Finite Automata

* Deterministic Finite Automaton (DFA):
  - for any given state with any given input character, there is never more than one possible outgoing transition labeled by that character.
  - no ambiguity

* Three steps to convert a regular expression to a DFA
  1) Convert regular expression into NFA
     NFA : Non-deterministic Finite Automaton
     NFA = DFA + a) there may be more than one transition out of a given state labeled by a given character.
       b) there may be ε transitions.
  2) Translate NFA into DFA
  3) Space optimization (final DFA with minimum number of possible states)

* Concatenation: $o \xrightarrow{a} o$ $o \xrightarrow{b} o \Rightarrow o \xrightarrow{a \cdot b} o$ \{ ab \}

* Alternation: $o \xrightarrow{a} o \xrightarrow{a} o \xrightarrow{b} o \xrightarrow{b}$ \{ a \mid b \}

* Kleene Closure: $o \xrightarrow{\varepsilon} o \xrightarrow{a} o \xrightarrow{\varepsilon} o \xrightarrow{\varepsilon} o \xrightarrow{\varepsilon} \{ a^* \}$
Example:

- Strings of all 0's and 1's where there is exactly two 0's:

d) Regular expression: \( 1^* 01^* 01^* \)

1) NFA:

2) DFA

3) Optimized DFA: