Claim: In any set of $n$ hats, all are the same color.

“Proof” by induction:
Base case: If $n = 1$, all hats are the same color.
Inductive case: Assume that the claim holds for all $n \leq k$. Examine any set of $n = k+1$ hats. Remove one hat from the set; the remaining set has $k$ hats and thus is of a single color. Replace the hat and remove a different hat from the set; the resulting set has $k$ hats and thus is of a single color. The two sets overlap, thus they must be of the same color. Therefore, all $k+1$ hats are the same color. $\square$

Claim: In any set of $n$ hats, all are the same color.

“Proof” by induction:
Base case: If $n = 1$, all hats are the same color.
Inductive case: Assume that the claim holds for all $n \leq k$. Examine any set of $n = k+1$ hats. Remove one hat from the set; the remaining set has $k$ hats and thus is of a single color. Replace the hat and remove a different hat from the set; the resulting set has $k$ hats and thus is of a single color. The two sets overlap, thus they must be of the same color. Therefore, all $k+1$ hats are the same color. $\square$
Deterministic Finite Automata

"An organism or an automaton receives stimuli via its sensory receptor organs, and performs actions via its effector organs. To say that certain actions are a response to certain stimuli means, in the simplest case, that the actions are performed when and only when those stimuli occur... So we ask what kind of events are capable of being represented in the state of an automaton."

-- Kleene, "Representation of events in nerve nets and finite automata" (1956)
Elevator Automaton

States

Alphabet

Transition

Start state

Final states

\[ Q \]
set of states

\[ \Sigma \]
the alphabet
**Q** set of states

**Σ** the alphabet

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

\[ q_0 \in Q \] start state

**F** subset of **Q** set of final states
\( Q = \{ 1, 2, 3 \} \)

\( \Sigma = \{ 1, 2, 3 \} \)

\( q_0 = 1 \quad F = \{ 3 \} \)

\( \delta \)

\begin{array}{c|ccc}
1 & 1 & 2 & 2 \\
2 & 1 & 2 & 3 \\
3 & 2 & 2 & 3 \\
\end{array}
What languages do these automata recognize?

$L = \{\varepsilon\}$
\[ L = \{ w \mid w \text{ contains at least 3 1s} \} \]

a) \( L = \{ w \mid w \text{ contains at most two 1s} \} \)

b) \( L = \{ w \mid |w| < 3 \} \)

c) \( L = \{ w \mid |w| < 3 \text{ or contains at least two 1s} \} \)

d) \( L = \{ w \mid w \text{ doesn’t contain 110} \} \)
a) \( L = \{ w \mid w \text{ contains at most two 1s} \} \)

b) \( L = \{ w \mid |w| < 3 \} \)

c) \( L = \{ w \mid |w| < 3 \text{ or contains at least two 1s} \} \)

d) \( L = \{ w \mid w \text{ doesn’t contain 110} \} \)
What automata recognize these languages?

$L = \{ w \mid |w| \text{ is odd and begins with 0 or is even and begins with } 1 \}$

$L = \{ w \mid w = \varepsilon \text{ or every odd position of } w \text{ is a } 1 \}$

$L = \{ w \mid |w| \text{ is odd and begins with 0 or is even and begins with } 1 \}$
$L = \{ w \mid w = \varepsilon \text{ or every odd position of } w \text{ is a 1} \}$

$L = \{ w \mid w \text{ is not the string 11 or 111} \}$

$L = \{ w \mid w \text{ is not the string 11 or 111} \}$
Definition

1. A language \( L \) is regular if some DFA recognizes it.

2. The language \( L(A) \) recognized by a DFA \( A \) is regular.

\[
L_1 = \{ w \mid |w| \text{ is odd and begins with 0 or is even and begins with 1} \}
\]

\[
L_2 = \{ w \mid w = \varepsilon \text{ or every odd position of } w \text{ is a 1} \}
\]

Can we build an automaton that accepts:

\[
L_3 = L_1 \cup L_2
\]

How can we combine two DFAs?
\[ Q = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2\} \quad Q = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2\} \]

\[ \Sigma = \Sigma_1 \cup \Sigma_2 \]

\[ \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)) \quad \delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)) \]

\[ q_0 = (q_1, q_2) \]
$Q = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$

$q_0 = (q_1, q_2)$

$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\}$

Regular languages are closed under union.

if $L_1$ and $L_2$ are regular languages then $L_1 \cup L_2$ is a regular language
WHAT AUTOMATON RECOGNIZES

\[ L = \{ w \mid w \neq \varepsilon \} \]