Formal Definition of a Deterministic Finite Automata

A DFA is a tuple $(Q, \Sigma, \delta, q_0, F)$

1.) Q is the set of states

2.) Σ is the alphabet for Input

3.) Transition Function

$$\delta: \underset{\text{current state}}{Q} \times \underset{\text{Input}}{\Sigma} \longrightarrow \underset{\text{state to transition}}{Q}$$

4.) $q_0 \Sigma Q$ is the start state

5.) $F \subseteq Q$ is the set of Final or accept state

Formal Definition of Computation

A DFA M accepts input string

$W = W_1, W_2, \ldots, W_k$ (each $w_i \in \Sigma$

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 $i = 0, \ldots, n-1$

3. $r_n \in F$

Def:

A DFA M recognizes the language A if

$A = \{ w \mid m \text{ accepts } w \}$

Def: A language recognized by a DFA is called a regular language

Design a DFA that accept an even number of 1's

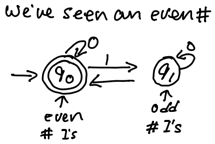
01001011010l

Consider the states:

Let there be 2 states

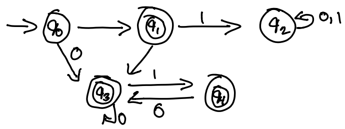
We've seen an odd # Of 1's or

we've seen an even # Of 1's

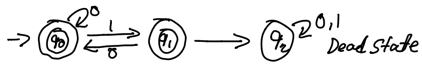


Design a DFA that never contains 11 as a substring

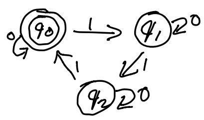
If current input is a 1, then another 1 transitions to a dead state.



or



Design a DFA that accepts string where the number of 1's is a multiple of 3



Design a DFA that accept $\{101\}^*$

$*$ = 0 or more copies of 101