The languages accepted by Finite Automaton (DFA or NFA) are called "regular languages".

\[ (010)^i \quad i \geq 0 \]

\[ \varepsilon, 010, 010010, (010)^i \]

\[ (01)^i \cup (010)^j \quad i, j \geq 1 \]
The class of regular expressions (r.e.)

**Basics cases:**

1. \( \emptyset \) : empty set
2. \( \varepsilon \) : \( \{ \varepsilon \} \)
3. \( a \in \varepsilon \) : \( \{ a \} \)

If \( r_1 \) and \( r_2 \) are r.e., then

1. \( r_1 + r_2 \) is also a r.e. representing \( r_1 \cup r_2 \)

2. \( r_1 \cdot r_2 \) represents the set of strings \( r_1 \cdot r_2 \)

\[ R_1 \cdot R_2 = \{ w_1 \cdot w_2 \mid w_1 \in R_1, w_2 \in R_2 \} \]

3. \( (r_1)^* \) represents \( R_1 \)
   
   and \( (r_1)^* \) represents \( R_1^* = \varepsilon \bigcup_{i \geq 1} (R_1)^i \)

Nothing else is a r.e.

4. \( a^+ \) is \( \bigcup_{i \geq 1} U(R)^i \)
Examples: \(01 + 100\) \(\leq \{0,1\}\)

1. \(0\) n.r.e. \(1\) r.e.
2. \(0.1\) n.r.e.
3. \(100\) in r.e.
4. The \(01 + 100\) n r.e.

2. \((10 + (101) \cdot 10)^*\) in a r.e.

3. \((011)^* = 011(011)^* + 10\) \(\times\)

1. Regular languages represent some subset of strings.
2. Reg Expr also represent same subset of strings.

1 and 2 are identical
- If $L$ can be recognized by a FA
  then there is a reg exp $r$
  s.t. $r$ represents $L$

- If $r$ uniquely corresponds to $L$
  then we can design a FA $M$
  s.t. $L(M) = L$

**FA:** DFA / NFA

NFA that uses $\varepsilon$ transitions

---

For every NFA with $\varepsilon$ transitions,
there is an NFA w/o $\varepsilon$ transitions
Observation: Using $\varepsilon$ transitions we can build NFA (will $\varepsilon$-transitions) having exactly one Final state.

First Question

$3 + 3 + 4$