COSE215: Theory of Computation

Lecture 3-1 — Deterministic Finite Automata

Hakjoo Oh
2016 Spring

A Finite Automaton is a String Recognizer

- Deterministic Finite Automata (DFA)

- Nondeterministic Finite Automata (NFA)

Deterministic Finite Automata

Definition (DFA)

A *deterministic finite automaton* (or *DFA*):

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

where

- Q: a finite set of *states*
- Σ: a finite set of *input symbols* (or input alphabet)
- $\delta : Q \times \Sigma \to Q$: a total function called *transition function*
- $q_0 \in Q$: the *initial state*
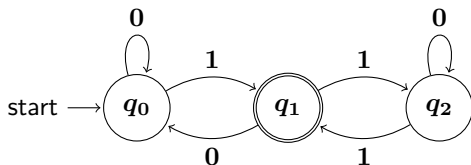- $F \subseteq Q$: a set of *final states*

Example

Definition:

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$$
$$\delta(q_0, 0) = q_0, \quad \delta(q_0, 1) = q_1$$
$$\delta(q_1, 0) = q_0, \quad \delta(q_1, 1) = q_2$$
$$\delta(q_2, 0) = q_2, \quad \delta(q_2, 1) = q_1$$

Transition graph:

Transition table:

	0	1
$\rightarrow q_0$	q_0	q_1
$*q_1$	q_0	q_2
q_2	q_2	q_1

Exercises

Design a DFA that accepts the language:

$$\{x01y \mid x \text{ and } y \text{ are any strings of } 0\text{'s and } 1\text{'s}\}$$

Extended Transition Function

Extend $\delta : Q \times \Sigma \to Q$ to input *strings*:

$$\delta^* : Q \times \Sigma^* \to Q$$

- (Basis) $s = \epsilon$:

$$\delta^*(q, \epsilon) = q$$

- (Induction) $s = wa$:

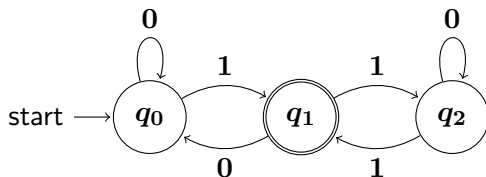$$\delta^*(q, wa) = \delta(\delta^*(q, w), a)$$

Example

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

$$\delta(q_0, 0) = q_0, \quad \delta(q_0, 1) = q_1$$
$$\delta(q_1, 0) = q_0, \quad \delta(q_1, 1) = q_2$$
$$\delta(q_2, 0) = q_2, \quad \delta(q_2, 1) = q_1$$

- $\delta^*(q_0, 011) =$

Language of Automata

Definition

A DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepts a string w if

$$\delta^*(q_0, w) \in F$$

and the *language* of automaton M, denoted $L(M)$, is defined as the set of all strings accepted by M:

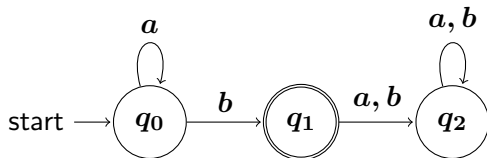$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}.$$

Definition

A language L is said to be *regular* iff there exists some DFA M such that

$$L = L(M)$$

Exercises

1. What is the language of the following automaton?

2. Design a DFA that accepts the language:

$$L(M) = \{w \in \{a, b\}^* \mid w = abu, u \in \{a, b\}^*\}$$

3. Design a DFA that accepts strings ending with 01:

$$L = \{w01 \mid w \in \{0, 1\}^*\}$$