Deterministic Finite Automata - examples

1. $L = \{w \in \{0,1,2\}^* \mid w$ is a ternary representation of a multiple of $5\}$.

   Assume that the empty string is in the language. That is, $D(\epsilon) = 0$.

   - For $w \in \{0,1,2\}^*$, let $D(w)$ denote its decimal value.
   - Let $w \in \{0,1,2\}^*$ and $D(w) \bmod 5 = i$. Then, $D(w) = 5 \times q + i$ for all natural numbers q and $i \in \{0,1,2,3,4\}$.
     - What about $D(wj)$ for $j \in \{0,1,2\}$?
       * $D(wj) = 3D(w) + j = 3 \times (5 \times q + i) + j$.
       * $D(wj) \bmod 5 = (3 \times i + j) \bmod 5$.
   - The DFA M will have 5 states q_i for $0 \leq i \leq 4$ such that:
     - q_0 is the start state,
     - $\hat{\delta}(q_0, w) = q_{D(w) \bmod 5}$, and
     - q_0 is the only accepting state.
   - To accomplish this, define the transition function as follows:
     - $\delta(q_i, j) = q_{(3i + j) \bmod 5}$.
   - CLAIM: $\hat{\delta}(q_0, w) = q_{D(w) \bmod 5}$
     - By induction on $|w|$.
     - $|w| = 0$: True.
     - $|w| \geq 1$: Let $w = uj$ for $j \in \{0,1,2\}$ and $u \in \{0,1,2\}^*$.

$$\begin{aligned}
\hat{\delta}(q_0, uj) &= \delta(\hat{\delta}(q_0, u), j) \\
&= \delta(q_{D(u) \bmod 5}, j) \\
&= q_{(3D(u) + j) \bmod 5} \\
&= q_{D(uj) \bmod 5}
\end{aligned}$$

2. Let $L_3 = \{w \in \{0,1\}^* \mid \text{the third symbol from the right is } 1\}$. Design a DFA for L_3.

Solution: Store the last 3 bits seen in the input in the state. That is, the DFA has 2^3 states labeled with all possible bit strings of length 3.

From a state labeled $x_1x_2x_3$ (where $x_i \in \{0,1\}$) there is a transition to state labeled x_2x_30 on input 0 and to state labeled x_2x_31 on input 1.

The final states are those that are labeled $1x_2x_3$ for $x_i \in \{0,1\}$.

3. Let A and B be two regular languages over a finite alphabet Σ.

   Define the language $perfect - shuffle(A, B)$ as follows:

   $\{w | \ w = a_1 b_1 \cdots a_k b_k, \text{ where each } a_i, b_i \in \Sigma, a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B\}$.

   Show that $perfect - shuffle(A, B)$ is regular.

   Solution: Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ be a DFA that accepts A. Let $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ be a DFA that accepts B. *Construct* a DFA M that, on an input, alternatively runs M_1 and M_2 on one symbol: if the current move is using M_1 (M_2) then make the next move using M_2 (M_1, respectively).

   Define a DFA $M = (Q, \Sigma, \delta, q_0, F)$ as follows.

   - $Q = Q_1 \times Q_2 \times \{1, 2\}$.
   - For all $p \in Q_1$, $q \in Q_2$, and $a \in \Sigma$ define:
     - $\delta(\langle p, q, 1\rangle, a) = \langle \delta_1(p, a), q, 2\rangle$.
     - $\delta(\langle p, q, 2\rangle, a) = \langle p, \delta_2(q, a), 1\rangle$.
   - $q_0 = \langle q_1, q_2, 1\rangle$.
   - $F = F_1 \times F_2 \times 1$.