# Linguaggi Formali e Traduttori

## 2.3 Automi a stati finiti con ε-transizioni

# Sommario

## Automi deterministici e non deterministici

Automi in cui ogni **transizione** corrisponde alla **lettura di un simbolo** nella stringa da riconoscere

$$\delta : Q \times \Sigma \to Q \qquad\qquad \delta : Q \times \Sigma \to \wp(Q)$$

## Automi con ε-transizioni

Automi che possono eseguire **transizioni spontanee** senza leggere alcun simbolo nella stringa da riconoscere

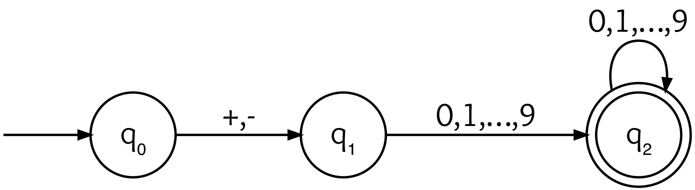
$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$$

## In questa lezione

1. Introduciamo la classe degli automi con **ε-transizioni**
2. Dimostriamo che ogni linguaggio riconosciuto da un automa con ε-transizioni può essere riconosciuto anche da un automa deterministico
3. Usiamo le ε-transizioni laddove conveniente, per esempio per rappresentare **parti facoltative** di una stringa da riconoscere o per costruire agevolmente automi complessi in maniera **modulare**

# Esempio: costanti numeriche con segno

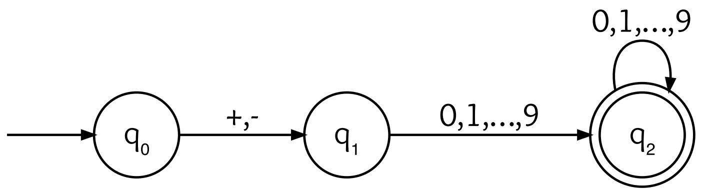
## Segno obbligatorio




- quando l'automa è nello stato $q_0$ si aspetta di riconoscere un segno (+ o -)

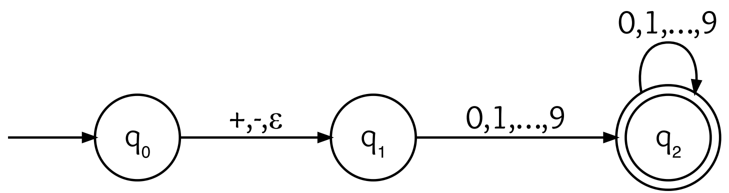
# Esempio: costanti numeriche con segno

## Segno obbligatorio




- quando l'automa è nello stato $q_0$ si aspetta di riconoscere un segno (+ o -)

## Segno **facoltativo**




- quando l'automa è nello stato $q_0$ può riconoscere un segno oppure spostarsi **spontaneamente** in $q_1$ e avvicinarsi allo stato finale
- abbiamo aggiunto **una transizione** da $q_0$ a $q_1$ invece di **10 transizioni** da $q_0$ a $q_2$

# Automi a stati finiti con **ε-transizioni**

## Definizione

Un **automa a stati finiti con ε-transizioni** (detto anche **ε-NFA**) è una quintupla $A = (Q, \Sigma, \delta, q_0, F)$ dove:

- $Q$ è un insieme **finito** di **stati**
- $\Sigma$ è l'**alfabeto** riconosciuto dall'automa
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$ è la **funzione di transizione** (notare il dominio)
- $q_0 \in Q$ è lo **stato iniziale**
- $F \subseteq Q$ è l'insieme di **stati finali**

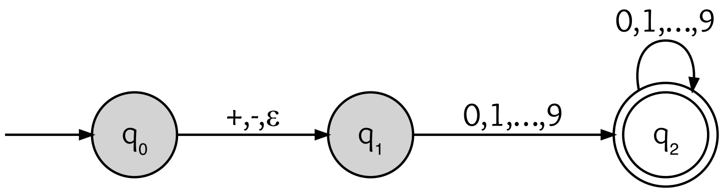
## Note

- $\delta(q, a)$ è l'insieme degli stati in cui l'ε-NFA può transire quando si trova nello stato $q$ leggendo il simbolo $a$
- $\delta(q, \varepsilon)$ è l'insieme degli stati in cui l'ε-NFA può **transire spontaneamente** quando si trova nello stato $q$, senza leggere alcun simbolo

# ε-chiusura

## Intuizione

Per definire il linguaggio riconosciuto da un ε-NFA, è importante riuscire a determinare quali stati sono raggiungibili grazie alle ε-transizioni




## Definizione

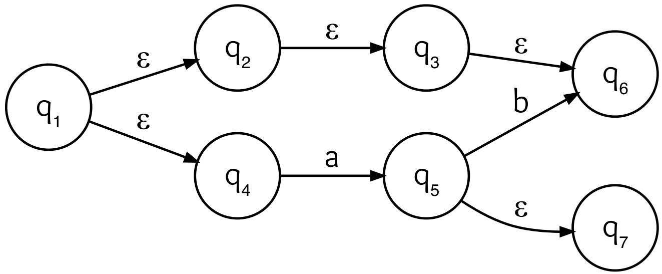
$\text{ECLOSE}(q)$ è il più piccolo insieme di stati tale che:

1. $q \in \text{ECLOSE}(q)$
2. se $p \in \text{ECLOSE}(q)$, allora $\delta(p, \varepsilon) \subseteq \text{ECLOSE}(q)$

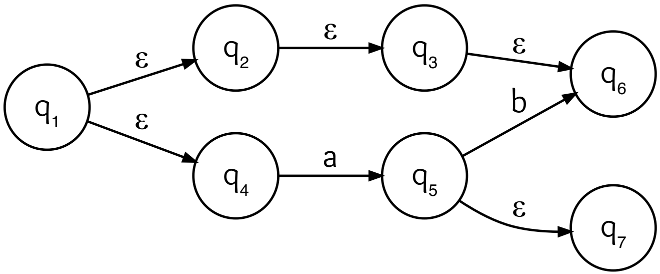
## Generalizzazione della ε-chiusura a insiemi di stati

Quando $S$ è un insieme di stati, definiamo $\text{ECLOSE}(S) = \bigcup_{q \in S} \text{ECLOSE}(q)$

# Esempio di calcolo della ε-chiusura



# Esempio di calcolo della ε-chiusura




- $\text{ECLOSE}(q_1) = \{q_1, q_2, q_3, q_4, q_6\}$
- $\text{ECLOSE}(q_2) = \{q_2, q_3, q_6\}$
- $\text{ECLOSE}(q_3) = \{q_3, q_6\}$
- $\text{ECLOSE}(q_4) = \{q_4\}$
- $\text{ECLOSE}(q_5) = \{q_5, q_7\}$
- $\text{ECLOSE}(q_6) = \{q_6\}$
- $\text{ECLOSE}(q_7) = \{q_7\}$

# Linguaggio riconosciuto da un ε-NFA

## Definizione

La **funzione di transizione estesa** dell'ε-NFA $A = (Q, \Sigma, \delta, q_0, F)$ è la funzione $\hat{\delta} : Q \times \Sigma^* \to \wp(Q)$ definita per induzione sul suo secondo argomento come segue:

$$\hat{\delta}(q, \varepsilon) = \text{ECLOSE}(q) \qquad \hat{\delta}(q, wa) = \{r \in \text{ECLOSE}(\delta(p, a)) \mid p \in \hat{\delta}(q, w)\}$$

## Definizione

Il **linguaggio riconosciuto** (o **accettato**) dall'ε-NFA $A = (Q, \Sigma, \delta, q_0, F)$ è denotato da $L(A)$ e definito come segue:

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

## Nota

- L'ε-NFA riconosce una stringa $w$ se **esiste** un percorso etichettato con $w$ che lo porta dallo stato iniziale $q_0$ a uno dei suoi stati finali in $F$

# NFA $\longrightarrow$ ε-NFA

## Teorema

Dato un NFA $N$, esiste un ε-NFA $E$ tale che $L(E) = L(N)$

## Dimostrazione

Basta osservare che un NFA è un caso particolare di ε-NFA in cui non ci sono ε-transizioni

## Conseguenze

- ogni linguaggio regolare (cioè riconosciuto da un DFA) è riconosciuto da un ε-NFA
- il potere riconoscitivo degli ε-NFA è almeno pari a quello dei DFA/NFA, che abbiamo già dimostrato essere equivalenti

# ε-NFA ⟶ DFA

## Teorema

Dato un ε-NFA $E$, esiste un DFA $D$ tale che $L(D) = L(E)$

## Intuizione

- usiamo la costruzione per sottoinsiemi come nel caso NFA ⟶ DFA
- occorre fare attenzione agli stati raggiungibili da ε-transizioni
- possiamo usare la nozione di ε-chiusura!

## Conseguenze

- ogni linguaggio riconosciuto da un ε-NFA è **regolare**
- combinando questo risultato e quello della slide 8, concludiamo che ε-NFA, NFA e DFA hanno lo **stesso potere riconoscitivo**

# ε-NFA $\longrightarrow$ DFA: costruzione per sottoinsiemi

Dato un ε-NFA $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$ definiamo $D = (Q_D, \Sigma, \delta_D, \text{ECLOSE}(q_0), F_D)$ dove

- $Q_D = \wp(Q_E)$, ovvero $Q_D$ è l'insieme dei sottoinsiemi di $Q_E$
- per ogni $S \subseteq Q_E$ e ogni $a \in \Sigma$ definiamo $\delta_D(S, a) = \bigcup_{q \in S} \text{ECLOSE}(\delta_E(q, a))$
- $F_D = \{S \subseteq Q_E \mid S \cap F_E \neq \emptyset\}$

Se si dimostra l'equazione

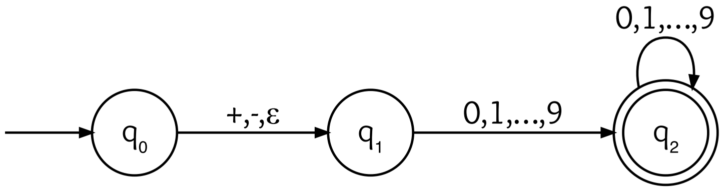
$$\hat{\delta}_E(q_0, w) = \hat{\delta}_D(\text{ECLOSE}(q_0), w)$$

si può concludere che

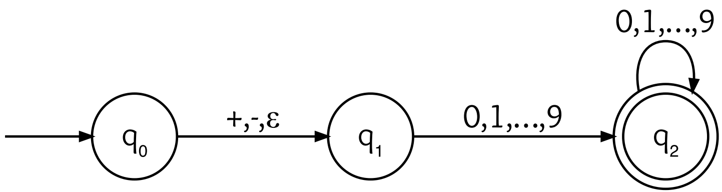
$$\begin{array}{rcll}
w \in L(E) & \iff & \hat{\delta}_E(q_0, w) \cap F_E \neq \emptyset & \text{def. di } L(E) \\
& \iff & \hat{\delta}_D(\text{ECLOSE}(q_0), w) \cap F_E \neq \emptyset & \text{equazione qui sopra} \\
& \iff & \hat{\delta}_D(\text{ECLOSE}(q_0), w) \in F_D & \text{def. di } F_D \\
& \iff & w \in L(D) & \text{def. di } L(D)
\end{array}$$

Dettagli nel libro di testo.

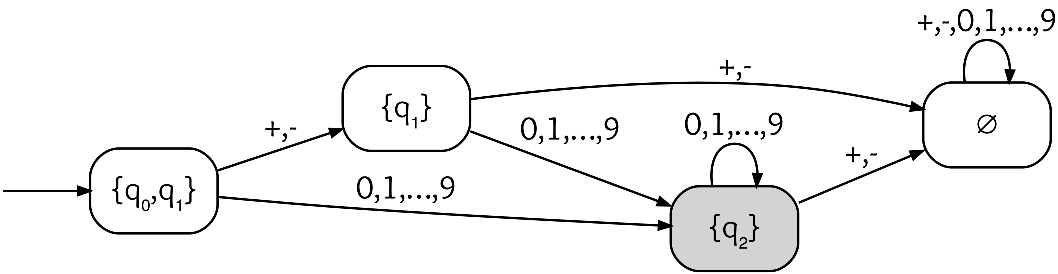
# Esempio: costanti numeriche con segno



# Esempio: costanti numeriche con segno




| | +,- | 0,1,...,9 |
|---|---|---|
| $\rightarrow \{q_0, q_1\}$ | $\{q_1\}$ | $\{q_2\}$ |
| $\{q_1\}$ | $\emptyset$ | $\{q_2\}$ |
| $*\{q_2\}$ | $\emptyset$ | $\{q_2\}$ |
| $\emptyset$ | $\emptyset$ | $\emptyset$ |



# Esempio: costruzione modulare di automi

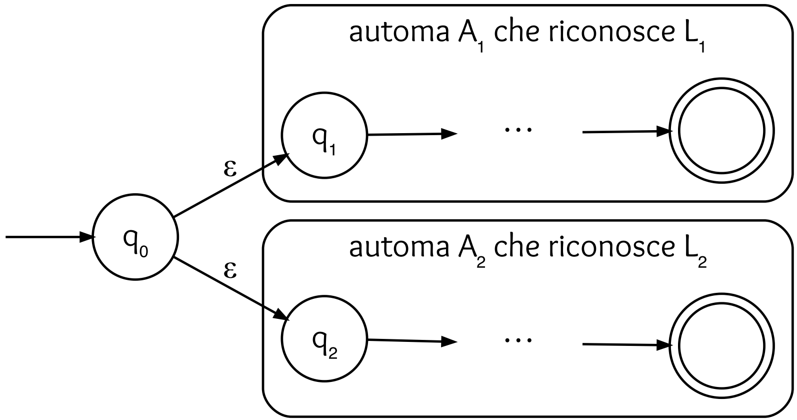
## Teorema

I linguaggi regolari sono **chiusi** rispetto all'operazione di **unione**, ovvero se $L_1$ ed $L_2$ sono regolari, allora anche $L_1 \cup L_2$ è regolare

## Dimostrazione

Se $L_1$ ed $L_2$ sono regolari, allora esistono due automi a stati finiti $A_1$ e $A_2$ tali che $L_1 = L(A_1)$ e $L_2 = L(A_2)$. Supponiamo $A_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ e $A_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$.

Costruiamo $A = (Q_1 \cup Q_2 \cup \{q_0\}, \Sigma, \delta, q_0, F_1 \cup F_2)$ dove




$$\delta(q, \alpha) = \begin{cases} \{q_1, q_2\} & \text{se } q = q_0 \text{ e } \alpha = \varepsilon \\ \delta_1(q, \alpha) & \text{se } q \in Q_1 \\ \delta_2(q, \alpha) & \text{se } q \in Q_2 \\ \emptyset & \text{altrimenti} \end{cases}$$

Abbiamo che $L(A) = L(A_1) \cup L(A_2)$

# Esercizi

1. Definire un ε-NFA che riconosca le stringhe composte da 0 o più a, seguite da 0 o più b, seguite da 0 o più c.
2. Definire un ε-NFA che riconosca le stringhe formate da 01 ripetuto una o più volte, o da 010 ripetuto una o più volte.

3. Per l'ε-NFA rappresentato in forma tabellare qui sotto, calcolare l'ε-chiusura di ciascuno stato, descrivere sommariamente il linguaggio accettato e convertire l'automa in DFA.

| | ε | a | b | c |
|---|---|---|---|---|
| $\rightarrow p$ | $\emptyset$ | $\{p\}$ | $\{q\}$ | $\{r\}$ |
| $q$ | $\{p\}$ | $\{q\}$ | $\{r\}$ | $\emptyset$ |
| $*r$ | $\{q\}$ | $\{r\}$ | $\emptyset$ | $\{p\}$ |

4. Per l'ε-NFA rappresentato in forma tabellare qui sotto, calcolare l'ε-chiusura di ciascuno stato, descrivere sommariamente il linguaggio accettato e convertire l'automa in DFA.

| | ε | a | b | c |
|---|---|---|---|---|
| $\rightarrow p$ | $\{q, r\}$ | $\emptyset$ | $\{q\}$ | $\{r\}$ |
| $q$ | $\emptyset$ | $\{p\}$ | $\{r\}$ | $\{p, q\}$ |
| $*r$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |

# Dimostrazioni

1. Dimostrare che per ogni DFA esiste un ε-NFA equivalente (cioè che riconosce lo stesso linguaggio) che ha esattamente **uno** stato finale
2. Dimostrare che i linguaggi regolari sono **chiusi** rispetto all'operazione di **concatenazione**. Suggerimento: se utile usare il risultato dimostrato nell'esercizio precedente