Do not draw your sword to kill a fly.

-Korean Proverb
Finite State Automata
7-Tuple TM

\[ Q \] is the set of states \((q_0, q_1, q_2, \ldots q_n)\)

\( q_0 \in Q \) is the initial state

\( \Gamma \) is the tape alphabet \((_\in\Gamma, \Sigma \subseteq \Gamma)\)

\( \delta \) is the transition function \((Q \times \Gamma \to \Gamma \times \{L, R\} \times Q)\)

\( q_{\text{accept}} \in Q \) is the accept state(s)

\( q_{\text{reject}} \in Q \) is the reject state(s)

\( \Sigma \) is the input alphabet \((_\not\in\Sigma)\)
7-Tuple TM

\( Q \) is the set of states \( (q_0, q_1, q_2, \ldots, q_n) \)

\( q_0 \in Q \) is the initial state

\( \Gamma \) is the tape alphabet \( (\_ \in \Gamma, \Sigma \subseteq \Gamma) \)

\( \delta \) is the transition function \( (Q \times \Gamma \rightarrow \Gamma \times \{L, R\} \times Q) \)

\( q_{\text{accept}} \in Q \) is the accept state(s)

\( q_{\text{reject}} \in Q \) is the reject state(s)

\( \Sigma \) is the input alphabet \( (\_ \notin \Sigma) \)
δ is the transition function \((Q \times \Gamma \rightarrow \times \times \{ \times, \times \} \times Q)\)
\( \delta \) is the transition function \((Q \times \Gamma \rightarrow \{x, x, R\} \times Q)\)
TM that decides \( L = \{ w \mid w \text{ ends with 1} \} \)
TM that decides $L = \{ w \mid w \text{ ends with 1} \}$
FSA that decides $L = \{ w \mid w \text{ ends with } 1 \}$
Turing Machine

\[ w \]

\[ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \]
Finite State Automata
FSA that decides $L = \{ w \mid w \text{ ends with } 1 \}$
5-Tuple FSA

\( Q \) is the set of states \((q_0, q_1, q_2, ... q_n)\)

\( q_0 \in Q \) is the initial state

\( \Sigma \) is the input alphabet \((_ \in \Sigma)\)

\( \delta \) is the transition function \((Q \times \Sigma \rightarrow Q)\)

\( q_{\text{accept}} \in Q \) is the accept state(s)
FSA that decides $L = \{ w \mid w \text{ starts with } 1 \}$
L = \{ 0 \{1, 2\} \}
$$L = \{ 0 \{1, 2\} \}$$
\[ L = \{ 0 \{1, 2\} \} \]
\[ L = \{ 0 \{1, 2\} 2^n \mid n \geq 0 \} \]
L = \{ \{0, 1, 2\}^n \mid \text{where the digits sum to} \ \text{a product of 3} \ (n \geq 0) \}
\[ L = \{ \{0, 1, 2\}^n \mid \text{where the digits sum to a product of 3} \} \] (n \geq 0) \]
If a language can be decided by an FSA...

It is called *regular*. 
FSA that decides $L = \{ w \mid w \text{ ends with } 1 \}$
L = \{ \{0, 1, 2\}^n \mid \text{where the digits sum to } \}
\text{a product of 3 } (n \geq 0)
\[ L = \{ \{0, 1, 2\}^n \mid \text{where the digits sum to a product of 3} \ (n \geq 0) \} \]
\[ L = \{ \{0, 1, 2\}^* \mid \text{where the digits sum to} \} \text{ a product of 3} \]
\[ L = \{ \{0 \ 1 \ 2\}^* \mid \text{where the digits sum to a product of 3} \} \]
L = \{ (0 \ 1 \ 2)^* \mid \text{where the digits sum to a product of 3} \}
\[ L = \{ (0 \ 1 \ 2)^* \} \]
\[ L_1 = \{ 0 \ 1 \ 2 \}^* \]  
\[ L_2 = \{ 0 (1 \ 2)^* \} \]
\[ L_1 = \{ 0 [1 2]^* \} \]

\[ L_2 = \{ 0 (1 2)^* \} \]

0
012
01212
$L_1 = \{ 0 [1 2]^* \}$
\[ L_2 = \{ 0 \ (1 \ 2)^* \} \]
\[ L_1 = \{ 0 [1 2]^* \} \]

\[ L_2 = \{ 0 (1 2)^* \} \]
\[ L = \{ w \mid w \text{ is of the form } (01)^*2 \} \]
\[ L = \{ w \mid w \text{ is of the form } (01)^*2 \} \]
$L = \{ \ w \mid w \text{ is of the form } (01)^{*}2 \ \}$

$((01)^{*}2)^{*}$
L = \{ w \mid w \text{ is of the form (01)*2 } \} \\
(M \mid m) \text{ onroe} \\
[a-z]*@[a-z]*.com \\
[0-9]^3-[0-9]^3-[0-9]^4
Regular Language

Finite State Automaton

Regular Expression
\[ L = \{ w \mid w \text{ is of the form } (01)^*2 \} \]
Regular Languages, closed under...

1.) **Union** \( L_{AUB} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \)
\[ L_{\text{AUB}} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \]
\[ L_{\text{AUB}} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \]
\[ L_{AUB} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \]

\[ Q_A \times Q_B \]

\[ q_{A0}q_{B0} \quad q_{A1}q_{B0} \]

\[ q_{A0}q_{B1} \quad q_{A1}q_{B1} \]

\[ Q_A \]

\[ q_{A0} \quad q_{A1} \]

\[ Q_B \]

\[ q_{B0} \quad q_{B1} \]
\[ L_{A \cup B} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \]
a

a, b

q

1

b

q

0

b

a

q

0

q

1

a, b

b

q

0

q

1
\( L_1 \) and \( L_2 \) are automata with states \( q_0 \) and \( q_1 \). \( L_1 \cup L_2 \) combines the two automata.
$L_1 \cap L_2$
Regular Languages, closed under...

1.) **Union** \( L_{A\cup B} = \{ x \mid x \in L_A \text{ or } x \in L_B \} \)

2.) **Concatenation**
\[
L_{A \cdot B} = \{ xy \mid x \in L_A \text{ and } y \in L_B \}
\]

3.) **Star** \( L_A^* = \{ x_1x_2\ldots x_n \mid x_i \in L_A \} \)
\[ L = \{ w \mid w \text{ is of the form } (01)^*2 \} \]
Regular Languages, closed under...

1.) **Union** \(L_{AUB} = \{ x \mid x \in L_A \text{ or } x \in L_B \}\)

2.) **Concatenation**

\[L_{A \cdot B} = \{ xy \mid x \in L_A \text{ and } y \in L_B \}\]

3.) **Star** \(L_A^* = \{ x_1x_2\ldots x_n \mid x_i \in L_A \}\)
$\Sigma = \{0, 1\}$

The diagram shows a finite automaton with two states: $q_0$ and $q_1$. The transitions are as follows:
- From $q_0$ to $q_0$ on input 0.
- From $q_0$ to $q_1$ on input 1.
- From $q_1$ to $q_1$ on input 0.
- From $q_1$ to $q_0$ on input 1.
\[ \Sigma = \{a, b\} \]
$L_1 \cdot L_b = \text{A string ending in 1 followed by a string ending in b}$
$$\Sigma = \{0, 1\}$$

$$\Sigma = \{a, b\}$$
$\Sigma = \{0, 1\}$

- $q_0 \xleftarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{b} q_3$

- $\Sigma = \{0, b\}$

- $q_0 \xleftarrow{0} q_0 \xrightarrow{b} q_3$

- $q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{b} q_3$

- $q_2 \xleftarrow{0} q_0 \xrightarrow{b} q_3$

- $q_3$