1 Variants of NFAs

1.1 Multiple Start States

Let $M = (Q, \Sigma, \delta, I, F)$, where $I \subseteq Q$ and $I \neq \emptyset$.

Theorem 1.1

If $L = \mathfrak{L}(M)$, then there exists an NFA N such that $\mathfrak{L}(N) = L$.

Proof.

Suppose $M = (Q, \Sigma, \delta, I, F)$.

We construct $M' = (Q \cup \{q_0\}, \Sigma, \delta', q_0, F')$, where $q_0 \notin Q$, and for each $a \in \Sigma$ we have $\delta'(q_0, a) = \bigcup\{\delta(q, a) \mid q \in I\}$ and $\delta'(q, a) = \delta(q, a)$, and

$$F' = \begin{cases} F, \text{ if } I \cap F = \emptyset \\ F \cup \{q_0\}, \text{ if } I \cap F \neq \emptyset. \end{cases}$$

We prove then that $\mathfrak{L}(M) = \mathfrak{L}(M')$. □

1.2 NFA with λ-Transitions

Let $M = (Q, \Sigma, \delta, q_0, F)$ and $\delta : Q \times (\Sigma \cup \lambda \to 2^Q)$.

Let $\mathfrak{L}(M) = \{x \in \Sigma^* \mid$ there is a path from q_0 to a final state such that x is a concatenation of the labels o

If $L = \mathfrak{L}(M)$, then there exists an NFA N such that $L = L(N)$.

For exery state $q \in Q$ let $E(q) = \{q' \mid$ there is a path from q to q' labelled by $\lambda\}$.

In particular, $q \in E(q)$.

Let $M' = (Q, \Sigma, \delta', E(q_0), F)$ and $\delta'(q, a) = \bigcup\{E(q') \mid q' \in \delta(q, a)\}$.

2 Closure Results

Suppose that $L_1, L_2 \in \Sigma^*$ are accepted by finite state automata.

Then $L_1 \cup L_2$, $L_1 \cdot L_2$, $\Sigma^* - L_1$, L_1^*, L_1^+, $L_1 \cap L_2$ can also be accepted by finite automata.

We prove now that this statement holds for any languages L_1 and L_2.

Let $M_1 = (Q, \Sigma, \delta, q_1, F_1)$ and $M_2 = (Q, \Sigma, \delta, q_2, F_2)$, with $\mathfrak{L}(M_1) = L_1$ and $\mathfrak{L}(M_2) = L_2$.

Assume $Q_1 \cap Q_2 = \emptyset$.

Let $M' = (Q, \Sigma, \delta_1, q, Q_1 - F_1)$ (an NFA with interchanged accept and reject states).

Then $\mathfrak{L}(M') = \Sigma^* - L_1$.

Proof.

Let $x \in \Sigma^*$. Then $x \in \mathfrak{L}(M')$ if and only if $\delta_1^*(q_1, x) \in Q - F_1$, which holds if and only if $\delta_1^*(q, x) \notin F_1$, which is equivalent to $x \notin \mathfrak{L}(M) = L_1$, and thus $x \in \Sigma^* - L_1$.

If M_1 is an NFA, then it is possible that $\delta_1^*(q_1, x) \cap F \neq \emptyset$ and $\delta_1^*(q_1, x) \cap (Q_1 - F) \neq \emptyset$.

Union

Let $M' = (Q_1 \cup Q_2, \Sigma, \delta, \{q_1, q_2\}, F_1 \cup F_2)$ and $\delta : (Q_1 \cup Q_2) \times \Sigma \to Q \cup Q_2$.

Then $\delta(q, a) = \begin{cases} \delta_1(q, a) \text{ if } q \in Q_1 \\ \delta_2(q, a) \text{ if } q \in Q_2 \end{cases}$

Then $\mathfrak{L}(M') = \mathfrak{L}(M_1) \cup \mathfrak{L}(M_2)$.

Intersection

Let $M' = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2), F_1 \times F_2)$, thus creating an NFA consisting of two subNFA running in parallel.

Then $\delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta(q_2, a))$ for all $a \in \Sigma$ and $q_1 \in Q_1$ and $q_2 \in Q_2$.

Concatenation Connect two NFAs by λ-transitions from the accepting states in one connected to the initial state in another.

L_1^+ Connect the accepting states to the initial states with λ-transitions.

L_1^* If $\lambda \in L_1$, then $L_1^* = L_1^+$ if $\lambda \notin L_1$.

Connect the accepting states by λ-transitions to the old initial state, adding a new accepting initial state connected to the old initial state by the λ-transition.

□

Corollary 2.1

Every regular language can be accepted by a finite state automaton.

Proof.

The proof proceeds by structural induction. □