q_2\}$
2. $\delta : Q \times \Sigma \to Q$ is defined as for each $(r_1, r_2) \in Q$, and each $a \in \Sigma$, let $\delta((r_1, r_2)) = (\delta_1(r_1), \delta_2(r_2))$
3. $q_0 = (q_1, q_2)$
4. $F = \{(r_1, r_2) | r_1 \in F_1 \ or \ r_2 \in F_2\}$

Thm 2.6 **If A_1, A_2 are regular languages, so is $A_1 A_2$.**

- **DFA: Deterministic Finite Automaton**
- **NFA: Nondeterministic Finite Automaton**

Def 2.7 **An NFA is a 5-tuple$(Q, \Sigma, \delta, q_0, F)$, where**

1. Q is a finite set of states.
2. Σ is the alphabet.
3. $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to P(Q)$ is the transition function.
4. $q_0 \in Q$ is the start state.
5. $F \subseteq Q$ is the set of accept states.

Def 2.8 **Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA, and let $w \in \Sigma^*$. Say N accepts w if we can write $w = y_1 y_2 \ldots y_n$, where $y_i \in \Sigma \cup \{\epsilon\}$, and there exist $r_0, r_1, \ldots, r_m \in Q$, s.t.**

1. $r_0 = q_0$
2. $r_{i+1} \in \delta(r_i, y_{i+1}) \ for \ i = 0, 1, \ldots, m-1$
3. $r_m \in F$

Thm 2.9 **Every NFA has an equivalent DFA.**