Theory of Computation, CSCI 438 spring 2019
Nondeterminism & Relation between NFAs and DFAs, pg. 47-63, Jan. 25

Write the definition of an NFA that is similar to the definition of a DFA.

Deterministic finite automaton - DFA	Nondeterministic finite automaton - NFA
$M = (Q,\Sigma,\delta,q_0,F)$	$M = (Q,\Sigma,\delta,q_0,F)$
* Q - finite set of states * Σ - finite set of symbols, input alphabet * $\delta: Q \times \Sigma \rightarrow Q$, transition function * $q_0 \in Q$, initial state * $F \subseteq Q$, set of accept states	* Q - finite set of states * Σ - finite set of symbols, input alphabet * $\delta: Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$, transition function ($\Sigma_\varepsilon$ is Σ augmented by ε which indicates that the machine can move forward without an input symbol) * $q_0 \in Q$, initial state * $F \subseteq Q$, set of accept states

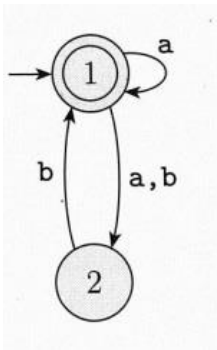
Do the same for strings accepted by an NFA.

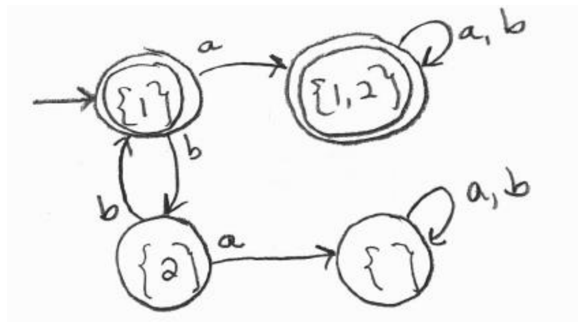
M accepts string $w \in \Sigma^*$ iff $\delta^*(q_0,w) \in F$	M accepts string $w \in \Sigma^*$ iff there exists some sequence such that $\delta^*(q_0,w) \in F$

Exercise 1.16 a & b

1.16 Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.

a.





b.