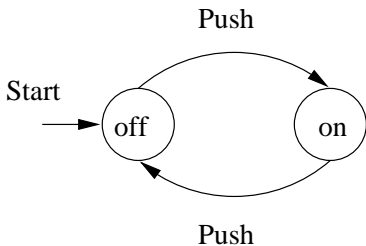


# Automati a stati finiti

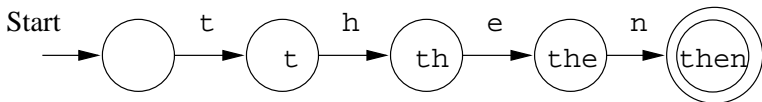
- Introduzione agli automi a stati finiti
- Concetti di base della teoria degli automi
- Tecniche di dimostrazione
- Automi a stati finiti deterministici (DFA)
- Automi a stati finiti non deterministici (NFA)
- Automi a stati finiti non deterministici con  $\epsilon$ -transizione ( $\epsilon$ -NFA)

- **Automa:** Dispositivo capace di eseguire una sequenza di azioni/operazioni in modo automatico
- **Automa a stati finiti:** Automa con una memoria finita
- **Automa stati finiti come modello per**
  - Software per la progettazione di circuiti digitali.
  - Analizzatori lessicali di un compilatore.
  - Ricerca di parole chiave in un file o sul web.
  - Software per verificare i protocolli di comunicazione.

- Automa a stati finiti per un interruttore on/off



- Automa a stati finiti che riconosce la stringa then



**Grafo:** la piu' semplice rappresentazione

- **Nodi** = stati
- **Archi** rappresentano transizioni
- **Etichette** sugli archi ci dicono cosa causa la transizione

- **Alfabeto:** Insieme finito e non vuoto di simboli
  - Esempio:  $\Sigma = \{0, 1\}$  alfabeto binario
  - Esempio:  $\Sigma = \{a, b, c, \dots, z\}$  insieme di tutte le lettere minuscole
  - Esempio: Insieme di tutti i caratteri ASCII
- **Stringa:** Sequenza finita di simboli presi da un alfabeto  $\Sigma$ 
  - Esempio: 0011001 (stringa su  $\Sigma = \{0, 1\}$ )
- **Stringa vuota:** La stringa con zero occorrenze di simboli da  $\Sigma$ 
  - La stringa vuota e' denotata con  $\epsilon$

- **Lunghezza di una stringa:** Numero di posizioni per i simboli nella stringa

$|w|$  denota la lunghezza della stringa  $w$

$$|0110| = 4, |\epsilon| = 0$$

- **Potenze di un alfabeto:**  $\Sigma^k$  = insieme delle stringhe di lunghezza  $k$  con simboli da  $\Sigma$

Example:  $\Sigma = \{0, 1\}$

$$\Sigma^1 = \{0, 1\}$$

$$\Sigma^2 = \{00, 01, 10, 11\}$$

$$\Sigma^0 = \{\epsilon\}$$

**Domanda:** Quante stringhe ci sono in  $\Sigma^3$ ?

- L'insieme di tutte le stringhe su  $\Sigma$  e' denotato da  $\Sigma^*$

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$$

Anche:

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

$$\Sigma^* = \Sigma^+ \cup \{\epsilon\}$$

- **Concatenazione:** Se  $x$  e  $y$  sono stringhe, allora  $xy$  e' la stringa ottenuta mettendo una copia di  $y$  immediatamente dopo una copia di  $x$

$$x = a_1 a_2 \dots a_i, y = b_1 b_2 \dots b_j$$

$$xy = a_1 a_2 \dots a_i b_1 b_2 \dots b_j$$

Esempio:  $x = 01101$ ,  $y = 110$ ,  $xy = 01101110$

**Nota:** Per ogni stringa  $x$

$$x\epsilon = \epsilon x = x$$



**Linguaggio:** insieme di stringhe scelte da  $\Sigma^*$ , dove  $\Sigma$  e' un alfabeto.  $L \subseteq \Sigma^*$  e' un linguaggio.

## Esempi di linguaggi:

- L'insieme delle parole italiane legali
- L'insieme dei programmi C legali
- L'insieme delle stringhe che consistono di  $n$  zeri seguiti da  $n$  uni, con  $n \geq 0$

$$\{\epsilon, 01, 0011, 000111, \dots\}$$

- L'insieme delle stringhe con un numero uguale di zeri e di uni

$$\{\epsilon, 01, 10, 0011, 0101, 1001, \dots\}$$

- L'insieme dei numeri binari il cui valore e' primo

$$L_P = \{10, 11, 101, 111, 1011, \dots\}$$

- Il linguaggio vuoto  $\emptyset$  che non contiene nessuna stringa
- Il linguaggio  $\{\epsilon\}$  consiste della stringa vuota

**Nota:**  $\emptyset \neq \{\epsilon\}$

**Nota:** L'alfabeto  $\Sigma$  e' sempre finito

**Problema:** Sia  $\Sigma$  un alfabeto,  $w \in \Sigma^*$  e  $L$  un linguaggio,  $w \in L$ ?  
La stringa  $w$  e' un elemento di un linguaggio  $L$ ?

- Esempio: Dato un numero binario, e' primo = e' un elemento di  $L_P$ ?

- Sequenza di enunciati la cui verità porta da un enunciato iniziale (l'ipotesi) ad un enunciato finale (la conclusione)
- Forma del teorema: **Se H, allora C** ( $H \Rightarrow C$ , H implica C)
- H= ipotesi, C= conclusione
  
- Teorema: se  $x \geq 4$ , allora  $2^x \geq x^2$
- x parametro quantificato universalmente (vale per tutti gli x)

- Teoremi della forma "C1 se e solo se C2" : due direzioni di prova. (Si deve provare che:  $C_1 \Rightarrow C_2$  e  $C_2 \Rightarrow C_1$ )
- Dimostrazione per assurdo di teoremi della forma  $H \Rightarrow C$   
H e non C implica il falso
- **Controesempio**: per dimostrare che un teorema non vale basta provare che in un caso non vale
  - Esercizio: E' vero che se x e' un numero primo allora x e' dispari? No, infatti 2 e' numero primo ma non e' dispari

- Utili quando ci sono cose definite ricorsivamente
- Esempio:  
0 e' un intero e  
se  $n$  e' un intero allora  $n+1$  e' un intero
- **Induzione sugli interi:** dobbiamo dimostrare un enunciato  $S(n)$  su un intero  $n$ 
  - **Base:** dimostriamo  $S(i)$  per un intero particolare  $i$  (0 o 1 di solito)
  - **Passo induttivo:** si suppone  $n \geq i$ , si dimostra che se vale  $S(n)$  allora vale  $S(n+1)$

Possiamo concludere che  $S(n)$  e' vero per ogni  $n \geq i$

- Se  $x \geq 4$ , allora  $2^x \geq x^2$
- **Base:**  $x = 4 \Rightarrow 2^x = 2^4 = 16$  e  $x^2 = 4^2 = 16$

## Induzione:

Si suppone che per  $x \geq 4$ ,  $2^x \geq x^2$  (ipotesi induttiva)

Dobbiamo dimostrare che  $2^{x+1} \geq (x+1)^2$

Abbiamo:

- $2^{x+1} = 2^x \times 2 \geq x^2 \times 2 = 2x^2$  (per ipotesi induttiva)
- Dimostriamo adesso che  $2x^2 \geq (x+1)^2 = x^2 + 2x + 1$
- Cioè dimostriamo che  $x^2 \geq 2x + 1$
- Cioè, semplificando (divido per  $x$ ), dimostriamo che  $x \geq 2 + 1/x$
- Se  $x \geq 4$ , allora  $1/x \leq 1/4$  e quindi  $2 + 1/x \leq 2.25$
- Quindi se  $x \geq 4$ , allora  $x \geq 2 + 1/x$

Abbiamo dimostrato che se  $x \geq 4$ , allora  $2^x \geq x^2$

- Molte strutture possono essere definite ricorsivamente
- **Espressioni:**
  - **caso base:** qualunque **numero o lettera** e' un'espressione
  - **caso induttivo:**  
se  $E$  e  $F$  sono espressioni,  
allora lo sono anche  $E + F$ ,  $E \times F$  e  $(E)$
  - Esempi:  $3+(4 \times 2)$ ,  $(2 \times (5+7)) \times 4$
- **Per dimostrare teoremi su una struttura  $X$ :**
  - si dimostra l'enunciato sul caso base della def. di  $X$ , poi
  - si suppone che l'enunciato sia vero  
sulle strutture di cui  $X$  e' composta secondo la definizione ricorsiva
  - si dimostra l'enunciato sulla struttura  $X$



- Dimostrare con induzione strutturale che ogni espressione ha un numero uguale di parentesi aperte e chiuse
  - Caso base: un carattere ha zero parentesi aperte e zero parentesi chiuse  $\Rightarrow$  vero
  - Induzione: Suppongo che  $E$  e  $F$  hanno un numero uguale di parentesi aperte e chiuse.  
Devo dimostrare che  $E + F$ ,  $E \times F$  e  $(E)$  hanno un numero uguale di parentesi aperte e chiuse
  - Per  $E + F$  e  $E \times F$ : se vale l'enunciato per  $E$  e  $F$ , supponiamo che  $E$  abbia  $n$  parentesi aperte e chiuse e  $F$  ne abbia  $m$   
 $\Rightarrow E + F$  ne ha  $n + m$  parentesi aperte e  $n + m$  parentesi chiuse
  - Per  $(E)$ : se vale per  $E$ , supponiamo che  $E$  abbia  $n$  parentesi aperte e  $n$  chiuse  
 $\Rightarrow (E)$  ha  $n + 1$  parentesi aperte e  $n + 1$  parentesi chiuse

- Non e' sempre possibile dimostrare un singolo enunciato per induzione
- Si devono dimostrare congiuntamente un gruppo di enunciati  $S_1(n), S_2(n), \dots, S_k(n)$  per induzione su  $n$ .

- Introduzione agli automi a stati finiti
- Concetti di base della teoria degli automi
- Tecniche di dimostrazione
- **Automi a stati finiti deterministici (DFA)**
- Automi a stati finiti non deterministici (NFA)
- Automi a stati finiti non deterministici con  $\epsilon$ -transizione ( $\epsilon$ -NFA)

Un **DFA** e' una quintupla

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

- $Q$  e' un insieme finito di *stati*
- $\Sigma$  e' un *alfabeto finito* (= simboli di input)
- $\delta$  e' una *funzione di transizione*:  $Q \times \Sigma \rightarrow Q, (q, a) \mapsto p$
- $q_0 \in Q$  e' lo *stato iniziale*
- $F \subseteq Q$  e' un insieme di *stati finali* (o stati accettanti)

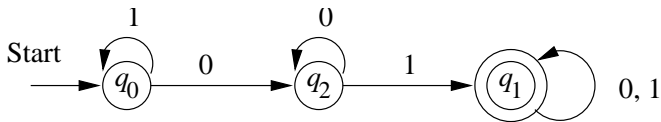
Esempio: Un automa  $A$  che accetta

$$L = \{x01y : x, y \in \{0, 1\}^*\}$$

L'automato  $A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$  come una *tabella di transizione*:

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

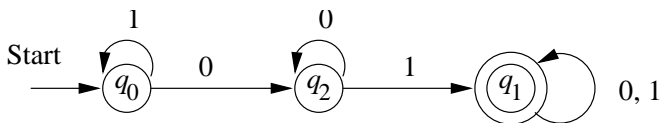
L'automato come un *diagramma di transizione*:



Un automa a stati finiti (FA) accetta una stringa  $w = a_1 a_2 \cdots a_n$  se esiste un cammino nel diagramma di transizione che

- 1 Inizia nello stato iniziale
- 2 Finisce in uno stato finale (di accettazione)
- 3 Ha una sequenza di etichette  $a_1 a_2 \cdots a_n$

Esercizio: Il DFA



accetta la stringa 01101?

- La **funzione di transizione**  $\delta$  puo' essere **estesa** a  $\hat{\delta}$  che opera su stati e stringhe (invece che su stati e simboli)

**Base:**  $\hat{\delta}(q, \epsilon) = q$

**Induzione:**  $\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

- Formalmente, il *linguaggio accettato da A* e'

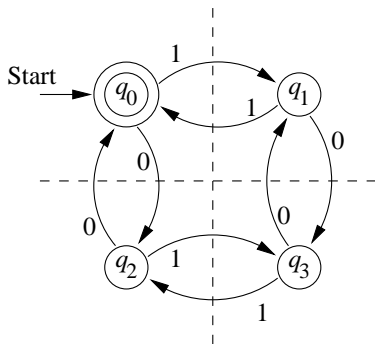
$$L(A) = \{w \in \Sigma^* : \hat{\delta}(q_0, w) \in F\}$$

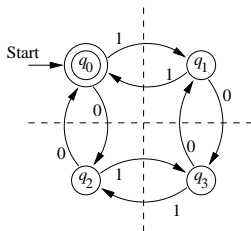
- I linguaggi accettati da automi a stati finiti sono detti *linguaggi regolari*

Costruire un DFA  $A$  che accetta tutte e sole le stringhe con un numero pari di zeri e un numero pari di uni



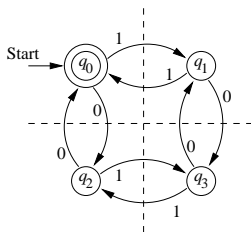
Costruire un DFA  $A$  che accetta tutte e sole le stringhe con un numero pari di zeri e un numero pari di uni





Rappresentazione tabulare dell'automa

	0	1
* → $q_0$	$q_2$	$q_1$
$q_1$	$q_3$	$q_0$
$q_2$	$q_0$	$q_3$
$q_3$	$q_1$	$q_2$



$w = 0101 \in L(A)$ ? Sì, poichè  $\hat{\delta}(q_0, 0101)$  è uno stato accettante.

$$\hat{\delta}(q_0, \epsilon) = q_0$$

$$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(q_0, 0) = q_2$$

$$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_2, 1) = q_3$$

$$\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0) = \delta(q_3, 0) = q_1$$

$$\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1) = \delta(q_1, 1) = q_0$$

- DFA A per i seguenti linguaggi sull'alfabeto  $\{0, 1\}$ :
  - ① Insieme di tutte le stringhe che finiscono con 00
  - ② Insieme di tutte le stringhe con tre zeri consecutivi
  - ③ Insieme delle stringhe con 011 come sottostringa
  - ④ Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01

- **Ex.5 - Primo Compitino 2013**

Sia  $L = \{w \in \{0,1\}^* \text{ tale che } w \text{ ha esattamente } \underline{\text{una sola}} \text{ occorrenza della sottostringa } 00\}$

- 1 Disegnare il diagramma di transizione di un DFA che riconosce  $L$ .
  - 2 Caratterizzare gli stati con una breve descrizione a parole (cioè per ogni stato descrivere le stringhe che portano ad esso).
  - 3 Assumendo di aver già dimostrato che gli stati non accettanti sono caratterizzati dalle stringhe descritte al punto precedente, dimostrare per mutua induzione che:
    - (1) se  $w \in L$  allora  $w \in L(A)$
    - (2) se  $w \in L(A)$  allora  $w \in L$ .
- **Esercizi per casa:** tutti gli esercizi delle dispense sui DFA

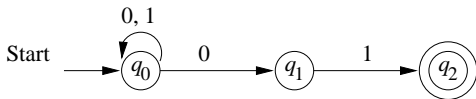
- Introduzione agli automi a stati finiti
- Concetti di base della teoria degli automi
- Tecniche di dimostrazione
- Automi a stati finiti deterministici (DFA): esercizi
- Automi a stati finiti non deterministici (NFA): definizioni
- Automi a stati finiti non deterministici con  $\epsilon$ -transizione ( $\epsilon$ -NFA)

- Costruire un DFA per il seguente linguaggio sull'alfabeto  $\{0, 1\}$ :  
Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01
- Definire il linguaggio accettato dai DFA descritti alla lavagna sull'alfabeto  $\{a, b\}$
- Sia  $L = \{w \in \{0, 1\}^* \mid w \text{ ha esattamente una sola occorrenza della sottostringa } 00\}$ 
  - Assumendo di aver già dimostrato che gli stati non accettanti sono caratterizzati dalle stringhe descritte al punto precedente, dimostrare per mutua induzione che:
    - (1) se  $w \in L$  allora  $w \in L(A)$
    - (2) se  $w \in L(A)$  allora  $w \in L$ .

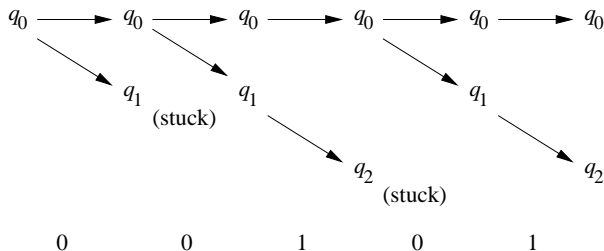
# Automi a stati finiti non deterministici (NFA)

Un NFA puo' essere in vari stati nello stesso momento

**Esempio:** un automa che accetta tutte e solo le stringhe che finiscono in 01.



Ecco cosa succede quando l'automata elabora l'input 00101



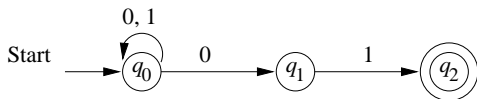


Formalmente, un NFA e' una quintupla

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

- $Q$  e' un insieme finito di stati
- $\Sigma$  e' un alfabeto finito
- $\delta$  e' una funzione di transizione  $Q \times \Sigma \rightarrow 2^Q$  (cioe' l'insieme dei sottoinsiemi di  $Q$ ),  $(q, a) \mapsto \{p_1, p_2, \dots, p_n\}$ ,  $n \geq 1$
- $q_0 \in Q$  e' lo *stato iniziale*
- $F \subseteq Q$  e' un insieme di *stati finali*

L'NFA appena visto



e' definito da

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

dove  $\delta$  e' la funzione di transizione

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
$q_1$	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

Funzione di **transizione estesa**  $\hat{\delta}$ .

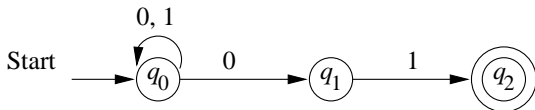
**Base:**  $\hat{\delta}(q, \epsilon) = \{q\}$

**Induzione:**

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

Il *linguaggio accettato da A* e'

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



La stringa 001 appartiene a  $L(A)$ ?

$$\hat{\delta}(q_0, \epsilon) = \{q_0\}$$

$$\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$$

$$\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$$

$$\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\} \text{ contiene } q_2 \text{ accettante quindi } 001 \in L(A)$$

- Introduzione agli automi a stati finiti
- Concetti di base della teoria degli automi
- Tecniche di dimostrazione
- Automi a stati finiti deterministici (DFA)
- Automi a stati finiti non deterministici (NFA): teoremi
- Automi a stati finiti non deterministici con  $\epsilon$ -transizione ( $\epsilon$ -NFA)

- Gli NFA sono di solito piu' facili da "programmare".
- Per ogni NFA  $N$  c'e' un DFA  $D$ , tale che  $L(D) = L(N)$ , e viceversa.

Per dimostrarlo si usa una **costruzione a sottoinsiemi**.

Dato un NFA

$$N = (Q_N, \Sigma, \delta_N, q_0, F_N)$$

costruiremo un DFA

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

tali che

$$L(D) = L(N)$$

I dettagli della costruzione a sottoinsiemi:

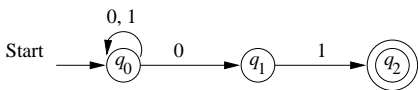
- $Q_D = \{S : S \subseteq Q_N\}$ .

Nota:  $|Q_D| = 2^{|Q_N|}$ , anche se la maggior parte degli stati in  $Q_D$  sono "garbage", cioè non raggiungibili dallo stato iniziale.

- $F_D = \{S \subseteq Q_N : S \cap F_N \neq \emptyset\}$

- Per ogni  $S \subseteq Q_N$  e  $a \in \Sigma$ ,

$$\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$$



Costruiamo  $\delta_D$  dall'NFA sopra:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\star\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\star\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\star\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$\star\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

In particolare:

$$\delta_D(\{q_0\}, 0) = \delta_N(q_0, 0) = \{q_0, q_1\}$$

$$\delta_D(\{q_0\}, 1) = \delta_N(q_0, 1) = \{q_0\}$$

$$\delta_D(\{q_0, q_1\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$$

$$\delta_D(\{q_0, q_1\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$$



**Nota:** Gli stati di  $D$  corrispondono a sottoinsiemi di stati di  $N$ , ma potevamo denotare gli stati di  $D$  in un altro modo, per esempio  $A, \dots, F$ .

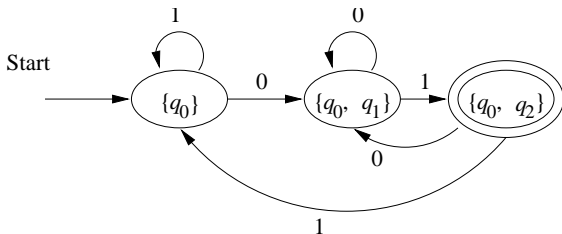
	0	1
$A$	$A$	$A$
$\rightarrow B$	$E$	$B$
$C$	$A$	$D$
$\star D$	$A$	$A$
$E$	$E$	$F$
$\star F$	$E$	$B$
$\star G$	$A$	$D$
$\star H$	$E$	$F$

Possiamo spesso evitare la crescita esponenziale degli stati costruendo la tabella di transizione per  $D$  **solo per stati accessibili**  $S$  come segue:

**Base:**  $S = \{q_0\}$  e' accessibile in  $D$

**Induzione:** Se lo stato  $S$  e' accessibile, anche gli stati in  $\bigcup_{a \in \Sigma} \delta_D(S, a)$  sono accessibili.

Esempio: Il DFA con stati accessibili solamente.



**Teorema 2.11:** Sia  $D$  il DFA ottenuto da un NFA  $N$  con la costruzione a sottoinsiemi. Allora  $L(D) = L(N)$ .

**Prova:**

$$L(N) = \{w \in \Sigma^* \mid \hat{\delta}_N(q_0, w) \cap F_N \neq \emptyset\}$$

$$L(D) = \{w \in \Sigma^* \mid \hat{\delta}_D(\{q_0\}, w) \in F_D\} = \{w \in \Sigma^* \mid \hat{\delta}_D(\{q_0\}, w) \cap F_N \neq \emptyset\}$$

Per provare  $L(D) = L(N)$  basta provare che,  $\forall w$ ,

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w) \quad (1)$$

Mostriamo l'enunciato (1) per induzione su  $|w|$

**Base:**  $w = \epsilon$ .

$$\hat{\delta}_D(\{q_0\}, \epsilon) = \{q_0\} \text{ per definizione di } \hat{\delta}_D$$

$$\hat{\delta}_N(q_0, \epsilon) = \{q_0\} \text{ per definizione di } \hat{\delta}_N$$

**Ip. Induttiva:** L'enunciato (1) vale  $\forall w$  con  $|w| \leq n$

**Passo Induttivo:** Devo dimostrare che l'enunciato (1) vale per  $w$  con  $|w| = n + 1$ . Sia  $w = xa$ .

$$\hat{\delta}_D(\{q_0\}, xa) \stackrel{\text{def}}{=} \delta_D(\hat{\delta}_D(\{q_0\}, x), a)$$

$$\stackrel{\text{i.h.}}{=} \delta_D(\hat{\delta}_N(q_0, x), a)$$

$$\stackrel{\text{cst}}{=} \bigcup_{p \in \hat{\delta}_N(q_0, x)} \delta_N(p, a)$$

$$\stackrel{\text{def}}{=} \hat{\delta}_N(q_0, xa)$$

Ora segue che  $L(D) = L(N)$ .

**Teorema 2.12:** Un linguaggio  $L$  e' accettato da un DFA  $\iff L$  e' accettato da un NFA.

**Prova:**

- La parte " $\Leftarrow$ " e' il Teorema 2.11.
  - Per la parte " $\Rightarrow$ " notiamo che un qualsiasi DFA puo' essere convertito in un NFA equivalente modificando la  $\delta_D$  in  $\delta_N$  secondo la regola seguente:
    - Se  $\delta_D(q, a) = p$ , allora  $\delta_N(q, a) = \{p\}$ .Si puo' mostrare per induzione su  $|w|$  che se  $\hat{\delta}_D(q_0, w) = p$ , allora  $\hat{\delta}_N(q_0, w) = \{p\}$ .
- 
- L'enunciato del teorema segue.

- **Esercizio 3 - I compito 2013:**

Dato il seguente NFA:  $N = (\{q_0, q_1\}, \{0, 1\}, \delta_N, q_0, \{q_1\})$ , dove  
 $\delta_N(q_0, 0) = \{q_0, q_1\}$ ,  $\delta_N(q_0, 1) = \{q_1\}$ ,  $\delta_N(q_1, 0) = \emptyset$ ,  
 $\delta_N(q_1, 1) = \{q_0, q_1\}$ .

- 1 Verificare se le stringhe  $w_1 = 101$  e  $w_2 = 0010$  vengono o meno accettate dall'automa mostrando tutti i passaggi.
- 2 Costruire il diagramma di transizione del DFA equivalente all'NFA dato
- 3 Descrivere a parole il linguaggio accettato dall'automa (suggerimento: è più facile osservando il DFA).

- **Esercizio 3 (1):**  $w_1 = 101$  e  $w_2 = 0010$  accettate dall'NFA  $N$ ?
  - $w_1 = 101 \in L(A)$ ?
    - $\hat{\delta}_N(q_0, \epsilon) = \{q_0\}$
    - $\hat{\delta}_N(q_0, 1) = \delta_N(q_0, 1) = \{q_1\}$
    - $\hat{\delta}_N(q_0, 10) = \delta_N(q_1, 0) = \emptyset$  quindi  $w_1 = 101 \notin L(A)$
  - $w_2 = 0010 \in L(A)$ ?
    - $\hat{\delta}_N(q_0, \epsilon) = \{q_0\}$
    - $\hat{\delta}_N(q_0, 0) = \delta_N(q_0, 0) = \{q_0, q_1\}$
    - $\hat{\delta}_N(q_0, 00) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
    - $\hat{\delta}_N(q_0, 001) = \delta_N(q_0, 1) \cup \delta_N(q_1, 1) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$
    - $\hat{\delta}_N(q_0, 0010) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
    - Poiche'  $\{q_0, q_1\} \cap q_1 \neq \emptyset$ ,  $w_2 = 0010 \in L(A)$

- **Esercizio 3 (2)** Costruire il diagramma di transizione del DFA equivalente all'NFA dato?

- $\delta_D(\{q_0\}, 0) = \delta_N(q_0, 0) = \{q_0, q_1\}$   
 $\delta_D(\{q_0\}, 1) = \delta_N(q_0, 1) = \{q_1\}$

- $\delta_D(\{q_0, q_1\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$   
 $\delta_D(\{q_0, q_1\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_1, 1) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$

- $\delta_D(\{q_1\}, 0) = \delta_N(q_1, 0) = \emptyset$   
 $\delta_D(\{q_1\}, 1) = \delta_N(q_1, 1) = \{q_0, q_1\}$

- Disegnare graficamente il diagramma:  
 $\{q_0\}$  stato iniziale,  $\{q_0, q_1\}$  e  $\{q_1\}$  stati accettanti

- **Esercizio 3 (3)** Descrivere il linguaggio accettato dall'automa

$$L(A) = \{w \in \{0, 1\}^+ \mid w = 1 \text{ o } w = 0x \text{ o } w = 11y, x, y \in \{0, 1\}^*\}$$



- Esercizi per casa:
  - Costruire un NFA che accetta l'insieme delle stringhe di 0 e di 1 con due 0 consecutivi o con due 1 consecutivi
  - Costruire un NFA che accetta l'insieme delle stringhe di  $a$  e di  $b$  e di  $c$  composte da zero o piu'  $a$  seguite da zero o piu'  $b$  seguite da zero o piu'  $c$
  - Costruire un NFA che accetta l'insieme delle stringhe di 0 e di 1 tale che almeno uno degli ultimi tre simboli e' 1
  - Esercizi 2.3.1 e 2.3.2 del libro:  
Convertire in un DFA un NFA dato
  - Esercizi delle dispense su NFA

# Automi a stati finiti non deterministici con $\epsilon$ -transizione ( $\epsilon$ -NFA)

- Sono un'estensione degli NFA dove permettono transizioni anche sulla stringa vuota
- Accettano sempre i linguaggi regolari
- Sono più facili da progettare degli NFA

Un  $\epsilon$ -NFA e' una quintupla

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

- $Q$  e' un insieme finito di stati
- $\Sigma$  e' un alfabeto finito
- $\delta$  e' una funzione da  $Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$
- $q_0 \in Q$  e' lo *stato iniziale*
- $F \subseteq Q$  e' un insieme di *stati finali*

NFA e  $\epsilon$ -NFA molto usati per la ricerca di un pattern in un testo

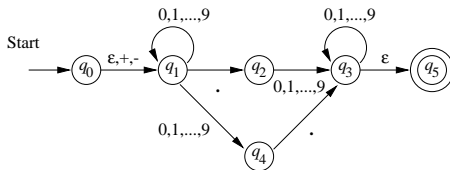
Esempio: Costruire un NFA che accetta l'insieme delle parole chiave  $\{web, ebay\}$

Esempio: Costruire un  $\epsilon$ -NFA che accetta l'insieme delle parole chiave  $\{web, ebay\}$

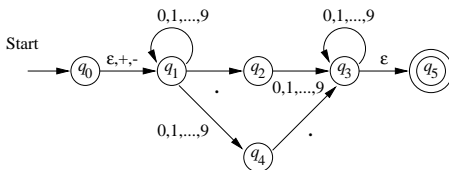
Un  $\epsilon$ -NFA che accetta numeri decimali consiste di:

- 1 Un segno + o -, opzionale
- 2 Una stringa di cifre decimali
- 3 un punto decimale
- 4 un'altra stringa di cifre decimali

Una delle stringhe (2) e (4) sono opzionali



# Esempio di $\epsilon$ -NFA



$$E = (\{q_0, q_1, \dots, q_5\}, \{., +, -, 0, 1, \dots, 9\}, \delta, q_0, \{q_5\})$$

dove la tabella delle transizioni per  $\delta$  e'

	$\epsilon$	$+, -$	$.$	$0, \dots, 9$
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\emptyset$
$q_1$	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_1, q_4\}$
$q_2$	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_3\}$
$q_3$	$\{q_5\}$	$\emptyset$	$\emptyset$	$\{q_3\}$
$q_4$	$\emptyset$	$\emptyset$	$\{q_3\}$	$\emptyset$
$\star q_5$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Chiudiamo uno stato aggiungendo tutti gli stati raggiungibili da lui tramite una sequenza  $\epsilon \epsilon \dots \epsilon$

Definizione induttiva di  $ECLOSE(q)$

**Base:**

$$q \in ECLOSE(q)$$

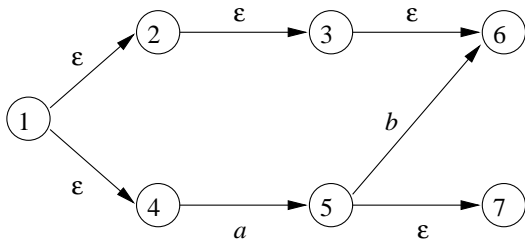
**Induzione:**

$$p \in ECLOSE(q) \text{ and } r \in \delta(p, \epsilon) \Rightarrow r \in ECLOSE(q)$$

Si puo' applicare  $ECLOSE$  anche a un insieme  $S$  di stati

$$ECLOSE(S) = \bigcup_{q \in S} ECLOSE(q)$$

# Esempio di epsilon-chiusura



Per esempio,

$$\text{ECLOSE}(1) = \{1, 2, 3, 4, 6\}$$



- Definizione induttiva di  $\hat{\delta}$  per automi  $\epsilon$ -NFA

**Base:**

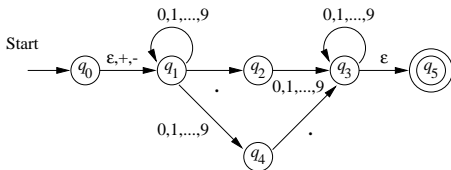
$$\hat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$

**Induzione:**

$$\hat{\delta}(q, xa) = \bigcup_{p \in \delta(t, a), t \in \hat{\delta}(q, x)} \text{ECLOSE}(p)$$

- Linguaggio accettato da un  $\epsilon$ -NFA  $A = (Q, \Sigma, \delta, q_0, F)$

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



Esercizio:  $5.6 \in L(E)$ ?

- $\hat{\delta}(q_0, \epsilon) = \text{ECLOSE}(q_0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 5) = \text{ECLOSE}(q_1) \cup \text{ECLOSE}(q_4) = \{q_1, q_4\}$
- $\hat{\delta}(q_0, 5.) = \text{ECLOSE}(q_2) \cup \text{ECLOSE}(q_3) = \{q_2, q_3, q_5\}$
- $\hat{\delta}(q_0, 5.6) = \text{ECLOSE}(q_3) = \{q_3, q_5\}$

$5.6 \in L(E)$  poiche'  $\hat{\delta}(q_0, 5.6) \cap q_5 \neq \emptyset$

- Introduzione agli automi a stati finiti
- Concetti di base della teoria degli automi
- Tecniche di dimostrazione
- Automi a stati finiti deterministici (DFA)
- Automi a stati finiti non deterministici (NFA): esercizi
- Automi a stati finiti non deterministici con  $\epsilon$ -transizione ( $\epsilon$ -NFA): teoremi ed esercizi

- 1 Costruire un NFA che accetta l'insieme delle stringhe di 0 e di 1 con due 0 consecutivi o con due 1 consecutivi
- 2 Costruire un NFA che accetta l'insieme delle stringhe di  $a$  e di  $b$  e di  $c$  composte da zero o piu'  $a$  seguite da zero o piu'  $b$  seguite da zero o piu'  $c$
- 3 Costruire un NFA che accetta l'insieme delle stringhe di 0 e di 1 tale che almeno uno degli ultimi tre simboli e' 1

Chiudiamo uno stato aggiungendo tutti gli stati raggiungibili da lui tramite una sequenza  $\epsilon \epsilon \cdots \epsilon$

Definizione induttiva di  $ECLOSE(q)$

**Base:**

$$q \in ECLOSE(q)$$

**Induzione:**

$$p \in ECLOSE(q) \text{ and } r \in \delta(p, \epsilon) \Rightarrow r \in ECLOSE(q)$$

Si puo' applicare  $ECLOSE$  anche a un insieme  $S$  di stati

$$ECLOSE(S) = \bigcup_{q \in S} ECLOSE(q)$$

- Definizione induttiva di  $\hat{\delta}$  per automi  $\epsilon$ -NFA

**Base:**

$$\hat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$

**Induzione:**

$$\hat{\delta}(q, xa) = \bigcup_{p \in \delta(t, a), t \in \hat{\delta}(q, x)} \text{ECLOSE}(p)$$

- Linguaggio accettato da un  $\epsilon$ -NFA  $A = (Q, \Sigma, \delta, q_0, F)$

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

Dato un  $\epsilon$ -NFA

$$E = (Q_E, \Sigma, \delta_E, q_0, F_E)$$

costruiremo un DFA

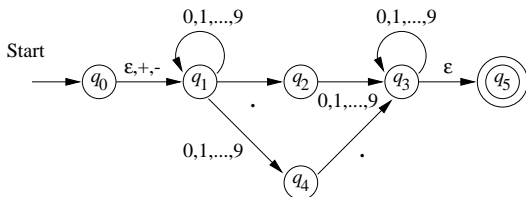
$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

tale che

$$L(D) = L(E)$$

Dettagli della costruzione:

- $Q_D = \{S : S \subseteq Q_E \text{ e } S = \text{ECLOSE}(S)\}$
- $q_D = \text{ECLOSE}(q_0)$
- $F_D = \{S : S \in Q_D \text{ e } S \cap F_E \neq \emptyset\}$
- $\delta_D(S, a) = \bigcup_{p \in \delta_E(t, a), t \in S} \text{ECLOSE}(p)$

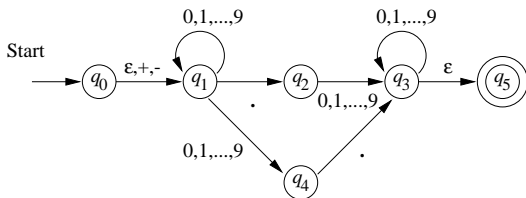
$\epsilon$ -NFA  $E$ Costruiamo DFA  $D$  corrispondente ad  $E$ :

- Si calcola  $q_D$  (stato iniziale)
- Per ogni stato  $S$  accessibile da  $q_D$  si calcola la funzione di transizione  $\delta_D(S, a)$ ,  $\forall a \in \Sigma = \{., +, -, 0, 1, \dots, 9\}$ 
  - $q_D = ECLOSE(q_0) = \{q_0, q_1\}$
  - $\delta_D(\{q_0, q_1\}, +) = ECLOSE(\delta_E(q_0, +)) \cup ECLOSE(\delta_E(q_1, +)) = ECLOSE(\{q_1\}) \cup ECLOSE(\emptyset) = \{q_1\} \cup \emptyset = \{q_1\}$
  - Completarlo per casa

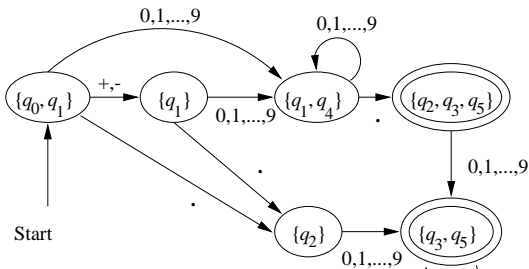


# Esempio

$\epsilon$ -NFA  $E$



DFA  $D$  corrispondente ad  $E$



**Teorema 2.22:** Un linguaggio  $L$  e' accettato da un  $\epsilon$ -NFA  $E \iff L$  e' accettato da un DFA.

**Prova:**

- ( $\Leftarrow$ ) Trasformiamo un DFA  $D$  in un  $\epsilon$ -NFA aggiungendo la transizione  $\delta_E(q, \epsilon) = \emptyset, \forall q$  in  $D$   
 $\forall q$  in  $D$  e  $\forall a \in \Sigma$  la transizione  $\delta_D(q, a) = p$  diventa  $\delta_E(q, a) = \{p\}$   
Poiche' le transizioni di  $D$  e  $E$  sono le stesse si conclude che  $L(E) = L(D)$ .
- ( $\Rightarrow$ ) Usiamo  $D$  costruito come sopra.  
 $L(E) = \{w \in \Sigma^* \mid \hat{\delta}_E(q_0, w) \cap F_E \neq \emptyset\}$   
 $L(D) = \{w \in \Sigma^* \mid \hat{\delta}_D(q_D, w) \in F_D\} = \{w \in \Sigma^* \mid \hat{\delta}_D(q_D, w) \cap F_E \neq \emptyset\}$   
Per dimostrare che  $L(D) = L(E)$  basta mostrare per induzione su  $|w|$  che  $\hat{\delta}_E(q_0, w) = \hat{\delta}_D(q_D, w)$

**Base:**  $\hat{\delta}_E(q_0, \epsilon) \stackrel{\text{def}}{=} \text{ECLOSE}(q_0) \stackrel{\text{cst}}{=} q_D \stackrel{\text{def}}{=} \hat{\delta}_D(q_D, \epsilon)$

**Ip. induttiva:**  $\hat{\delta}_E(q_0, x) = \hat{\delta}(q_D, x)$

**Induzione:**

$$\hat{\delta}_D(q_D, xa)$$

$$\stackrel{\text{def}}{=} \delta_D(\hat{\delta}_D(q_D, x), a)$$

$$\stackrel{\text{ip.ind}}{=} \delta_D(\hat{\delta}_E(q_0, x), a)$$

$$\stackrel{\text{cst}}{=} \bigcup_{p \in \delta_E(t, a), t \in \hat{\delta}_E(q_0, x)} \text{ECLOSE}(p)$$

$$\stackrel{\text{def}}{=} \hat{\delta}_E(q_0, xa)$$

- 1 Costruire un  $\epsilon$ -NFA che accetta l'insieme delle stringhe di  $a$  e di  $b$  e di  $c$  composte da zero o piu'  $a$  seguite da zero o piu'  $b$  seguite da zero o piu'  $c$
- 2 Costruire un  $\epsilon$ -NFA che accetta l'insieme delle stringhe di 0 e di 1 formate da 01 ripetuto una o piu' volte o da 010 ripetuto una o piu' volte
- 3 Costruire un  $\epsilon$ -NFA che accetta l'insieme delle stringhe di 0 e di 1 tale che almeno uno degli ultimi tre simboli e' 1
- 4 Esercizi 2.5.1\* e 2.5.2 del libro  
Dato un  $\epsilon$ -NFA  $E$ 
  - Calcolare l' $\epsilon$ -close di ogni stato di  $E$
  - Elencare tutte le stringhe di lunghezza minore o uguale a 3 accettate da  $E$
  - Trasformare  $E$  in un DFA equivalente
- 5 Esercizi delle dispense su  $\epsilon$ -NFA