

To construct DFA M from NFA N:

Let $N = (\underline{Q}, \Sigma, \underline{\delta}, q_0, \underline{F})$. Define

$M = (\underline{P(Q)}, \Sigma, \underline{\delta'}, q', \{X \subseteq Q \mid X \cap F \neq \emptyset\})$

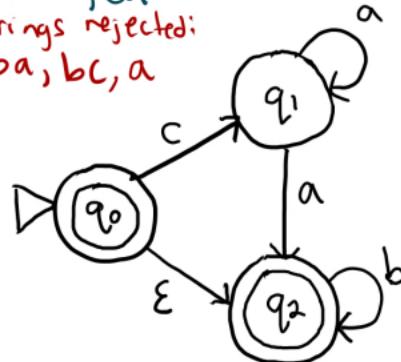
all possible combinations
of states from NFA

Examples of strings accepted:

$cab, \epsilon, b, bbb, ca$

Examples of strings rejected:

cb, ba, bc, a



where $q' = \{q \in Q \mid q = q_0 \text{ or is accessible from } q_0 \text{ by spontaneous moves in } N\}$

state in DFA (labeled by a set of states from NFA)

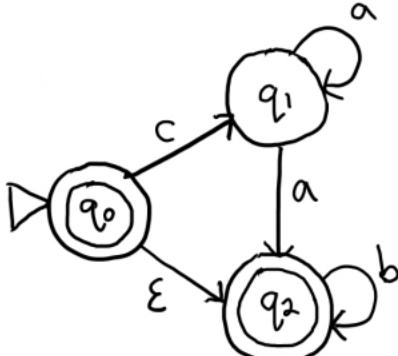
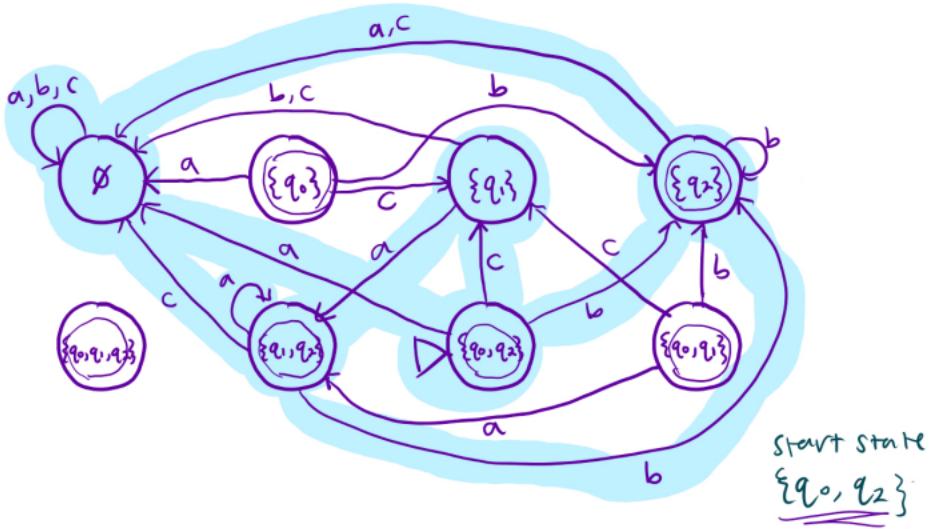
$\delta'(\underline{(X, Y)}) = \{q \in Q \mid q \in \delta((r, x)) \text{ for some } r \in X \text{ or } q \text{ is accessible from such an } r \text{ by spontaneous moves in } N\}$

label for a single state in the new DFA

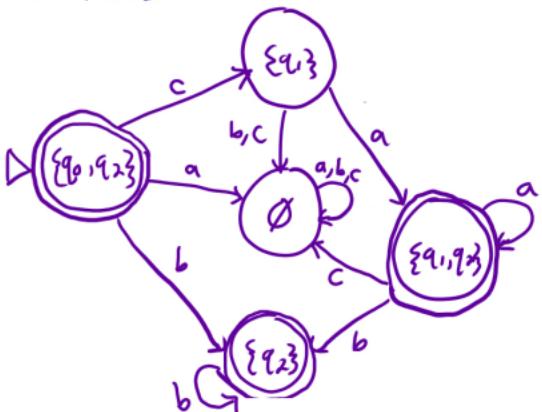
label for the state to transition to

$$\delta'(\underline{(q_0, q_2)}, c) = \delta((q_0, c)) \cup \delta((q_2, c)) = \{q_1\} \cup \emptyset = \boxed{\{q_1\}}$$

different inputs $\rightarrow \delta'(\{q_0\}, c)$

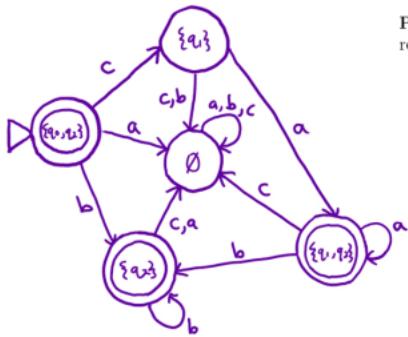


Remove unreachable states:



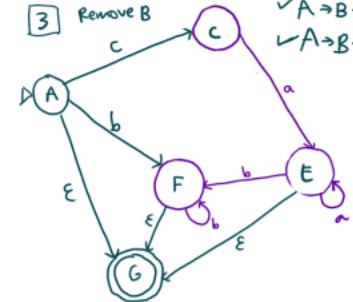
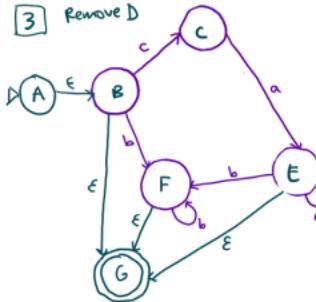
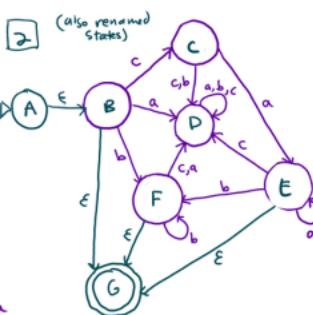
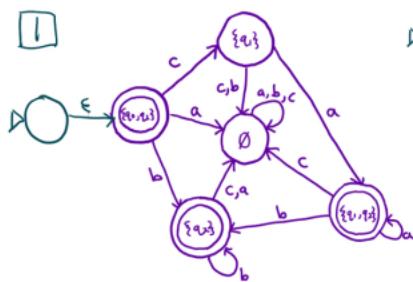


Proof idea: Trace all possible paths from start state to accept state. Express labels of these paths as regular expressions, and union them all.



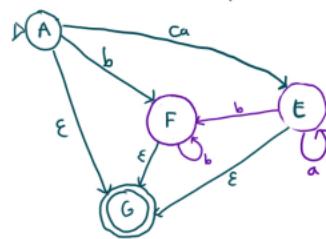
1. Add new start state with ϵ arrow to old start state.
2. Add new accept state with ϵ arrow from old accept states. Make old accept states non-accept.
3. Remove one (of the old) states at a time: modify regular expressions on arrows that went through removed state to restore language recognized by machine.

1



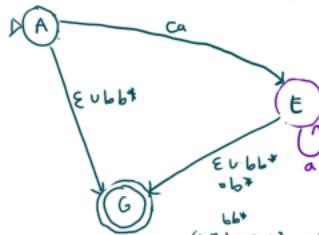
3 Remove C

$\checkmark A \rightarrow C \rightarrow E$



3 Remove F

$\checkmark A \rightarrow F \rightarrow G$
 $\checkmark E \rightarrow F \rightarrow G$



3 Remove E

$\checkmark A \rightarrow E \rightarrow G$



$caa^*b^* \cup b^*$