Def.: A deterministic finite automaton (DFA) is a 5-tuple

\[ M = (Q, \Sigma, \delta, s, F) \]

where

- \( Q \) is a finite set of states.
- \( \Sigma \) is an alphabet — that is, a finite set of characters.
- \( s \) is a member of \( Q \), the start state (\( q_0 \) in text).
- \( F \) is a subset of \( Q \), the set of final states (accepting).
- \( \delta \): the transition function:

\[ \delta: Q \times \Sigma \rightarrow Q \]

\[ (q, c) \rightarrow q' \]

class DFA {
    set <state> Q;
    set <char> \Sigma;
    state s;
    set <state> F;
    state delta (state p, char c); }

KWR prefers:
set delta
set <Triple <state, char, state>> delta;

Visualisation:

\[ \delta \subseteq (Q \times \Sigma) \times Q \]

\[ p, q \in Q, c \in \Sigma \]

- \( Q \) is a set of nodes.
- \( S \) is a set of edges with labels from \( \Sigma \).

\[ P \xrightarrow{c} Q \]

\[ \text{Start} \quad \text{New state} \]

\[ x = cc \quad x' = c \]
Example: Tell whether a given string $X$ over $\Sigma = \{0, 1\}$ has an odd number of 1s. ($\#_1(X) = \text{number of 1's}$.) If $\#_1(X)$ is odd or not??

$Q = \{\text{even, odd}\}.$  

$s = \text{even, since we have seen zero of 1's.}$  

$S = \{(s, 0, s), (s, 1, \text{odd}), (\text{odd, 0, odd}), (\text{odd, 1, even})\}.$

The language $L(M)$ of this DFA $M$ equals $\{x \in \{0, 1\}^* : \#_1(x) \text{ is odd zero or more}\}$.

Def: A computation by a DFA $M = (Q, \Sigma, S, s, F)$ is a sequence

$C = (q_0, x_1, q_1, x_2, \ldots, x_{n-1}, q_{n-1}, x_n, q_n)$ where:

$n = |x| \text{ (the length of } x)\)$

$x = x_1 \ldots x_n \text{ where } x_i \text{ is } i^\text{th} \text{ bit:}$

$q_0 = s, \text{ each } q_i \in Q \text{ and}$

$C \text{ is accepting if also } q_n \in F.$

$\overline{\text{Def}}: L(M) = \{x \in \Sigma^* : M \text{ has an accepting computation on input } x\}.$
\( \Sigma = \{0, 1\} \)

\( M_0 = \)

\( Q = \{s\}, F = \emptyset \)

\( L(M_0) = \emptyset \)

\( \emptyset \) empty set.

\( \epsilon \) empty string

\( M_{\text{acc}} = \)

\( Q = \{s\}, \Sigma = \{0, 1\} \)

\( F = \{s\} \)

\( L(M_{\text{acc}}) = \Sigma^* = \{0, 1\}^* \)

other DFAs \( M \) st. \( L(M) = \Sigma^* \):

\( q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_0 \)

\( q = \{q_0, q_1, q_2\} \)

\( F = \{q_0, q_2\} \)

Hence a DFA need not be "in lowest terms".

One More Example:
Tell whether a given string \( X \) has the property that

\( \#0 \equiv \#1(X) \equiv 0 \mod 3 \) ?

\( Q = \{q_0, q_1, q_2, q_3\} \)

\( = 0 \equiv 1 \equiv 2 \)

\( S = q_0 \)

\( F = \{q_0, q_2\} \)

\( \Sigma = \{0, 1, 2\} \)

change \( \Sigma = \{0, 1, 2\} \) if the sum of digits in \( X \) is a multiple of

\( \equiv 0 \mod 3 \)

\( \equiv 1 \mod 3 \).
set operation:
union \[ A \cup B = \{ x : x \in A \text{ or } x \in B \} \]
intersection \[ A \cap B = \{ x : x \in A \text{ and } x \in B \} \]
Difference of sets \[ A \setminus B = \{ x : x \in A \text{ but } x \notin B \} \]
(in the Text, '−')
symmetric difference \[ A \Delta B = \{ x : x \in A \text{ XOR } x \in B \} \]
\[ (A \setminus B) \cup (B \setminus A) \]