A nondeterministic finite automaton (NFA) is a 5-tuple \( N = (Q, \Sigma, \delta, s, F) \) where:
- \( Q \) is a finite set of states
- \( \Sigma \) is a finite alphabet
- \( s \in Q \) is the start state
- \( F \subseteq Q \) is the set of final states
- \( \delta \subseteq (Q \times \Sigma) \times Q \) is the instruction set

A DFA is an NFA in which no two instructions \( (p, c, r) \) and \( (q, d, r') \) have both \( p = q \) and \( c = d \), and for which every \( q \in Q, c \in \Sigma \), has an instruction \( (q, c, r) \).

The text uses as default the NFA with \( \varepsilon \)-transitions, where
\[ \delta \subseteq (Q \times (\Sigma \cup \{\varepsilon\})) \times Q \]
Typical instructions:
\[ (p, c, q) \quad c \in \Sigma \]
or \( (p, \varepsilon, q) \) also allowed.

Text defines \( \delta : (Q \times (\Sigma \cup \{\varepsilon\})) \rightarrow P(Q) \)
\[ \delta(p, c) = \{ \text{all } q \text{ s.t. } (p, c, q) \text{ is an instruction} \} \]

\[ \delta(p, \varepsilon) = \{ q \text{ s.t. } (p, \varepsilon, q) \text{ is allowed} \} \]
Examples: Let $L = \{ X \in \{ a, b \}^* : \text{the substring } ab \text{ occurs in } X \text{ at most once} \}$.

Note $\tilde{L} = \{ X \in \{ a, b \}^* : ab \text{ occurs in } X \text{ at least twice} \}$.

Design a DFA $M_1$ s.t. $L(M_1) = \tilde{L}$ first.

- $b$ changes nothing at start
- $a$ does not kill progress
- Halfway to the second $ab$.
- Two $ab$s seen, good enough.

Then $M_0$ s.t. $L(M_0) = L$ is obtained by complementing accept states

$Q_0 = Q_1 = \{1, 2, 3, 4, 5\}$

$F_1 = \tilde{F_0} = \emptyset$, $F_0 = Q \setminus F_1 = \{1, 2, 3, 4, 5\}$

Claim: This $N_1$ is correct because $\tilde{L} = (a^*b)^*ab(a^*b)^*ab(a^*b)^*$

To execute $N_1$ on a string such as $X = abbabaab$, we keep track of all the possible states $N_1$ could be in after processing the $i$th character.
Process for converting the NFA $N$, which has no ε-quests, into an equivalent DFA. Use the functional view of $S$. DFA $M$ is $(Q,Σ,Δ,S,δ)$ where

$Δ: Q × Σ → Q$.

$Δ(p, c) → \{ q : \text{for some } p ∈ P, (p, c, q) ∈ S \}$

$P ≤ Q$, is the $Q$ of $N_1$.

$δ = \{ 1 \}$

$QF = \{ R ∈ Q : R \text{ includes an accepting state} \}$

$S = \{ 1 \}$

$Δ(\{1, 2, 3, 4\}, a) = \{(1, a)vS(2, a)\} = \{\{1, 2\}v\{\}\} = \{1, 2\}$.

$Δ(\{1, 2\}, b) = \{(1, b)vS(2, b)\} = \{\{1\}v\emptyset\} = \{1\}$.

$S = \{1\}$

$5\{1, 3, 5\}$

$\emptyset \{1, 3, 5\}$

$\{1, 2, 3, 4\}$

This is a correct DFA $M$, and we can complement it as before.

**Extra/Footnote:** Note that this last DFA has two accepting states, not just the one of our earlier $M$, designed by logical reasoning. You can of course just merge $\{1, 3, 5\}$ and $\{1, 2, 3, 4, 5\}$ together. What this means is that the “NFA to DFA” algorithm does not always produce a minimum DFA.