CS304: Automata and Formal Languages

Lec 7

DFA = NFA

Rachit Nimavat

August 12, 2025

Outline

Alphabet Σ: A finite, non-empty set of symbols

String w: A finite sequence of symbols taken from an alphabet

Language L: A set of strings over a particular alphabet

Consider $\Sigma = \{0, 1\}$ and $L = \{0\}^* \{1\} \equiv 0^*1$.

Deterministic Finite Automata (DFA)

Formal Definition

DFA is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$

‘symbol’	description	in our code
Q	finite set of states	`state` $\in \{0, 1, 2\}$
Σ	underlying finite alphabet	$\Sigma = \{$‘0’, ‘1’$\}$
δ	transition function between states	the ‘core logic’ in C code
$q_0 \in Q$	start state	`state` $= 0$
$F \subseteq Q$	accepting states	`state` $= 1$

Deterministic Finite Automata (DFA)

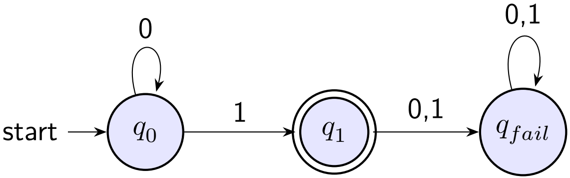
Visualizing DFA

State Diagram is the intuitive way we visualize and work with DFAs

Conventions:

- States Q are circles
- Start state (q_0) has an incoming arrow labeled start
- Accepting states (F) are double circles
- Transitions (δ) are arrows between states, labeled with input symbols from Σ

Our DFA for $L = 0^*1$:



Non-Deterministic Finite Automata (NFA)

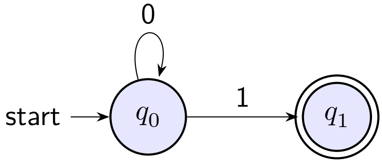
Visualizing NFA

State Diagram is the intuitive way we visualize and work with NFAs

Conventions:

- States Q are circles
- Start state (q_0) has an incoming arrow labeled start
- Accepting states (F) are double circles
- Transitions (δ) are arrows between states, labeled with input symbols from Σ

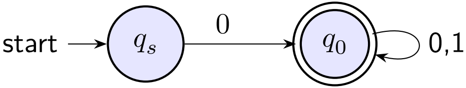
Our NFA for $L = 0^*1$:

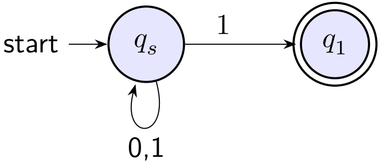


NFA Example

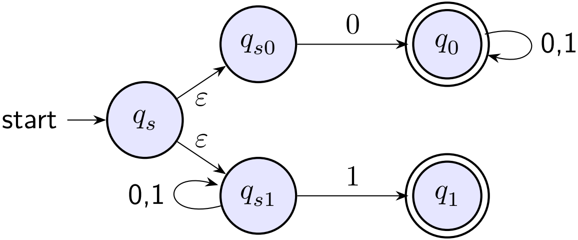
$L_1 = \{w \in \{0,1\}^* \mid w \text{ ends with } 1\}$

$L_0 = \{w \in \{0,1\}^* \mid w \text{ starts with } 0\}$





$L = L_0 \vee L_1 = \{w \in \{0,1\}^* \mid w \text{ starts with } 0 \text{ or } w \text{ ends with } 1\}.$



NFA vs DFA

NFA

$N = (Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto 2^Q$

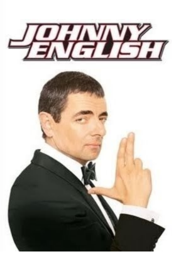
Multiple Transitions. Possibly multiple outgoing arrows for same symbol
Zero Transitions. Possibly no outgoing arrow for a symbol (that path "dies" has license to kill)
ε-Transitions. Can change state without consuming an input symbol

DFA

$D = (Q, \Sigma, \delta, q_0, F)$

$\delta : Q \times \Sigma \mapsto Q$

No such powers!



Are NFAs really more powerful than DFAs?

NFA = DFA

Theorem 2.1

A language is recognized by an NFA if and only if it is recognized by a DFA.

To prove that the set of languages recognized by NFAs is the same as the set of languages recognized by DFAs, we must show two things:

1. **(Easy Part)** For every DFA, there is an equivalent NFA that accepts the same language. **trivial!!**
2. **(Hard Part)** For every NFA, there is an equivalent DFA that accepts the same language. **we'll prove this today**

Proof Idea: Assume NFA doesn't have any ε-transitions

Let the NFA be $N = (Q, \Sigma, \delta, q_0, F)$. We will construct a DFA $D = (Q', \Sigma, \delta', q'_0, F')$ that recognizes the same language.

The Construction of D.

States: $Q' = 2^Q$ each state of DFA corresponds to a subset of states of NFA

Start State: $q'_0 = \{q_0\}$

Accepting States: $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\}$

Transition Function: For a given state $q' \in Q'$ in DFA and an input symbol $a \in \Sigma$, the next state is one corresponding to the union of all states that the NFA could transition to from any state in q'.

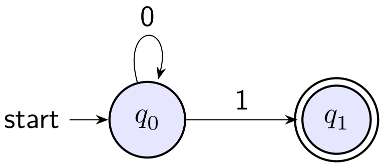
$$\delta'(q', a) = \bigcup_{q \in q'} \delta(q, a)$$

Proof of Correctness. Need to prove that for all strings w over Σ, $\hat{\delta}'(q'_0, w) = \hat{\delta}(q_0, w)$. Recall, $\hat{\delta}$ is the extended transition function. **On board!**

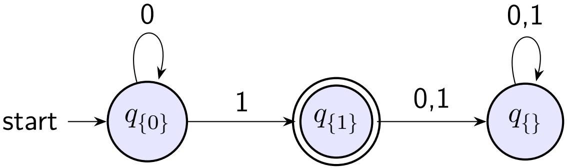
Example 1

$L = 0^*1.$

NFA



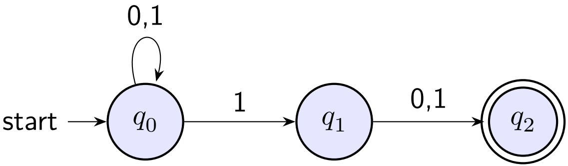
DFA



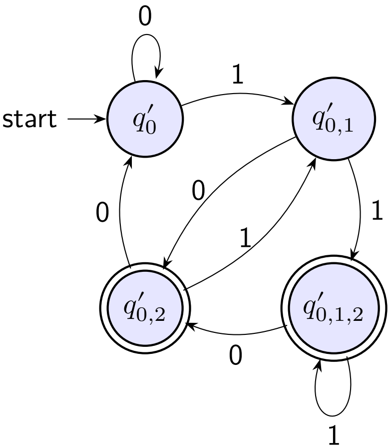
Example 2

$L = \{w \in \{0,1\}^* \mid \text{second-last symbol of w is 1}\}$.

NFA



DFA



ε-closure

ε-closure is the set of all states reachable from a given state using only ε-transitions

- For a single state q, $\varepsilon_{\text{closure}}(q)$ is the set of states that can be reached from state q by following zero or more ε-transitions

  notice: $q \in \varepsilon_{\text{closure}}(q)$

- For a set $S \subseteq Q$ of states,

$$\varepsilon_{\text{closure}}(S) := \bigcup_{q \in S} \varepsilon_{\text{closure}}(q)$$

Proof Idea: With ε-transitions

Let the NFA be $N = (Q, \Sigma, \delta, q_0, F)$. We will construct a DFA $D = (Q', \Sigma, \delta', q_0', F')$ that recognizes the same language.

The Construction of D.

States: $Q' = 2^Q$, the sets of states of NFA
Start State: $q_0' = \varepsilon_{\text{closure}}(q_0)$
Accepting States: $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\}$
Transition Function: For a given state $q' \in Q'$ in DFA and an input symbol $a \in \Sigma$, the next state is one corresponding to the union of all states that the NFA could transition to from any state in q'.

$$\delta'(q', a) = \bigcup_{q \in q'} \varepsilon_{\text{closure}}(\delta(q, a))$$

Proof of Correctness. Need to prove that for all strings w over Σ, $\hat{\delta}'(q_0', w) = \hat{\delta}(q_0, w)$. Recall, $\hat{\delta}$ is the extended transition function. **On board!**