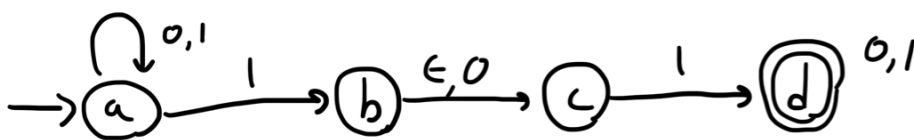


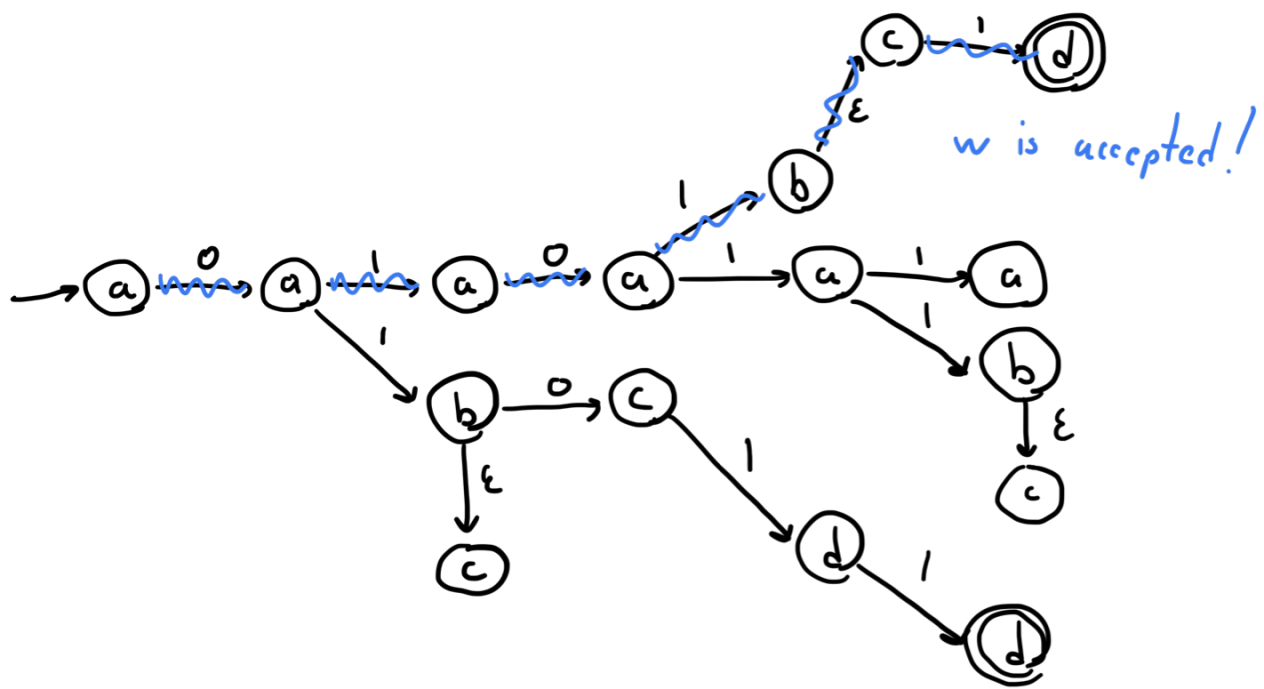
- Determinism - Given the current state, there is one next state, or "one future"
- Nondeterminism - Given the current state, there are multiple next states, which all exist simultaneously
(equivalent to saying that we always choose the "correct" next state)
- There are two kinds of FSMs
 - ↳ deterministic finite automaton (DFA)
 - ↳ nondeterministic finite automaton (NFA)

NFA's can have multiple edges with the same label; can have epsilon edges



$L = \{w \mid w \text{ contains } 11 \text{ or } 101 \text{ as a substring}\}$

Let $w = 01011$ and imagine a "computation tree".



Formal Def

An NFA is a 5-tuple

1. Q is a finite set of states
2. Σ = alphabet
3. $\delta: Q \times \Sigma_{\epsilon} \rightarrow P(Q)$ is the transition function
($\Sigma_{\epsilon} = \Sigma \cup \{\epsilon\}$)
4. $q_0 \in Q$ start state
5. $F \subseteq Q$ final state(s)

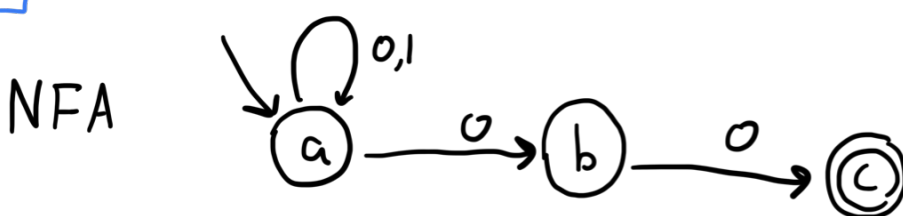
Two differences
1. δ is now in range of $P(Q)$
2. allows ϵ transitions

Power set of Q
 $P(\{a,b,c\}) = \left\{ \emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{a,c\}, \{a,b,c\} \right\}$

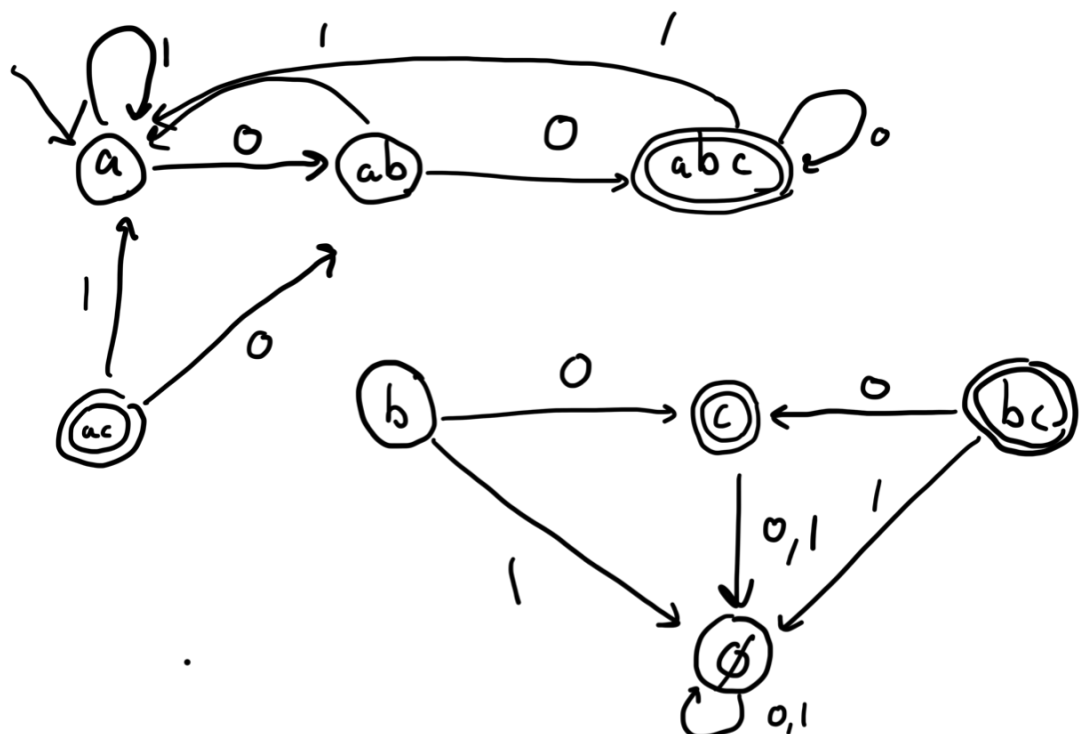
DFA, NFA equivalence

They have the same expressive power!
We convert via "subset construction".

Ex 1

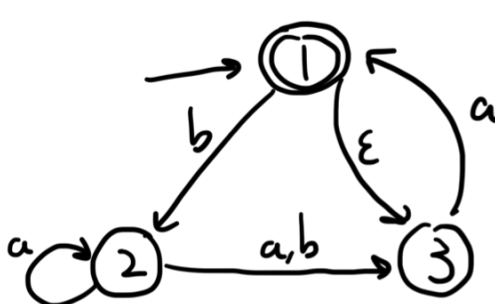


DFA



Ex 2

NFA



DFA

