Finite Automata

U.D. Computación

DSIC - UPV

Deterministic Finite Automaton (*DFA*)

Deterministic Finite Automaton (*DFA*)

A Deterministic Finite Automaton (*DFA*) is the following 5-tuple: $A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of states.
- Σ is a finite set of symbols called *alphabet*
- $\delta : Q \times \Sigma \to Q$ is a partial function called *transition function*.
- $q_0 \in Q$ is the *initial state*.
- $F \subseteq Q$ is the *set of final states*.

When the transition function is total the automaton is called *complete*.

Automata Representation:
Transition table

$|Q|$ rows and $|\Sigma|$ columns. (i, j) is the state $\delta(q_i, a_j)$ where q_i is the ith element of Q and a_j the jth element of Σ.

Example

Let $A = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_0, q_1\})$ where δ is defined as follows:

$$\begin{array}{lll} \delta(q_0, a) = q_0 & \delta(q_1, a) = q_2 & \delta(q_2, a) = q_2 \\ \delta(q_0, b) = q_1 & \delta(q_1, b) = q_1 & \delta(q_2, b) = q_2 \end{array}$$

Its transition table is:

	a	**b**
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

Automata Representation:
Transition diagram

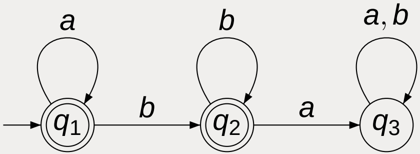
Is a labeled directed graph, where:

- Its number of nodes (vertices) is $|Q|$. Every node corresponds to a state.
- $\forall q_i, q_j \in Q, \forall a_k \in \Sigma$, if $\delta(q_i, a_k) = q_j$ there is an arc (arrow) that goes from state q_i to state q_j labeled with a_k.
- The initial state has an input arrow coming from nowhere.
- The final states are drawn with a double circle.

Automata representation

Example

The transition diagram of the previous example is:



Extension of δ to strings

The function $\hat{\delta} : Q \times \Sigma^* \to Q$ is defined as follows:
$\forall q \in Q, x \in \Sigma^*, a \in \Sigma$

- $\hat{\delta}(q, \lambda) = q$
- $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

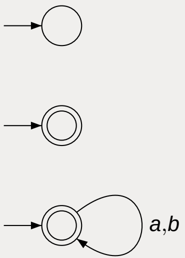
As $\hat{\delta}(q, a) = \hat{\delta}(q, \lambda a) = \delta(\hat{\delta}(q, \lambda), a) = \delta(q, a)$, in the sequel, we will write δ instead of $\hat{\delta}$.

Language acepted by a *DFA*

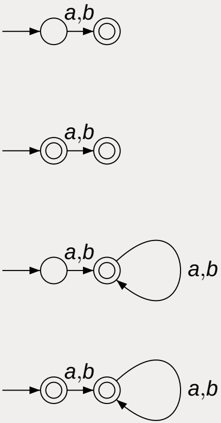
- Let $A = (Q, \Sigma, \delta, q_0, F)$ be a *DFA*, and let $x \in \Sigma^*$. A string x is accepted by A when $\delta(q_0, x) \in F$.
- The *language accepted* by the *DFA* A is defined as:

$$L(A) = \{x \in \Sigma^* | \delta(q_0, x) \in F\}$$

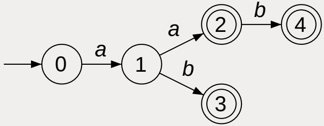
Examples of *DFA*

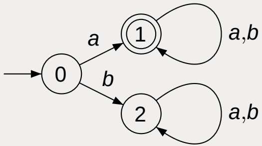


Examples of *DFA*

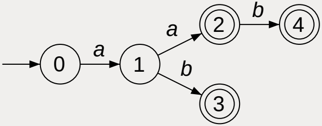


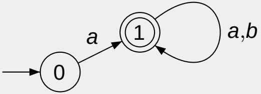
Examples of *DFA*



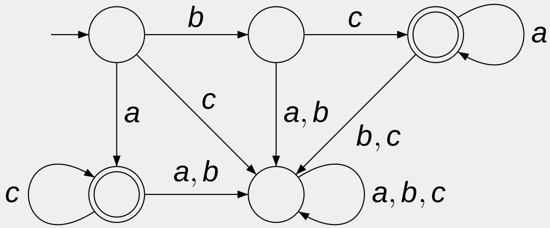


Examples of *DFA*

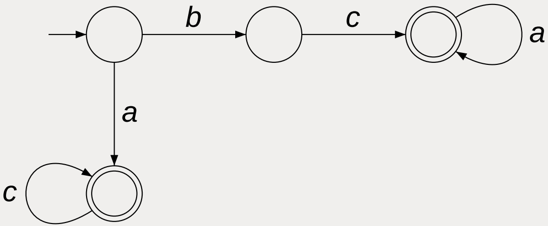




Examples of *DFA*



Examples of *DFA*



Non Deterministic Finite Automata (*NFA*)

Non Deterministic Finite Automata (*NFA*)

A Non Deterministic Finite Automata (*NFA*) is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q, Σ, $q_0 \in Q$, y $F \subseteq Q$ are the same as in the definition of $DFA's$
- The transition function is now the partial function $\delta : Q \times \Sigma \to 2^Q$.

NFA Representation

Example

Let $A = (\{q_0, q_1, q_2\}, \{a, b, c\}, \delta, q_0, \{q_0, q_1, q_2\})$ where the transition function is defined as:

$$
\begin{array}{lll}
\delta(q_0, a) = \{q_0, q_1, q_2\} & \delta(q_1, a) = \emptyset & \delta(q_2, a) = \emptyset \\
\delta(q_0, b) = \{q_1, q_2\} & \delta(q_1, b) = \{q_1, q_2\} & \delta(q_2, b) = \emptyset \\
\delta(q_0, c) = \{q_2\} & \delta(q_1, c) = \{q_2\} & \delta(q_2, c) = \{q_2\}
\end{array}
$$

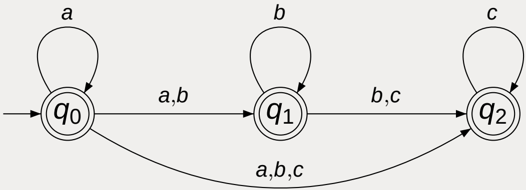
Its transition table is:

	a	**b**	**c**
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$	$\{q_2\}$

NFA Representation

Example

Transition diagram for this automaton:



Extension of δ to strings

The function $\hat{\delta} : Q \times \Sigma^* \rightarrow 2^Q$ is defined as follows:
$\forall q \in Q, x \in \Sigma^*, a \in \Sigma$:

- $\hat{\delta}(q, \lambda) = \{q\}$
- $\hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)$

As $\hat{\delta}(q, a) = \bigcup_{p \in \hat{\delta}(q,\lambda)} \delta(p, a) = \bigcup_{p \in \{q\}} \delta(p, a) = \delta(q, a)$, in the sequel we will use δ instead of $\hat{\delta}$.

Language accepted by a *NFA*

Let $A = (Q, \Sigma, \delta, q_0, F)$ a *NFA*, the*language accepted* by the NFA A is defined as

$$L(A) = \{x \in \Sigma^* | \delta(q_0, x) \cap F \neq \emptyset\}$$

Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:

$\longrightarrow$ 1	
$\longrightarrow$ 2	
$\longrightarrow$ 3	

Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:

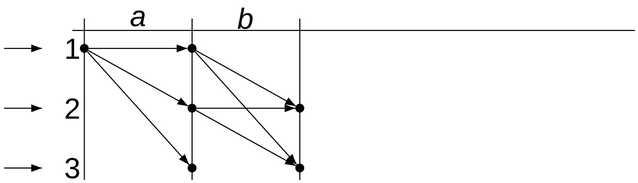
		a	
→	1 •		
→	2		
→	3		

Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:

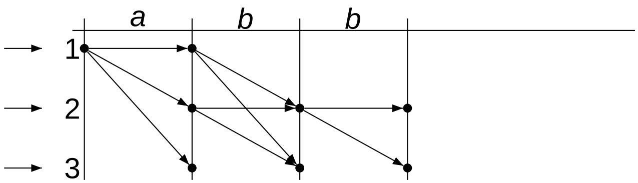
Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:



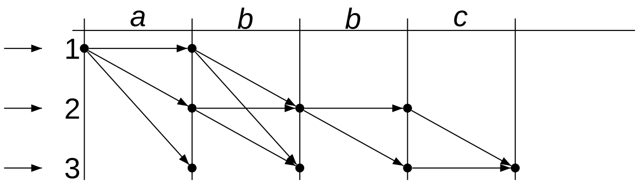
Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:



Non deterministic behavior

Analysis the word *abbc* ($\delta(q_0, abbc)$) by the previous *NFA*:



Equivalence *DFA* – *NFA*

- Every *DFA* is a particular instance of a *NFA*.
- The way to obtain an equivalent *DFA* from a *NFA* is based in the fact that the power set (set of subsets) of a finite set is also finte and thus:
  - The states of the *DFA* are subsets of the set of states of the *NFA*.
  - The transition function of a symbol acts between set of states.

Equivalence *DFA* – *NFA*

- Extension of the transition function to set of states, $\delta' : 2^Q \times \Sigma \to 2^Q$:
  $\forall P \subseteq Q \quad \delta'(P, a) = \bigcup_{p \in P} \delta(p, a)$
- Extension of the transition function to set of states and strings, $\delta'' : 2^Q \times \Sigma^* \to 2^Q$:
  - $\forall P \subseteq Q \quad \delta''(P, \lambda) = P$
  - $\forall P \subseteq Q, x \in \Sigma^*, a \in \Sigma \quad \delta''(P, xa) = \delta'(\delta''(P, x), a)$

Equivalence *DFA* – *NFA*

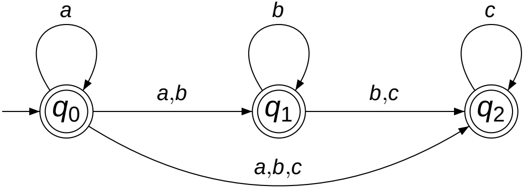
Equivalence *DFA* – *NFA*

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a *NFA* such that $L = L(A)$.
We define a *DFA* $A' = (Q', \Sigma, \delta', q'_0, F')$ as:

- $Q' = 2^Q, q'_0 = \{q_0\}$,
- $F' = \{q' \in Q' | q' \cap F \neq \emptyset\}$,
- The transition function δ' is defined as the extension of the function δ to set of states.

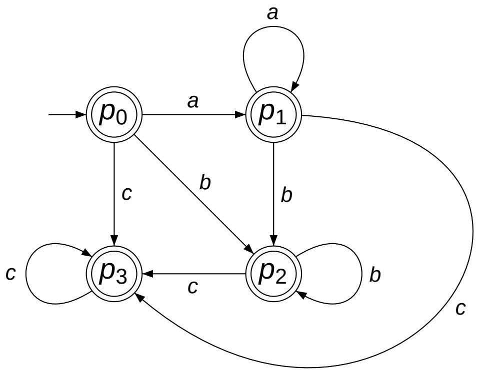
- The automaton A' so defined is a *DFA*, as its transition function is $\delta' : Q' \times \Sigma \rightarrow Q'$.
- It can be proved that $L(A') = L$.

Example



	a	b	c
$\{q_0\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_1, q_2\}$	$\emptyset$	$\{q_1, q_2\}$	$\{q_2\}$
$\{q_2\}$	$\emptyset$	$\emptyset$	$\{q_2\}$

Example



Finite Automata with empty moves (λ-FA)

Finite Automata with empty moves (λ-FA)

A Finite Automata with empty moves ($FA\lambda$) is a 5-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where:

- Q, Σ, $q_0 \in Q$, y $F \subseteq Q$ are the same as in the definition of *DFAs*.
- $\delta : Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$ is the *transition function*, (also a partial function).

Representation

Ejemplo

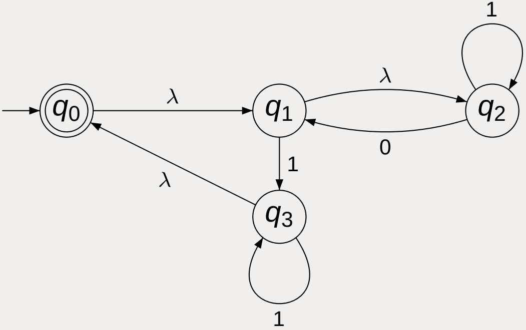
Let $A = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\})$ where the transition function is:

	0	**1**	λ
q_0	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\emptyset$	$\{q_3\}$	$\{q_2\}$
q_2	$\{q_1\}$	$\{q_2\}$	$\emptyset$
q_3	$\emptyset$	$\{q_3\}$	$\{q_0\}$

Representation

Example

Transition diagram:



Extension of the transition function to strings

λ-closure

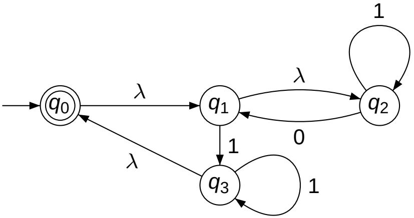
- Let $q \in Q, \lambda - clausure(q) = \{q\} \cup \{q' | q'$ can be reached from q through ways labeled with $\lambda\}$.
- Let $P \subseteq Q, \lambda - closure(P) = \bigcup_{p \in P} \lambda - closure(p)$.

Extension to strings

$\forall q \in Q, x \in \Sigma^*, a \in \Sigma$:

- $\hat{\delta}(q, \lambda) = \lambda - closure(q)$
- $\hat{\delta}(q, xa) = \lambda - closure\left(\bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)\right)$

Example



$\lambda - closure$

- $\lambda - closure(q_0) = \{q_0, q_1, q_2\}$
- $\lambda - closure(q_1) = \{q_1, q_2\}$
- $\lambda - closure(q_2) = \{q_2\}$
- $\lambda - closure(q_3) = \{q_0, q_1, q_2, q_3\}$

Example

Calculus of $\hat{\delta}(q_0, 01)$

$\hat{\delta}(q_0, 01) = \lambda - closure(\bigcup_{p \in \hat{\delta}(q_0, 0)} \delta(p, 1))$ (1)

$\hat{\delta}(q_0, 0) = \lambda - closure(\bigcup_{p \in \hat{\delta}(q_0, \lambda)} \delta(p, 0))$ (2)

$\hat{\delta}(q_0, \lambda) = \lambda - closure(q_0) = \{q_0, q_1, q_2\}$.

Substituting in (2):

$\hat{\delta}(q_0, 0) = \lambda - closure(\bigcup_{p \in \{q_0, q_1, q_2\}} \delta(p, 0)) =$
$\lambda - closure(\emptyset \cup \emptyset \cup \{q_1\}) = \lambda - closure(\{q_1\}) = \{q_1, q_2\}$.

Substituting in (1):

$\hat{\delta}(q_0, 01) = \lambda - closure(\bigcup_{p \in \{q_1, q_2\}} \delta(p, 1)) =$
$\lambda - closure(\{q_2\} \cup \{q_3\}) = \lambda - closure(\{q_2, q_3\}) =$
$\{q_0, q_1, q_2, q_3\}$.

In a λ-FA, $\hat{\delta}(q, a)$ and $\delta(q, a)$ are not necessarily equal and the same happens to $\hat{\delta}(q, \lambda)$ and $\delta(q, \lambda)$.
we have to distinguish between δ and $\hat{\delta}$.

Language accepted by a λ-FA

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a *FA*λ, we define the *Language accepted* by the *FA*λ A as

$$L(A) = \{x \in \Sigma^* | \hat{\delta}(q_0, x) \cap F \neq \emptyset\}.$$

Equivalence $NFA - \lambda$-FA

Every *NFA* can be seen as a $FA\lambda$ in which
$\forall q \in Q \quad \lambda - closure(q) = \{q\}$.

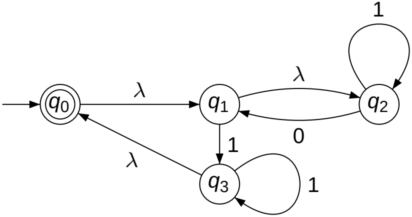
Equivalence *NFA* - $FA\lambda$

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a $AF\lambda$. We define a *NFA* $A' = (Q, \Sigma, \delta', q_0, F')$ where:

$$F' = \begin{cases} F \cup \{q_0\} & \text{if} \quad \lambda - closure(q_0) \cap F \neq \emptyset \\ F & \text{otherwise} \end{cases}$$

The function δ' is defined as:
$\forall q \in Q, a \in \Sigma \quad \delta'(q, a) = \hat{\delta}(q, a)$.

Example



λ – *closures*

- $\lambda - closure(q_0) = \{q_0, q_1, q_2\}$
- $\lambda - closure(q_1) = \{q_1, q_2\}$
- $\lambda - closure(q_2) = \{q_2\}$
- $\lambda - closure(q_3) = \{q_0, q_1, q_2, q_3\}$

Example

	0	**1**
q_0	$\{q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
q_1	$\{q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
q_2	$\{q_1, q_2\}$	$\{q_2\}$
q_3	$\{q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$

The set of final states $F = \{q_0\}$ contains the initial state, and thus $F' = F$.