Intro:
- non-determinism important concept in CS. Does it give extra power? P vs. NP
  \( \text{NON-DET} \neq \text{RANDOMIZATION} \)
- many closure properties of reg. languages can be proved very easily with NFAs.
  \( \text{Often math is all about coming up with the right definition.} \)

\[ q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_3 \xrightarrow{1} \]
\[ q_3 \xrightarrow{0} q_4 \]

\( 001 \) accepts
\( 01011 \) accepts
\( 00100 \) rejects

Comparison with DFAs:
* not all possible transitions present.
* multiple transitions with same label from a state.
* \( \epsilon \) transitions.

Rule for accepting/rejecting:
Accept \( w \) if there is a "valid" path that leads to an accepting state.
Reject \( w \) if all paths lead to a rejecting state.
Example computation trace

0 0 1 1 0

q_0 q_1 q_1 q_2 q_2 q_4

q_3 q_1 q_2

q_4

q_3 q_1 q_1

q_4

q_3

q_3
**Def:** An NFA is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) where

- \(Q\) finite non-empty set (set of states)
- \(\Sigma\) is a "" alphabet"
- \(\delta: Q \times \Sigma \rightarrow \mathcal{P}(Q)\) (transition function)
- \(q_0 \in Q\) (starting state)
- \(F \subseteq Q\) (set of accepting states)

**Exercise:** Define formally the notion of an NFA accepting a string.

**Drawing NFAs**

All binary strings that have either odd \# 0's or \# 1's is not a multiple of 3.

\[ L = \{ w \in \{0,1\}^* : \text{lw}l \geq 3 \quad \text{and} \quad w_{n-2} = 1 \quad \text{where} \quad n=\text{lw}l \} \]

**Do this last!**

\[ L_k = \{ w \in \{0,1\}^* : lw_1 \geq k \quad \text{and} \quad w_{n-(lw)} = 1 \quad \text{where} \quad n = lw \} \]

Best DFA needs 2\(k\) states.
L: all binary strings that end with 101.
L: all binary strings that contain 101 as a substring.

Thm: NFA = DFA (any language recognized by a NFA is regular)

Proof Sketch: Threading idea works.

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. (Suppose $N$ has no $e$-transitions.) Construct DFA $M = (Q', \Sigma, \delta', q_0', F')$ as follows:

$Q' = P(Q)$

$\delta': Q' \times \Sigma \rightarrow Q'$, $\forall SEQ', a \in \Sigma$, $\delta'(s,a) = \bigcup_{s \in S} \delta(s,a)$

$q_0' = \{q_0\}$

$F' = \{SEQ': S \cap F \neq \emptyset\}$

If $N$ has $e$-transitions, define, for $SEQ'$, $E(S) = \{q \in Q: q$ can be reached from a state in $S$ using zero or more $e$-transitions $\}$

$E(S, a) = E(\delta'(S, a))$

$q_0'' = E(\{q_0\})$

Comment: # States in DFAs vs NFAs
Closure Properties of Regular Languages

Closure under union:

Build NFA

Closure under concatenation

Closure under star

First attempt
Closure under reversal

reverse every transition.