Theory of Computation

Tutorial - NFAs

Cesare Spinoso-Di Piano

Plan for today

1. NFAs
2. NFA-to-DFA

NFAs

NFAs

Formal definition of an NFA

Definition. A **nondeterministic finite automaton (NFA)** M is a 5 element tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

- Q is the set of all states
- Σ is the alphabet
- * δ is the transition function $\delta : Q \times \{\Sigma \cup \{\lambda\}\} \to 2^Q$
- q_0 is the (unique) initial state
- F is the set of final states

An NFA is a machine that reads an input string and either accepts it or rejects it.

***Unlike a DFA:** The transition function of an NFA can accept λ and **always** returns a set.

NFAs as language accepters

Non-Determinism. Let N be an NFA with input string w.
Non-determinism allows N to try *all possible walks*. If at least one walk leads to a final state, then N accepts w.

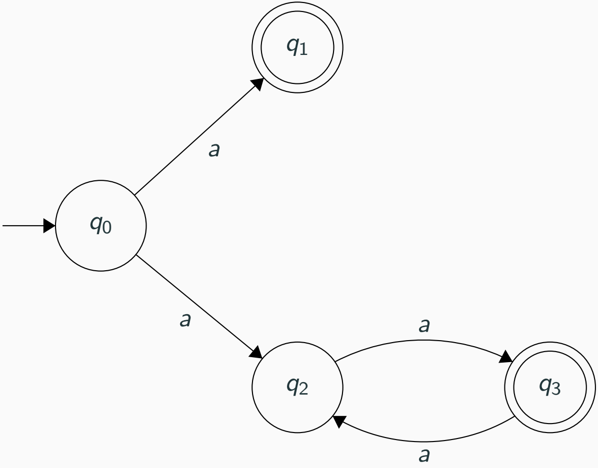
Definition. Formally, given an NFA N the language accepted by N is

$$L(N) = \{w \in \Sigma^* : \delta^*(q_0, w) \cap F \neq \emptyset\}$$

where δ^* is the extended transition function $\delta^* : Q \times \Sigma^* \to 2^Q$.

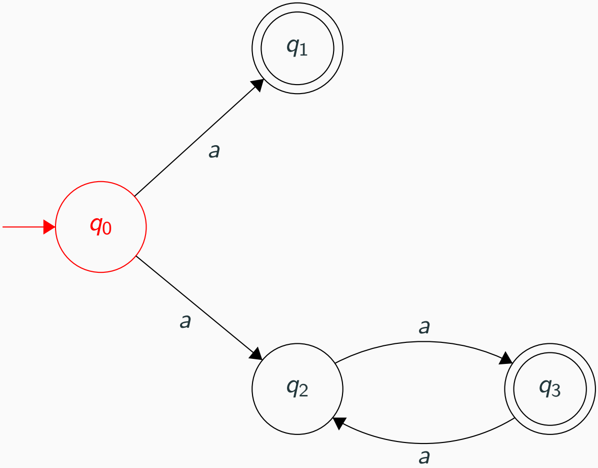
Example

Example. Consider the following NFA over the alphabet $\Sigma = \{a\}$.



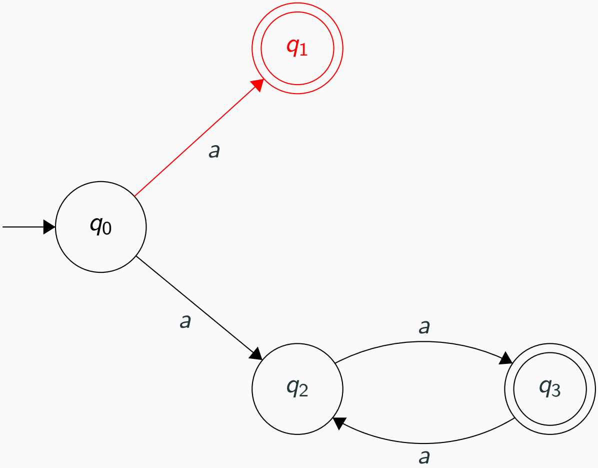
Example - Tracing Input

Input String: ^aa



Example - Tracing Input

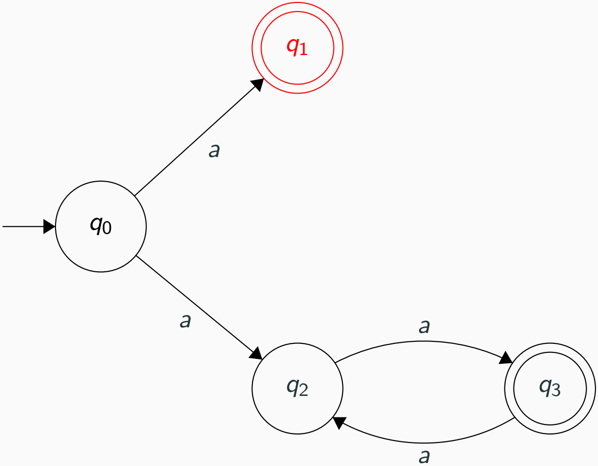
Input String: aa



Example - Tracing Input

Input String: aa

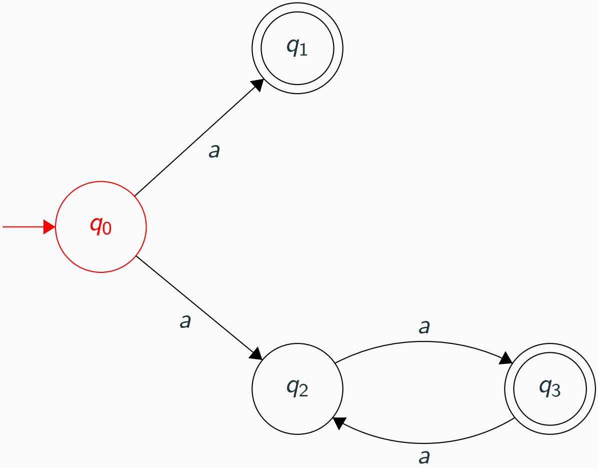
$\delta(q_1, a) = \emptyset$, no where to go. Are we done?



Example - Tracing Input

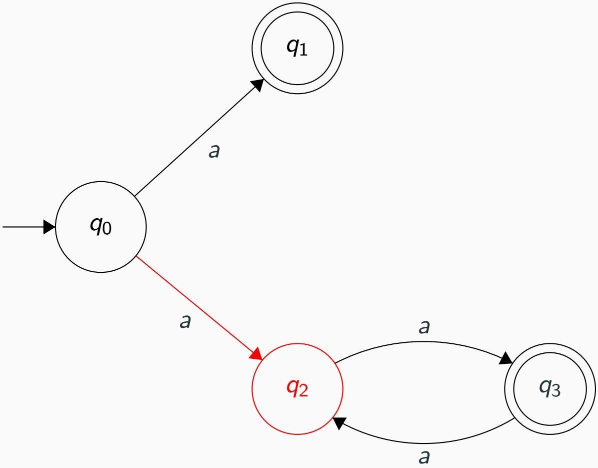
Input String: ^aa

Are we done? No, **M** tries another walk.



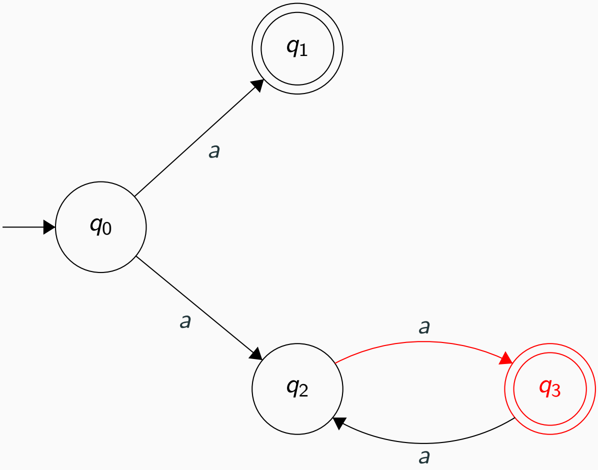
Example - Tracing Input

Input String: aa



Example - Tracing Input

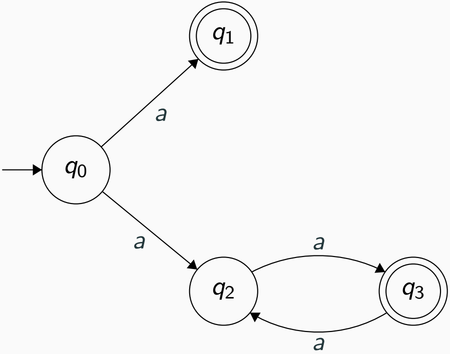
Input String: aa



One of the possible walks ends in a final state, **M** accepts this string.

Example - Tracing Input

Input String: aaa

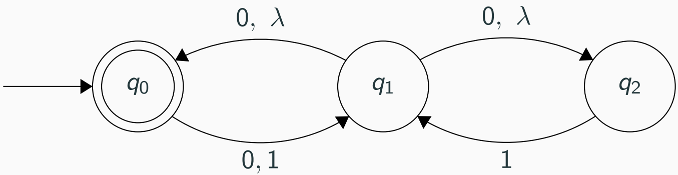


In both traces (top and bottom), do not end up in a final state. **M** rejects this string.

What is the language accepted by this NFA?

Exercise

Exercise. Given the following NFA **M**,



- What is $\delta^*(q_0, 01)$?
- Is 01 accepted by this NFA?
- What is the language accepted by this NFA?

Exercise

Exercise. Create an NFA that accepts the language $\{xwx : x \in \{0,1\}, w \in \{0,1\}^*\}$.

NFA-to-DFA

NFA-to-DFA

NFA-to-DFA Algorithm

Given an NFA $N = (Q, \Sigma, \delta, q_0, F)$, how can we convert it to a DFA $M = (Q', \Sigma, \delta', S_0, F')$ such that $L(N) = L(M)$?

Procedure:

Step 1: Create an initial state S_0. Assign it to $\{q_0\}$.

Step 2: For each symbol $a \in \Sigma$, find all the states in N that can be reached from each of the states in S_0. The set of reachable states in N will become a state in M i.e. $\delta'(S, a) = \bigcup_{p \in S_0} \delta(p, a)$.

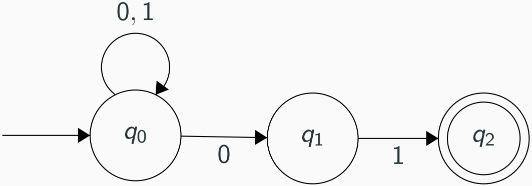
Step 3: Repeat Step 2 on every new state S (which are *sets* of states of N) that is generated. Repeat until no new states are found.

Step 4: Assign Q' to the set of states generated in Step 3. δ' is defined as above.

The initial state for the DFA will be $S_0 := \{q_0\}$. The final states of the DFA will be all those states S that contain a final state from F (from the original N). Finally, if the original NFA N accepts λ, make S_0 a final state.

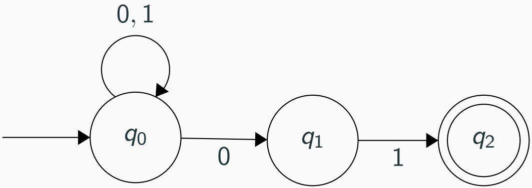
Example

Example. Consider the following NFA N. $\Sigma = \{0, 1\}$. What is the language accepted by N?



Example

Example. Convert the NFA N to a DFA M such that $L(N) = L(M)$.



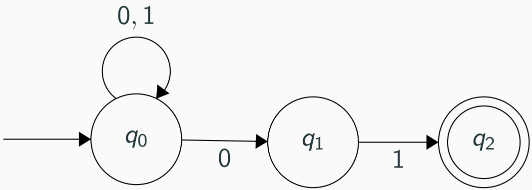
Example

Example. Converting the NFA N to a DFA M using the procedure presented above.

Step 1. Begin with the start state $S_0 \coloneqq \{q_0\}$.

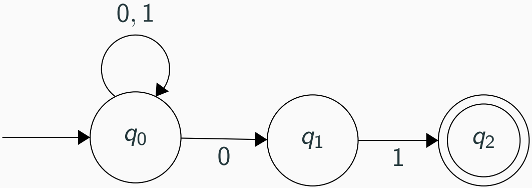
Step 2. Find all the states that can be reached from S_0 for each letter in Σ.

$\delta'(S_0, 0) = \underbrace{\{q_0, q_1\}}_{\text{New state!}}$, $\delta'(S_0, 1) = \{q_0\}$



Example

Step 3. Repeat Step 2 on every new state that is generated. Repeat until no new states are produced. This is shown in the table below.

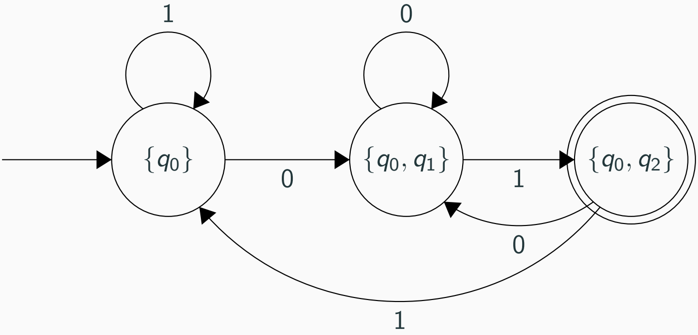


States S	$\bigcup_{p \in S} \delta(p, 0)$	$\bigcup_{p \in S} \delta(p, 1)$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$

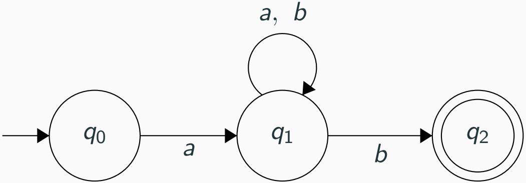
Example

Step 4. Build the DFA.

States S	$\delta'(S, 0)$	$\delta'(S, 1)$
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$



Exercise. Convert the following NFA N into an equivalent DFA M. Your conversion should ensure that $L(N) = L(M)$. $\Sigma = \{a, b\}$.



Theorem

Theorem. The set of all languages accepted by NFAs is the same as the set of all languages accepted by DFAs.
Why?

1. Every DFA is an NFA.
2. Every NFA can be converted into a DFA.

Corollary. A language is regular if and only if there is an FA (either a DFA or NFA) that accepts it.

Exercise

Exercise. Prove that the language $\{xwx : x \in \{0,1\}, w \in \{0,1\}^*\}$ is regular.