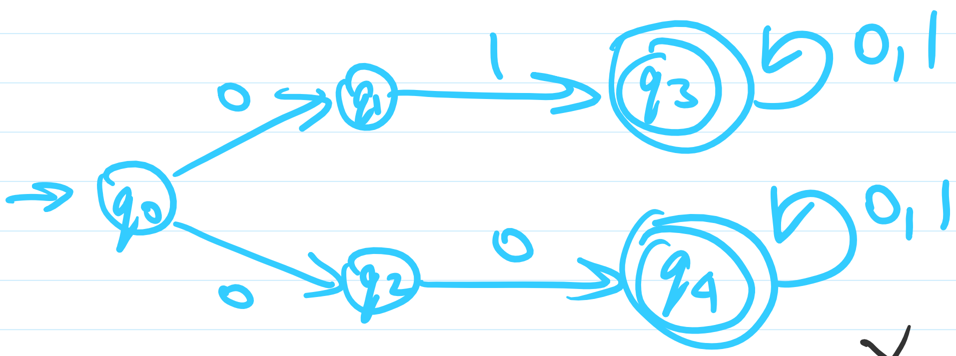
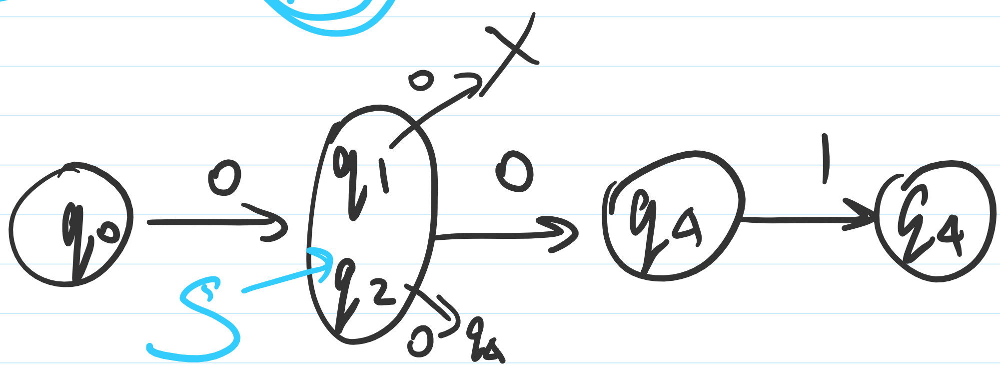
NFA & DFA Equivalence

Claim: A language $L \subseteq \Sigma^*$ is accepted by some DFA iff L is accepted by some NFA.

Proof: DFA $\rightarrow$ NFA ✓
NFA $\rightarrow$ DFA



001



Subset Construction: NFA $\rightarrow$ DFA

NFA $A = (Q, \Sigma, \delta, q_0, F)$

$\downarrow$

DFA $B = (Q', \Sigma, \delta', q_0', F')$

$Q' = \mathcal{P}(Q)$ power set of Q

$\forall a \in \Sigma$

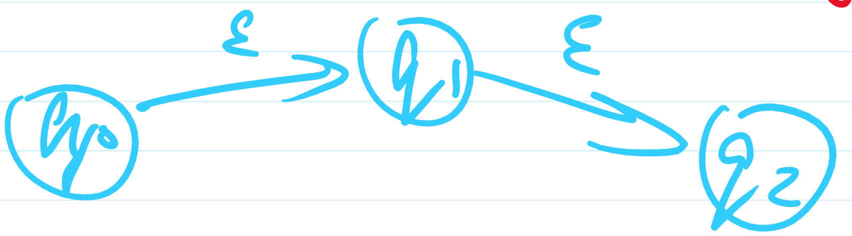
$\rightarrow \delta'(S, a) = \bigcup_{q \in S} E(\delta(q, a))$

$q_0' = E(\{q_0\})$, $F' = \{S \in Q' \mid S \cap F \neq \emptyset\}$

ε-closure E:

$\forall S \subseteq Q$, $E(S) = \{q \mid q$ can be reached from some $s \in S$ by a path of ε-edges$\}$

Ex:



$S = \{q_0\}$

$E(S) = \{q_0, q_1, q_2\}$

In general, NFA on K states

In general, NFA on K states
Then can get a DFA on 2^K states, & it's necessary!

<u>Conclusion</u>: Reg. languages = DFA
= NFA

FA (DFA, NFA)

<u>Regular Expressions</u>

- $\emptyset$, ε, a $\forall a \in \Sigma$

Expression	Language
$\emptyset$	$\emptyset$
ε	$\{\varepsilon\}$
a	$\{a\}$

- <u>Inductive Step</u>:

  A, B reg. expressions

$A \cup B \rightarrow L(A) \cup L(B)$

$A \circ B \rightarrow L(A) \circ L(B)$

$A^* \rightarrow (L(A))^*$

$(0 \cup 1)^*$ all bin. strings

$0(0 \cup 1)^* 1$

01 ✓

001 ✓

101 ✗

00001 ✓

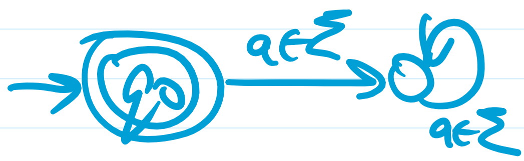
NFA

Constructing NFA from a reg exp. $\mathcal{E}$.

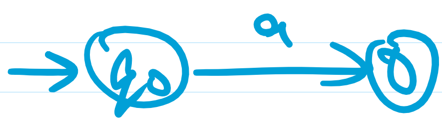
Base Case:

$\emptyset \rightarrow A_\emptyset$ $\quad L(A_\emptyset) = \emptyset$

$\varepsilon \rightarrow A_\varepsilon$ $\quad L(A_\varepsilon) = \{\varepsilon\}$



$a,\ a \in \Sigma \rightarrow A_a$ $\quad L(A_a) = \{a\}$



Ind. Step:

- $A \cup B$

Have NFA for A,

Have NFA for A,
NFA for B by Ind. Hypo.
Build from them an NFA for $A \cup B$
(as in the last lecture).
Other cases: $A \circ B$ and A^*
are similar!