Lecture 5 — ε-NFA

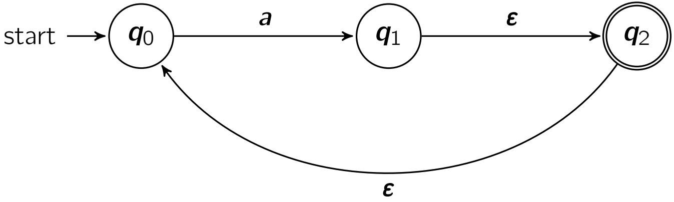
CSE205: Theory of Computation

Minseok Jeon

2026 Fall

NFA with ε-Transitions

An ε-NFA is an NFA that is also allowed to make transitions on the empty string ε (i.e. without reading any input symbol).



$$M = (\{q_0, q_1, q_2\},\ \{a\},\ \delta,\ q_0,\ \{q_2\})$$

$$\delta(q_0, a) = \{q_1\} \qquad \delta(q_0, \varepsilon) = \emptyset$$
$$\delta(q_1, a) = \emptyset \qquad \delta(q_1, \varepsilon) = \{q_2\}$$
$$\delta(q_2, a) = \emptyset \qquad \delta(q_2, \varepsilon) = \{q_0\}$$

NFA with ε-Transitions

Definition (ε-NFA)

An ε-NFA is a 5-tuple

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

where

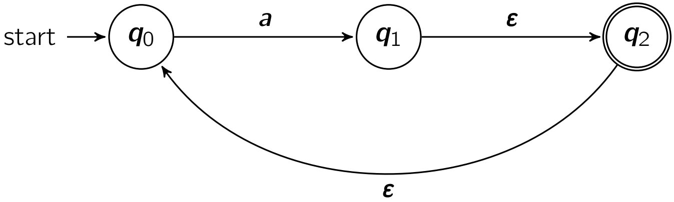
- Q: a finite set of states
- Σ: a finite set of input symbols (the input alphabet)
- $q_0 \in Q$: the initial state
- $F \subseteq Q$: a set of final states
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^{Q}$: transition function

The only change from an NFA is that ε is now an allowed input to δ.

Extended Transition Function

Definition

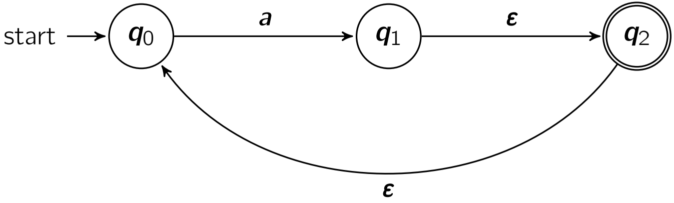
For an ε-NFA, the extended transition function $\hat{\delta} : Q \times \Sigma^* \to 2^Q$ is defined so that $\hat{\delta}(q_i, w)$ contains q_j iff there is a path in the transition graph from q_i to q_j labeled by w (where ε-edges contribute nothing to the label).



$$\hat{\delta}(q_1, a) = \{q_0, q_1, q_2\}, \qquad \hat{\delta}(q_2, \varepsilon) = \{q_0, q_2\}, \qquad \hat{\delta}(q_2, aa) = \{q_0, q_1, q_2\}.$$

ε-Closures

EClose(q): the set of states reachable from q using only ε-transitions.

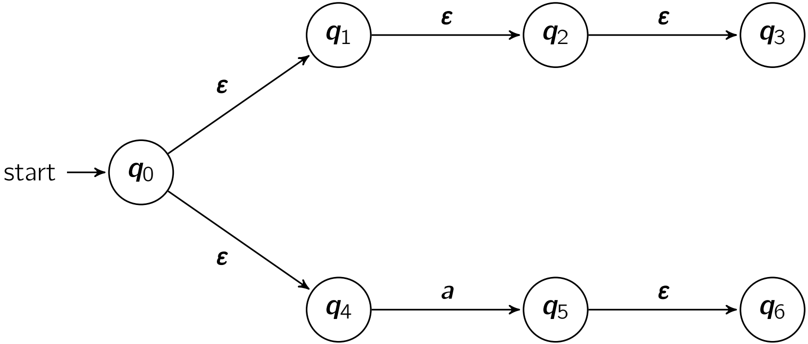


$$\mathrm{EClose}(q_0) = \{q_0\}, \quad \mathrm{EClose}(q_1) = \{q_0, q_1, q_2\}, \quad \mathrm{EClose}(q_2) = \{q_0, q_2\}.$$

Recursive definition

- **(Basis)** $q \in \mathrm{EClose}(q)$.
- **(Induction)** If $p \in \mathrm{EClose}(q)$ and there is an ε-transition from p to r, then $r \in \mathrm{EClose}(q)$.

Example



$$\mathrm{EClose}(q_0) = \{q_0, q_1, q_2, q_3, q_4\}$$

$$\mathrm{EClose}(\{q_0, q_5\}) = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$$

EClose extends to sets by taking the union of the closures.

Formal Definition of $\hat{\delta}$

$$\hat{\delta} : Q \times \Sigma^* \to 2^Q$$

- **(Basis)**
$$\hat{\delta}(q, \varepsilon) = \text{EClose}(q)$$
- **(Induction)** $s = ua$ with $a \in \Sigma$:
$$\hat{\delta}(q, ua) = \text{EClose}\left(\bigcup_{s_i \in \hat{\delta}(q,u)} \delta(s_i, a) \right)$$

We take an ε-closure both at the start and after every input symbol.

Language of an ε-NFA

An ε-NFA $M = (Q, \Sigma, \delta, q_0, F)$ *accepts* a string w if

$$\hat{\delta}(q_0, w) \cap F \neq \emptyset,$$

and the language of M is

$$L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

From ε-NFA to DFA

Given an ε-NFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$, define a DFA

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

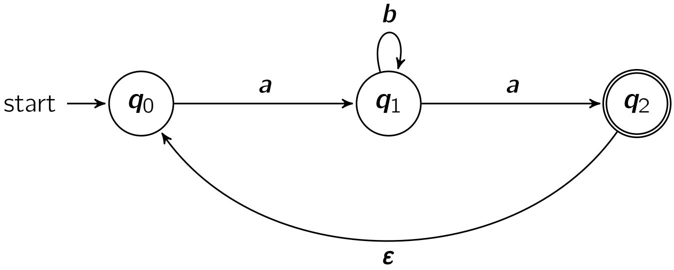
where

- $Q_D = \{S \subseteq Q_E \mid S = \mathrm{EClose}(S)\}$
- $q_D = \mathrm{EClose}(q_0)$
- $F_D = \{S \in Q_D \mid S \cap F_E \neq \emptyset\}$
- for each $S \in Q_D$ and $a \in \Sigma$:

$$\delta_D(S, a) = \mathrm{EClose}\left(\bigcup_{p \in S} \delta_E(p, a)\right)$$

Exercise

Convert the following ε-NFA into an equivalent DFA.



Hint: start from $q_D = \mathrm{EClose}(q_0) = \{q_0\}$ and apply the subset construction with ε-closures after each symbol.

Equivalence of ε-NFA and DFA

Theorem

A language L is accepted by some ε-NFA if and only if L is accepted by some DFA.

Proof.

- **(If)** Easy: a DFA $D = (Q, \Sigma, \delta_D, q_0, F)$ becomes an equivalent ε-NFA by taking $\delta(q, a) = \{\delta_D(q, a)\}$ and $\delta(q, \varepsilon) = \emptyset$.
- **(Only if)** By the ε-NFA to DFA construction above (left as an exercise).

$\square$

Hence DFA, NFA, and ε-NFA are all *equivalent* in power: they accept exactly the regular languages.