To construct DFA \( M \) from NFA \( N \):

\[ \exists q_0, q_1, q_2, q_3 \]

Let \( N = (Q, \Sigma, \delta, q_0, F) \). Define

\[ M = (P(Q), \Sigma, \delta', q_1', \{X \subseteq Q \mid X \cap F \neq \emptyset\}) \]

where \( q_1' = \{q \in Q \mid q = q_0 \text{ or is accessible from } q_0 \text{ by spontaneous moves in } N\} \)

Examples of strings accepted:
\[ ca \in b \]

Examples of strings rejected:
\[ c, bc, aaa \]

State in DFA (labeled by a subset of states from NFA \( N \))

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

Symbol to read

\[ \delta'((q_0, q_2, 3, c)) = \delta((q_0, c)) \cup \delta((q_2, c)) = \{q_1, 3\} \cup \emptyset = \{q_1, 3\} \]
**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 $\varepsilon$ arrow to old start state.
2. Add new accept state with $\varepsilon$ 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.

Examples of removing states:

$A \rightarrow B \rightarrow F$
$A \rightarrow B \rightarrow C$
$A \rightarrow B \rightarrow G$

$\{a^n b^n \mid n \geq 1\} \cup \{b^n \mid n \geq 0\} = \{a^n b^n \mid n \geq 0\}$

$caab^* \cup b^*$