Theoretical computer science

Tutorial - week 10

March 25, 2021

Innopolis University

Agenda

- Recap FSA
- Non-deterministic FSA

FSA (Formal definition)

A complete Finite State Automaton

A complete Finite State Automaton is a tuple $< Q, \Sigma, q_0, A, \delta >$, where

Q is a finite set of *states*;
Σ is a finite *input alphabet*;
$q_0 \in Q$ is the *initial* state;
$A \subseteq Q$ is the set of *accepting* states;
$\delta : Q \times \Sigma \rightarrow Q$ is a total *transition* function.

For any element q of Q and any symbol $\sigma \in \Sigma$, we interpret $\delta(q, \sigma)$ as the state to which the FSA moves, if it is in state q and receives the input σ.

The extended transition δ^*

A move sequence starts from an initial state and is *accepting* if it reaches one of the final states (informally explained with the previous example).
Formally, this transition is defined recursively:

the extended transition δ^*

Let $M =< Q, \Sigma, q_0, A, \delta >$ be a complete finite state automaton.
We define the extended transition function

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

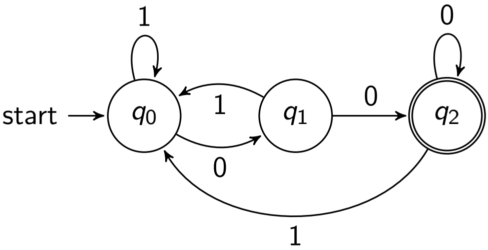
as follows:

1. For every $q \in Q$, $\delta^*(q, \epsilon) = q$
2. For every $q \in Q$, every $y \in \Sigma^*$, and every $\sigma \in \Sigma$,

$$\delta^*(q, y\sigma) = \delta(\delta^*(q, y), \sigma)$$

Example

The complete FSA M_1 accepting strings ending with 00



Non-deterministic FSA

Non-deterministic Finite State Automata (NDFSA)

Definition: NDFSA

A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e. set of all possible subsets)

Non-deterministic Finite State Automata (NDFSA)

Definition: NDFSA

A NDFSA is a tuple $\langle Q, \Sigma, q_0, A, \delta \rangle$, where Q, Σ, q_0, A are defined as in (D)FSA and the transition function is defined as

$$\delta : Q \times \Sigma \rightarrow \mathbb{P}(Q)$$

$\mathbb{P}$ is the powerset function (i.e. set of all possible subsets)

A NDFSA modifies the definition of a FSA to permit transitions at each stage to either zero, one, or more than one states.

The extended transition δ^* for NDFSA

the extended transition δ^* for NDFSA

Let $M = \langle Q, \Sigma, \delta, q_0, A \rangle$ be a NDFSA. We define the extended transition function as follows:

1. For every $q \in Q$, $\delta^*(q, \epsilon) = \{q\}$
2. For every $q \in Q$, every $y \in \Sigma^*$, and every $i \in \Sigma$,

$$\delta^*(q, yi) = \bigcup_{q' \in \delta^*(q,y)} \delta(q', i)$$

Acceptance by a NDFSA

Acceptance by a NDFSA

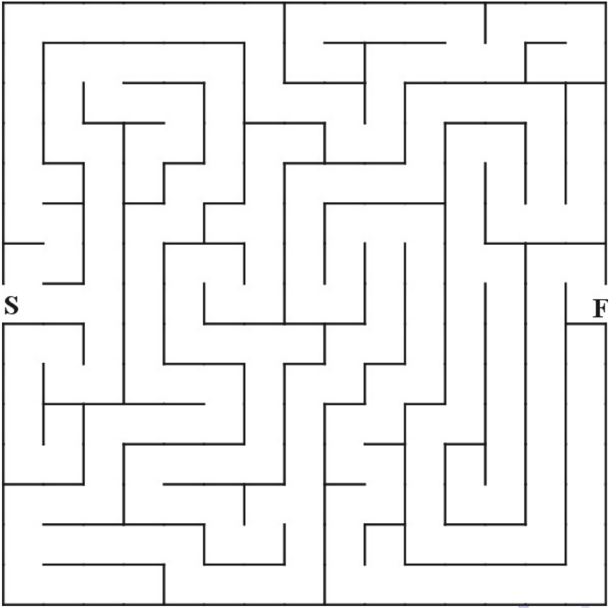
Let $M = \langle Q, \Sigma, q_0, A, \delta \rangle$ be a NDFSA, and let $x \in \Sigma^*$. The string x is accepted by M iff

$$\delta^*(q_0, x) \cap A \neq \varnothing$$

and it is rejected by M otherwise.

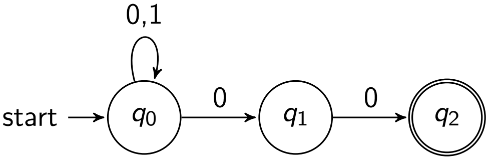
Notion: Among the various possible runs (with the same input) of the NDFSA, it is sufficient that one of them succeeds to accept the input string.

Maze analogy

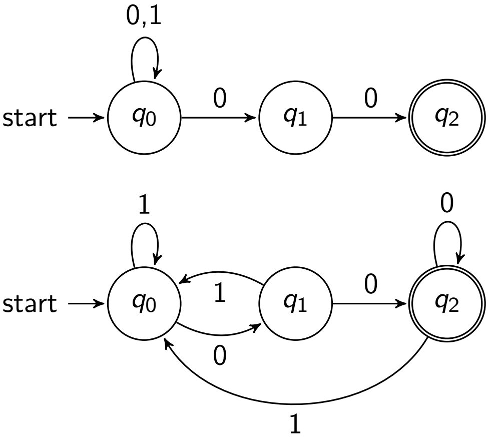


Example

The NDFSA M_2 accepting strings ending with 00

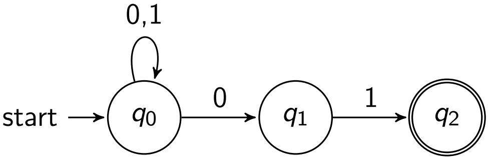


Example: M_1 and M_2

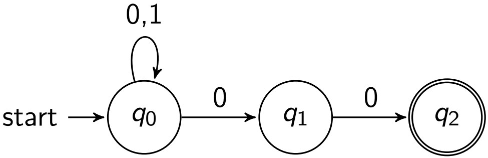


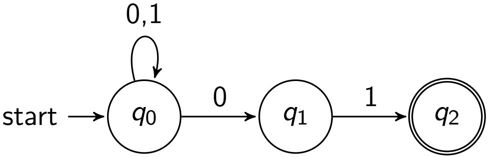
Example

The NDFSA M_3 accepting strings ending with 01



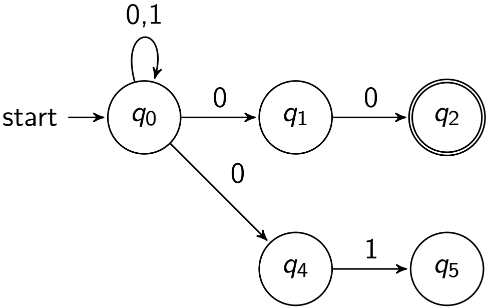
Example: M_1 and M_3





Example

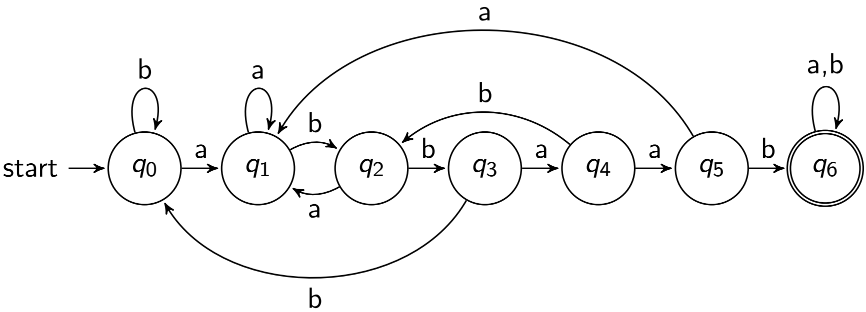
The NDFSA M_3 accepting strings ending with 01



Exercise

Let Σ be the alphabet $\Sigma = \{0, 1\}$

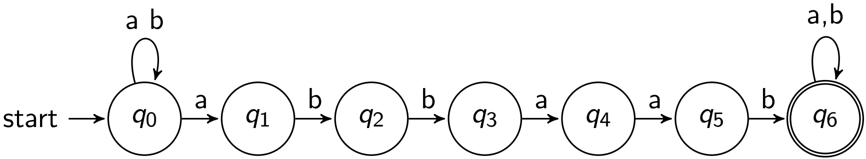
- $L = \{x \in \Sigma^* \mid \; x \text{ contains the substring } abbaab\}$;



Exercise

Let Σ be the alphabet $\Sigma = \{0, 1\}$

- $L = \{x \in \Sigma^* \mid \ x$ contains the substring $abbaab\}$;



Wrap up

- What have you learnt today?

Wrap up

- What have you learnt today?
- What for this could be useful?