Deterministic Finite Auomtaton

CENG 280

Deterministic Finite Automaton

- DFA
- DFA semanttics

An FSA

- has a finite number of states (fixed internal memory)
- has a reading head
- reads symbols from input tape

Deterministic Finite Automaton

An FSA

- has a finite number of states (fixed internal memory)
- has a reading head
- reads symbols from input tape

FSA working principle

a- starts from the leftmost position

  1- reads a symbol
  2- updates the internal state according to the input read and the current state,
  3- moves reading head one position to the right

b- repeats steps 1-3 until the end of the input string is reached

c- accepts or rejects the input

Deterministic Finite Automaton

Definition (Deterministic finite state automaton)

Deterministic finite state automaton is a quintuple $M = (K, \Sigma, \delta, s, F)$, where

- K is a finite set of states,
- Σ is an alphabet,
- $s \in K$ is the initial state,
- $F \subseteq K$ is the set of final states, and
- $\delta : K \times \Sigma \to K$ is the transition function.

Deterministic Finite Automaton

Definition (Deterministic finite state automaton)

Deterministic finite state automaton is a quintuple $M = (K, \Sigma, \delta, s, F)$, where

- K is a finite set of states,
- Σ is an alphabet,
- $s \in K$ is the initial state,
- $F \subseteq K$ is the set of final states, and
- $\delta : K \times \Sigma \to K$ is the transition function.

$\delta(q, \sigma)$ is the state that the machine transitions to when it reads σ at state q.

Definition (Deterministic finite state automaton)

Deterministic finite state automaton is a quintuple $M = (K, \Sigma, \delta, s, F)$, where

- K is a finite set of states,
- Σ is an alphabet,
- $s \in K$ is the initial state,
- $F \subseteq K$ is the set of final states, and
- $\delta : K \times \Sigma \to K$ is the transition function.

$\delta(q, \sigma)$ is the state that the machine transitions to when it reads σ at state q.

Example

Consider languages $L_1 = \{w \in \{a, b\}^\star \mid w$ has even number of b'$s\}$, $L_2 = \{w \in \{a, b\}^\star \mid w$ includes aa as a substring.$\}$.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.

$$M = (K, \Sigma, \delta, s, F)$$

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- The binary $\vdash_M$ (yields) relation holds between two configurations of M if and only if the machine can pass from one configuration to another one as a result of a single move.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- The binary $\vdash_M$ (yields) relation holds between two configurations of M if and only if the machine can pass from one configuration to another one as a result of a single move.
- Let (q, w) and (q', w') be two configurations of M. Then $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$.

$$M = (K, \Sigma, \delta, s, F)$$

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- The binary $\vdash_M$ (yields) relation holds between two configurations of M if and only if the machine can pass from one configuration to another one as a result of a single move.
- Let (q, w) and (q', w') be two configurations of M. Then $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

- The **configuration** of the machine is the current state and the unread part of the input string, i.e., a configuration is an element of $K \times \Sigma^\star$.
- The binary $\vdash_M$ (yields) relation holds between two configurations of M if and only if the machine can pass from one configuration to another one as a result of a single move.
- Let (q, w) and (q', w') be two configurations of M. Then $(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$.
- $(q, w) \vdash_M (q', w')$ reads (q, w) **yields** (q', w') **in one step**.
- Note that $\vdash_M$ is a function from $K \times \Sigma^+$ to $K \times \Sigma^\star$, hence, for every configuration except (q, e) there exists a uniquely determined next configuration.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.
- $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.
- $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if $(s, w) \vdash_M^\star (f, e)$ for some $f \in F$.

Deterministic Finite Automaton

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.
- $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if $(s, w) \vdash_M^\star (f, e)$ for some $f \in F$.
- The **language** of M, $L(M)$, is the set of strings accepted by M.

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.
- $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if $(s, w) \vdash_M^\star (f, e)$ for some $f \in F$.
- The **language** of M, $L(M)$, is the set of strings accepted by M.
- In tabular representation, the transition function is shown with a table.

$$M = (K, \Sigma, \delta, s, F)$$

$(q, w) \vdash_M (q', w')$ if and only if $w = aw'$ for some $a \in \Sigma$ and $q' = \delta(q, a)$

- $\vdash_M^\star$ is the **reflexive transitive closure** of $\vdash_M$.
- $(q, w) \vdash_M^\star (q', w')$ reads (q, w) **yields** (q', w').
- A string $w \in \Sigma^\star$ is **accepted** by M if and only if $(s, w) \vdash_M^\star (f, e)$ for some $f \in F$.
- The **language** of M, $L(M)$, is the set of strings accepted by M.
- In tabular representation, the transition function is shown with a table.
- In state diagram representation (a directed graph), the states are shown with nodes and the edges represent the transitions. There is a transition from node q to q' labeled with a if $q' = \delta(q, a)$.

Deterministic Finite Automaton - Examples

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid$ each a is immediately preceded by a $b\}$

Deterministic Finite Automaton - Examples

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid$ each a is immediately preceded by a $b\}$

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid w$ does not have aa as a substring $\}$

Deterministic Finite Automaton - Examples

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid$ each a is immediately preceded by a $b\}$

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid w$ does not have aa as a substring $\}$

Example

Construct a DFA M with $L(M) = \emptyset$.

Deterministic Finite Automaton - Examples

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid$ each a is immediately preceded by a $b\}$

Example

Construct a DFA accepting
$L = \{w \in \{a, b\}^\star \mid w$ does not have aa as a substring $\}$

Example

Construct a DFA M with $L(M) = \emptyset$.

Example

When $e \in L(M)$?

Deterministic Finite Automaton - Examples

Example

Construct M with $\Sigma = \{\}$ and $L(M) \neq \emptyset$

Deterministic Finite Automaton - Examples

Example

Construct M with $\Sigma = \{\}$ and $L(M) \neq \emptyset$

Example

$\Sigma = \{a, b\}$. $K = \{q_0, q_1\}$. How many different automata can one define?

Deterministic Finite Automaton - Examples

Example

Construct M with $\Sigma = \{\}$ and $L(M) \neq \emptyset$

Example

$\Sigma = \{a, b\}$. $K = \{q_0, q_1\}$. How many different automata can one define?

Example

Construct a DFA M such that
$L(M) = \{w \in \{a, b\}^{\star} \mid w\text{contains even number of substrings } ba\}$