

# Automi a stati finiti

Corso di Fondamenti di Informatica - modulo 1  
 Corso di Laurea in Informatica  
 Università di Roma "Tor Vergata"  
 a.a. 2021-2022

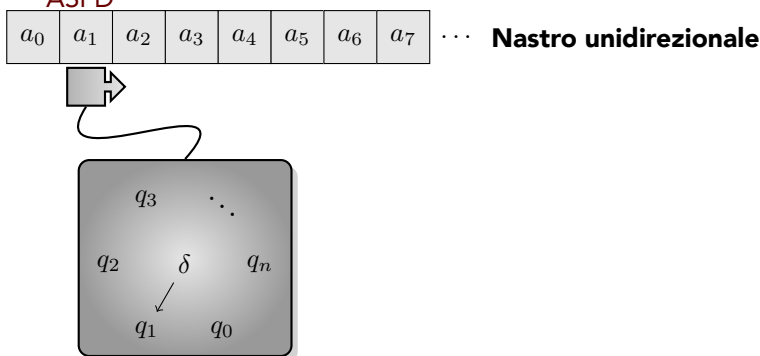
Giorgio Gambosi

## ASFD

Un **automa a stati finiti deterministico** (ASFD) è una quintupla  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ , dove

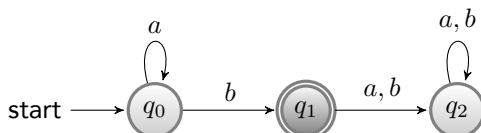
- $\Sigma = \{a_1, \dots, a_n\}$  è l'**alfabeto** di input
- $Q = \{q_0, \dots, q_m\}$  è un insieme finito e non vuoto di **stati**
- $q_0 \in Q$  è lo **stato iniziale**
- $F \subseteq Q$  è un insieme di **stati finali**
- $\delta : Q \times \Sigma \mapsto Q$  è la **funzione (totale) di transizione** che ad ogni coppia  $\langle \text{stato}, \text{carattere in input} \rangle$  associa uno stato successivo.

## ASFD



Funzione di transizione

$\delta$	a	b
$q_0$	$q_0$	$q_1$
$q_1$	$q_2$	$q_2$
$q_2$	$q_2$	$q_2$



## Configurazione di un ASF

Dato un automa a stati finiti  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ , una configurazione di  $\mathcal{A}$  è una coppia  $(q, x)$ , con  $q \in Q$  e  $x \in \Sigma^*$ .

Una configurazione  $\langle q, x \rangle$ ,  $q \in Q$  ed  $x \in \Sigma^*$ , di  $\mathcal{A}$ , è detta:

- **iniziale** se  $q = q_0$

- **finale** se  $x = \varepsilon$
- **accettante** se  $x = \varepsilon$  e  $q \in F$

#### Transizioni di un ASF

Dato un ASFD  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$  e due configurazioni  $(q, x)$  e  $(q', y)$  di  $\mathcal{A}$ , avremo che  $(q, x) \vdash_{\mathcal{A}}^* (q', y)$  se e solo se valgono le due condizioni:

1.  $x = ay$ , per un qualche  $a \in \Sigma$
2.  $\delta(q, a) = q'$ .

#### Accettazione da un ASFD

Dato un automa a stati finiti deterministico  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ , una stringa  $x \in \Sigma^*$  è accettata da  $\mathcal{A}$  se e solo se

$$(q_0, x) \vdash_{\mathcal{A}}^* (q, \varepsilon)$$

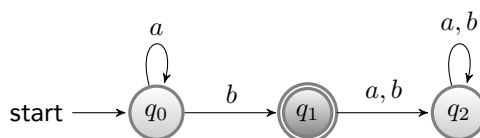
con  $q \in F$

Possiamo definire il linguaggio riconosciuto da  $\mathcal{A}$  come

$$L(\mathcal{A}) = \left\{ x \in \Sigma^* \mid (q_0, x) \vdash_{\mathcal{A}}^* (q, \varepsilon), q \in F \right\}.$$

#### Esempio

La stringa  $aab$  è accettata dall'automa a stati finiti deterministico



Infatti, a partire dalla configurazione iniziale  $(q_0, aab)$  l'automa raggiunge la configurazione di accettazione  $(q_1, \varepsilon)$  per mezzo della computazione

$$(q_0, aab) \vdash (q_0, ab) \vdash (q_0, b) \vdash (q_1, \varepsilon)$$

#### Funzione di transizione estesa

Dato un automa a stati finiti deterministico  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ , la sua **funzione di transizione estesa**

$$\bar{\delta} : Q \times \Sigma^* \mapsto Q$$

è definita come chiusura transitiva della  $\delta$ :

$$\begin{aligned} \bar{\delta}(q, \varepsilon) &= q \\ \bar{\delta}(q, xa) &= \delta(\bar{\delta}(q, x), a), \end{aligned}$$

dove  $a \in \Sigma, x \in \Sigma^*$ .

Una stringa  $x \in \Sigma^*$  è accettata da  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$  se e solo se  $\bar{\delta}(q_0, x) \in F$

#### Linguaggio riconosciuto da ASFD

Il **linguaggio riconosciuto** da un automa a stati finiti deterministico  $\mathcal{A}$  è l'insieme

$$L(\mathcal{A}) = \{ x \in \Sigma^* \mid \bar{\delta}(q_0, x) \in F \}.$$

### Esempio

Al fine di verificare che la stringa  $aab$  è accettata dall'ASFD precedente, deriviamo il valore di  $\delta(q_0, aab)$  nel modo seguente:

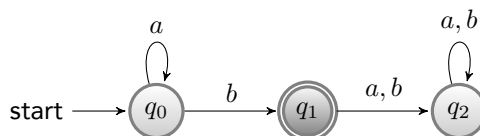
$$\begin{aligned}
 \bar{\delta}(q_0, aab) &= \delta(\bar{\delta}(q_0, aa), b) \\
 &= \delta(\delta(\bar{\delta}(q_0, a), a), b) \\
 &= \delta(\delta(\delta(\bar{\delta}(q_0, \varepsilon), a), a), b) \\
 &= \delta(\delta(\delta(q_0, a), a), b) \\
 &= \delta(\delta(q_0, a), b) \\
 &= \delta(q_0, b) \\
 &= q_1
 \end{aligned}$$

### Esercizio

Si consideri il linguaggio  $L = \{a^n b \mid n \geq 0\}$ . Questo linguaggio è generato dalla grammatica  $\mathcal{G}$  le cui produzioni sono:

$$S \rightarrow aS \mid b.$$

Si dimostri che il linguaggio  $L$  è il linguaggio riconosciuto dall'automa



### Esempio

Il linguaggio delle parole sull'alfabeto  $\{a, b\}$  che contengono un numero pari di  $a$  o un numero pari di  $b$  è riconosciuto dall'automa  $\mathcal{A} = \langle \{a, b\}, \{q_0, q_1, q_2, q_3\}, \delta, q_0, \{q_0, q_1, q_2\} \rangle$ , con funzione di transizione

$\delta$	$a$	$b$
$q_0$	$q_1$	$q_2$
$q_1$	$q_0$	$q_3$
$q_2$	$q_3$	$q_0$
$q_3$	$q_2$	$q_1$

### Esercizio

Definire un ASFD che riconosce le stringhe su  $\{a, b\}$  caratterizzate dal fatto che il secondo carattere è  $b$ .

### Esercizio

Definire un ASFD che riconosce le stringhe su  $\{a, b\}$  caratterizzate dal fatto che il penultimo carattere è  $b$ .

### Funzione di transizione parziale

La funzione di transizione  $\delta$  è stata definita come totale.

Ogni ASFD  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$  con funzione di transizione  $\delta$  non totale può essere trasformato in un ASFD  $\mathcal{A}' = \langle \Sigma, Q', \delta', q_0, F \rangle$  con funzione di transizione totale ed equivalente, ponendo  $Q' = Q \cup \{\bar{q}\}$  e  $\delta'$  tale che come:

1. Se  $\delta(q, a), q \in Q$  è definito allora  $\delta'(q, a) = \delta(q, a)$
2. Se  $\delta(q, a), q \in Q$  non è definito allora  $\delta'(q, a) = \bar{q}$
3.  $\delta'(\bar{q}, a) = \bar{q}$  per ogni  $a \in \Sigma$

### Linguaggi decisi da ASFD

Insieme  $\mathcal{R}$  dei linguaggi decisi da automi a stati finiti deterministici:

$$\mathcal{R} = \{L \subseteq \Sigma^* \mid \exists \text{ ASFD } \mathcal{A} \text{ tale che } L = L(\mathcal{A})\}$$

Questa classe di linguaggi coincide con quella dei linguaggi generati dalle grammatiche di tipo 3 e con quella dei linguaggi definiti da espressioni regolari

### ASFND

Un **automa a stati finiti non deterministico** è una quintupla  $\mathcal{A}_N = \langle \Sigma, Q, \delta_N, q_0, F \rangle$ , in cui

- $\Sigma = \{a_1, \dots, a_n\}$  è l'**alfabeto di input**
- $Q = \{q_0, \dots, q_m\}$  è un insieme finito e non vuoto di **stati**
- $q_0 \in Q$  è lo **stato iniziale**
- $F \subseteq Q$  è l'insieme degli **stati finali**
- $\delta_N : Q \times \Sigma \mapsto \mathcal{P}(Q)$  è una funzione (parziale), detta **funzione di transizione**, che ad ogni coppia  $\langle \text{stato}, \text{carattere} \rangle$  su cui è definita associa un sottoinsieme di  $Q$  (eventualmente vuoto)

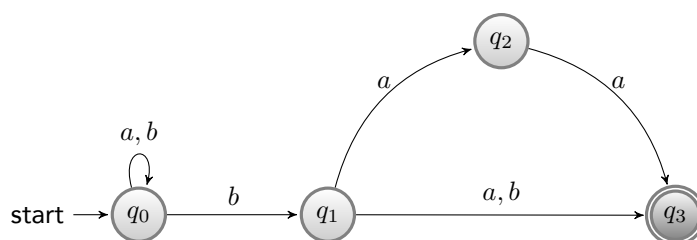
### Esempio

Funzione di transizione di un automa a stati finiti non deterministico:

$\delta_N$	a	b
$q_0$	$\{q_0\}$	$\{q_0, q_1\}$
$q_1$	$\{q_2, q_3\}$	$\{q_3\}$
$q_2$	$\{q_3\}$	$\emptyset$
$q_3$	$\emptyset$	$\emptyset$

### Esempio

Un automa a stati finiti non deterministico può essere anch'esso descritto, così come un ASFD, per mezzo di un grafo di transizione



### Computazioni di un AFSND

Un automa a stati finiti non deterministico definisce, data una stringa in input, un insieme di computazioni.

Alternativamente, possiamo considerare che l'automa esegua una sola **computazione non deterministica** nel corso della quale, per ogni carattere letto, assume non uno solo, ma un insieme di stati attuali e transita, ad ogni nuovo carattere, non da stato a stato ma da un insieme di stati ad un insieme di stati.

### Esempio

L'automa precedente definisce, in corrispondenza alla stringa in input  $bba$ , le tre computazioni:

- $(q_0, bba) \vdash (q_0, ba) \vdash (q_0, a) \vdash (q_0, \varepsilon)$ ;
- $(q_0, bba) \vdash (q_0, ba) \vdash (q_1, a) \vdash (q_2, \varepsilon)$ ;
- $(q_0, bba) \vdash (q_0, ba) \vdash (q_1, a) \vdash (q_3, \varepsilon)$ .

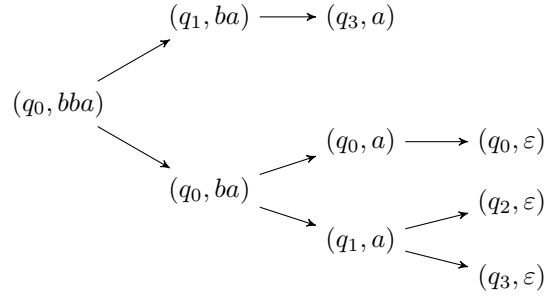
Il prefisso  $bb$  della stringa di input dà luogo anche alla computazione:

- $(q_0, bb) \vdash (q_1, b) \vdash (q_3, \varepsilon)$ ;

la quale però non presenta continuazioni possibili.

### Esempio

Albero di computazione corrispondente



### Esempio

Alternativamente, possiamo considerare che l'automa definisca la computazione non deterministica:

$$(\{q_0\}, bba) \vdash^* (\{q_0, q_1\}, ba) \vdash^* (\{q_0, q_1, q_3\}, a) \vdash^* (\{q_0, q_2, q_3\}, \varepsilon)$$

### Accettazione da ASFND

Una stringa  $x$  viene accettata da un automa a stati finiti non deterministico se almeno una delle computazioni definite per la stringa stessa è di accettazione, quindi se

$$(\{q_0\}, x) \vdash^* (\mathcal{Q}, \varepsilon) \quad \text{con} \quad \mathcal{Q} \subseteq Q$$

e

$$\mathcal{Q} \cap F \neq \emptyset$$

### Funzione di transizione estesa di un ASFND

Dato un ASFND, la **funzione di transizione estesa** è la funzione  $\bar{\delta}_N : Q \times \Sigma^* \mapsto \mathcal{P}(Q)$ , definita nel seguente modo

$$\begin{aligned} \bar{\delta}_N(q, \varepsilon) &= \{q\} \\ \bar{\delta}_N(q, xa) &= \bigcup_{p \in \bar{\delta}_N(q, x)} \delta_N(p, a) \end{aligned}$$

dove  $a \in \Sigma$ ,  $x \in \Sigma^*$ ,  $p \in Q$ .

Dato uno stato  $q$  ed una stringa  $x$  in input,  $q' \in \bar{\delta}_N(q, x)$  se e solo se esiste una computazione dell'automa la quale, a partire da  $q$  ed in conseguenza della lettura della stringa  $x$ , conduce allo stato  $q'$

### Linguaggio accettato da UN ASFND

Il linguaggio  $L(\mathcal{A})$  accettato da un ASFND  $\mathcal{A}$  è definito come:

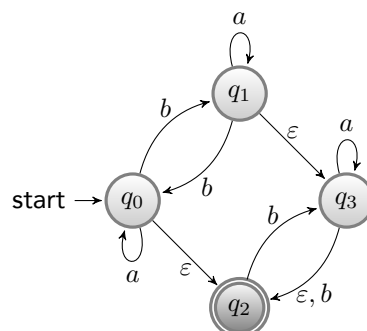
$$L(\mathcal{A}) = \left\{ x \in \Sigma^* \mid (\{q_0\}, x) \vdash^* (\mathcal{Q}, \varepsilon), \mathcal{Q} \cap F \neq \emptyset \right\}.$$

o anche come

$$L(\mathcal{A}_N) = \{x \in \Sigma^* \mid \bar{\delta}_N(q_0, x) \cap F \neq \emptyset\}.$$

### $\varepsilon$ -ASFND

Un  $\varepsilon$ -ASFND è un ASFND con  $\delta_N : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto \mathcal{P}(Q)$ , nella quale sono quindi ammesse  $\varepsilon$ -transizioni.



### Non determinismo e $\varepsilon$ -transizioni

La presenza di  $\varepsilon$  transizioni in un ASF rende di per sé l'automa non deterministico

La singola  $\varepsilon$ -transizione, potendo aver luogo o meno, è inerentemente non deterministica

### $\varepsilon$ -chiusura

Dato uno stato  $q$  dell'automa, la sua  $\varepsilon$ -chiusura è l'insieme  $\varepsilon(q)$  degli stati raggiungibili da  $q$  mediante una sequenza (anche nulla) di  $\varepsilon$ -transizioni.

Con riferimento all'automa precedente:

$q$	$\varepsilon(q)$
$q_0$	$\{q_0, q_2\}$
$q_1$	$\{q_1, q_2, q_3\}$
$q_2$	$\{q_2\}$
$q_3$	$\{q_2, q_3\}$

### Funzione di transizione estesa

La **funzione di transizione estesa**  $\hat{\delta}$  è definita come:

1.  $\hat{\delta}(q, \varepsilon) = \varepsilon(q)$
2. Per ogni  $x \in \Sigma^*$  e per ogni  $a \in \Sigma$ ,  $\hat{\delta}(q, xa) = \bigcup_{p \in P} \varepsilon(p)$ , dove  $P = \{p \mid \exists r \in \hat{\delta}(q, x) \text{ tale che } p \in \delta(r, a)\}$ ; vale a dire l'unione delle  $\varepsilon$ -chiusure di tutti gli stati raggiungibili da un qualche stato in  $\hat{\delta}(q, x)$ , avendo in input il carattere  $a$

Dato  $P \subseteq Q$ ,  $\varepsilon(P)$  è l'unione delle  $\varepsilon$ -chiusure di tutti gli stati in  $P$ :  $\varepsilon(P) = \bigcup_{p \in P} \varepsilon(p)$ .

### Esempio

Per l'automa precedente,

$\hat{\delta}(q_0, \varepsilon) = \varepsilon(q_0) = \{q_0, q_2\}$ , e quindi  $\hat{\delta}(q_0, a) = \varepsilon(P)$ , dove  $P = \delta(q_0, a) \cup \delta(q_2, a) = \{q_0\}$ , per cui  $\hat{\delta}(q_0, a) = \varepsilon(q_0) = \{q_0, q_2\}$

$\hat{\delta}(q_0, ab) = \varepsilon(P)$ , con  $P = \delta(q_0, b) \cup \delta(q_2, b) = \{q_1, q_3\}$ , per cui  $\hat{\delta}(q_0, ab) = \varepsilon(q_1) \cup \varepsilon(q_3) = \{q_1, q_2, q_3\}$

### Equivalenza ASFND e $\varepsilon$ -ASFND

Dato un ASFND che riconosce un linguaggio  $L$ , esiste corrispondentemente un  $\varepsilon$ -ASFND che riconosce lo stesso linguaggio  $L$ ; viceversa, dato un  $\varepsilon$ -ASFND che riconosce un linguaggio  $L'$ , esiste un ASFND che riconosce lo stesso linguaggio  $L'$ .

La prima implicazione è evidente, in quanto un ASFND è un caso particolare di  $\varepsilon$ -ASFND

### Equivalenza ASFND e $\varepsilon$ -ASFND

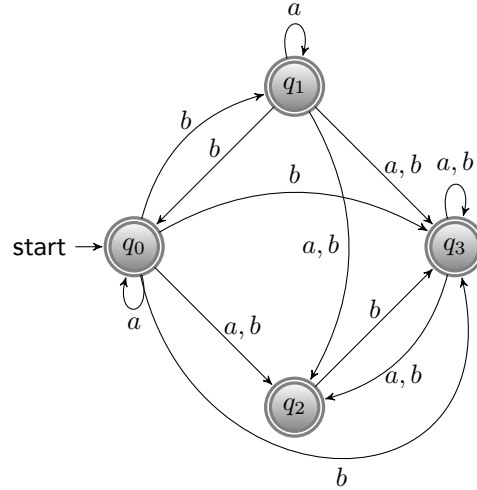
Per l'implicazione inversa, definiamo una procedura algoritmica che deriva una ASFND equivalente da un  $\varepsilon$ -ASFND dato.

1. per ogni  $q \in Q$  e per ogni  $a \in \Sigma$ ,  $\delta_N(q, a) = \hat{\delta}(q, a)$
2.  $q'_0 = q_0$
3.  $F' = \{q \in Q \mid \varepsilon(q) \cap F \neq \emptyset\}$

Problema: mostrare che i due automi sono equivalenti.

### Equivalenza ASFND e $\varepsilon$ -ASFND

Per l'automa precedente, si ottiene



### Esercizio

Definiamo come  $\varepsilon$ -ASFD un automa a stati finiti deterministico esteso da un insieme di  $\varepsilon$ -transizioni.

Mostrare l'equivalenza tra  $\varepsilon$ -ASFD e ASFND.

### Equivalenza tra ASFD e ASFND

Dato un ASFD che riconosce un linguaggio  $L$ , esiste corrispondentemente un ASFND che riconosce lo stesso linguaggio  $L$ ; viceversa, dato un ASFND che riconosce un linguaggio  $L'$ , esiste un ASFD che riconosce lo stesso linguaggio  $L'$ .

### Equivalenza tra ASFD e ASFND

Dato un ASFD  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ , un ASFND equivalente  $\mathcal{A}_N = \langle \Sigma, Q', \delta_N, q'_0, F' \rangle$  è derivabile immediatamente ponendo

- $Q' = Q$
- $q'_0 = q_0$
- $F' = F$
- $\delta_N$  tale che  $\forall a \in \Sigma, q \in Q, \delta_N(q, a) = \{q'\}$  se e solo se  $\delta(q, a) = q'$

### Equivalenza tra ASFD e ASFND

Dato un ASFND  $\mathcal{A}_N = \langle \Sigma, Q, \delta, q_0, F \rangle$ , un ASFD equivalente  $\mathcal{A}' = \langle \Sigma, Q', \delta', q'_0, F' \rangle$  è derivabile nel modo seguente:

L'insieme  $Q'$  è in corrispondenza biunivoca con  $\mathcal{P}(Q)$  (quindi  $|Q'| = 2^{|Q|}$ ).

Indichiamo come  $[q_{i_1}, \dots, q_{i_k}] \in Q'$  lo stato corrispondente all'insieme  $\{q_{i_1}, \dots, q_{i_k}\} \subseteq Q$ : i nomi degli stati di  $Q$  sono ordinati lessicograficamente.

Quindi  $Q'$  risulta definito come:

$$Q' = \{[q_{i_1}, \dots, q_{i_k}] \mid \{q_{i_1}, \dots, q_{i_k}\} \in \mathcal{P}(Q)\}.$$

### Equivalenza tra ASFD e ASFND

Lo stato iniziale è  $q'_0 = [q_0]$ .

Gli stati finali  $F'$  corrispondono ai sottoinsiemi di  $Q$  che contengono almeno un elemento di  $F$ :

$$F' = \{[q_{i_1}, \dots, q_{i_k}] \mid \{q_{i_1}, \dots, q_{i_k}\} \in \mathcal{P}(Q) \wedge \{q_{i_1}, \dots, q_{i_k}\} \cap F \neq \emptyset\}.$$

### Equivalenza tra ASFD e ASFND

$\delta'$  è definita nel seguente modo:

$$\forall q_{i_1}, \dots, q_{i_k} \in Q, \forall a \in \Sigma, \delta'([q_{i_1}, \dots, q_{i_k}], a) = [q_{j_1}, \dots, q_{j_h}],$$

se e solo se

$$\delta_N(q_{i_1}, a) \cup \dots \cup \delta_N(q_{i_k}, a) = \{q_{j_1}, \dots, q_{j_h}\}$$

con  $k > 0$  e  $h \geq 0$ .

Inoltre, si assume che per ogni  $a \in \Sigma$  sia  $\delta'([ ], a) = [ ]$ .

#### Equivalenza tra ASFD e ASFND

Come mostrare che  $\mathcal{A}_N$  e  $\mathcal{A}$  sono equivalenti?

È necessario mostrare che, per ogni  $x \in \Sigma^*$ , se  $x$  è accettata da  $\mathcal{A}_N$  allora è accettata da  $\mathcal{A}$ . Inoltre, se è accettata da  $\mathcal{A}$  allora è accettata anche da  $\mathcal{A}_N$ .

#### Equivalenza tra ASFD e ASFND

Dimostrazione "più forte", che ad ogni computazione effettuata dall'automa  $\mathcal{A}$  ne corrisponde una equivalente dell'automa  $\mathcal{A}_N$  e viceversa. Cioè che  $\forall x \in \Sigma^*$ ,

$$\bar{\delta}'([q_0], x) = [q_{j_1}, \dots, q_{j_h}] \iff \bar{\delta}_N(q_0, x) = \{q_{j_1}, \dots, q_{j_h}\}.$$

Se questo è vero, ne deriva che ogni stringa  $x$  o è accettata da sia da  $\mathcal{A}$  che da  $\mathcal{A}_N$ , o non è accettata da nessuno dei due automi.

#### Equivalenza tra ASFD e ASFND

Dimostrazione per induzione su  $|x|$ .

Passo base: ( $|x| = 0$ ). In questo caso vale necessariamente  $x = \varepsilon$ , per cui abbiamo  $\bar{\delta}_N(q_0, \varepsilon) = \{q_0\}$  e  $\bar{\delta}'([q_0], \varepsilon) = [q_0]$ .

#### Equivalenza tra ASFD e ASFND

Passo induttivo: ( $|x| > 0$ ). Supponiamo che la proprietà sia vera per  $|x| = m$  e dimostriamo che esso continua a valere per  $|x| = m + 1$ .

Poniamo  $x = x'a$ , con  $|x'| = m$ . Per  $\bar{\delta}_N$  abbiamo:

$$\bar{\delta}_N(q_0, x'a) = \bigcup_{p \in \bar{\delta}_N(q_0, x')} \delta_N(p, a).$$

Supponendo che  $\bar{\delta}_N(q_0, x') = \{q_{i_1}, \dots, q_{i_k}\}$  e che  $\delta_N(q_{i_1}, a) \cup \dots \cup \delta_N(q_{i_k}, a) = \{q_{j_1}, \dots, q_{j_h}\}$  otteniamo

$$\bar{\delta}_N(q_0, x'a) = \{q_{j_1}, \dots, q_{j_h}\}.$$

#### Equivalenza tra ASFD e ASFND

Per  $\bar{\delta}'$  vale:

$$\bar{\delta}'(q_0, x'a) = \delta'(\bar{\delta}'([q_0], x'), a).$$

Essendo  $|x'| = m$  possiamo sfruttare l'ipotesi induttiva, e quindi:

$$\delta'(\bar{\delta}'([q_0], x'), a) = \delta'([q_{i_1}, \dots, q_{i_k}], a),$$

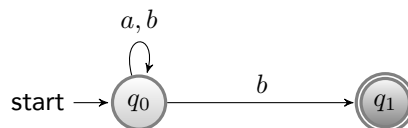
che, per costruzione, vale proprio  $[q_{j_1}, \dots, q_{j_h}]$ .

#### Equivalenza tra ASFD e ASFND

La dimostrazione è completata osservando che  $\bar{\delta}'([q_0], x) \in F'$  esattamente quando  $\bar{\delta}_N(q_0, x)$  contiene uno stato di  $Q$  che è in  $F$ .

#### Esempio

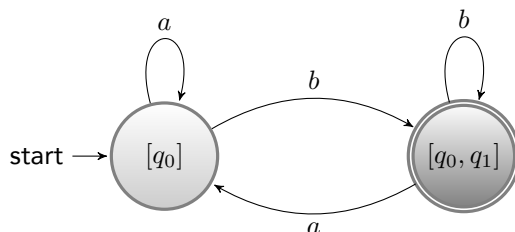
ASFND che riconosce le stringhe in  $\{a, b\}^*$  terminanti con  $b$





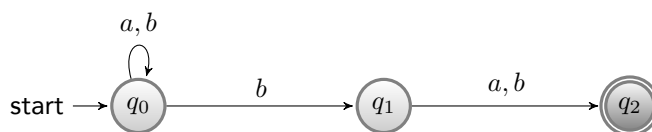
### Esempio

ASFD che riconosce le stringhe in  $\{a, b\}^*$  terminanti con  $b$



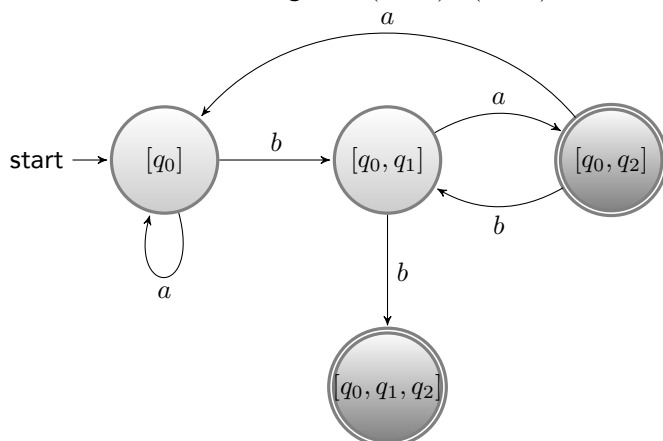
### Esempio

ASFND che riconosce le stringhe in  $(a + b)^*b(a + b)$



### Esempio

ASFD che riconosce le stringhe in  $(a + b)^*b(a + b)$



### Esercizio

Costruire l'ASFND e l'ASFD che riconoscono le stringhe  $(a + b)^*b(a + b)^k$ , per  $k = 2, 3$  e  $4$  e valutare come cresce la dimensione dei due automi al crescere di  $k$ .

### Esercizio

Data una stringa  $x = a_1 \cdots a_r$  definiamo stringa **riflessa** di  $x$  la stringa  $\tilde{x} = a_r \cdots a_1$ .

Dimostrare che se un linguaggio  $L$  è riconosciuto da un ASF, allora esiste un ASF che riconosce il linguaggio  $\{\tilde{x} \mid x \in L\}$ .

### Minimizzazione di automi a stati finiti

L'ASFD con minimo numero di stati che riconosce un dato linguaggio  $L$  può essere derivato partizionando l'insieme  $Q$  degli stati di un automa che riconosce  $L$  in classi di equivalenza rispetto alla relazione

$$q_i \equiv q_j \iff (\forall x \in \Sigma^* \quad \bar{\delta}(q_i, x) \in F \iff \bar{\delta}(q_j, x) \in F).$$

Quindi,  $q_i \equiv q_j$  se e solo se ogni stringa che porta da  $q_i$  ad uno stato finale porta anche da  $q_j$  ad uno stato finale (e vice versa).

### Minimizzazione di automi a stati finiti

$\equiv$  è una relazione di equivalenza.

Se  $q_i \equiv q_j$  i due stati sono detti **indistinguibili**

Se esiste una stringa  $x \in \Sigma^*$  per cui  $\bar{\delta}(q_i, x) \in F$  e  $\bar{\delta}(q_j, x) \in Q - F$  (o viceversa) diremo che  $q_i$  e  $q_j$  sono **distinguibili** tramite  $x$ .

#### Minimizzazione di automi a stati finiti

La costruzione è basata sul teorema di Myhill-Nerode.

**Teorema 1** (Myhill-Nerode). Dato un linguaggio  $L$  definiamo la relazione di equivalenza  $R_L$  su  $\Sigma^*$  come:

$$x R_L y \iff (\forall z \in \Sigma^* \quad xz \in L \iff yz \in L).$$

Un linguaggio  $L \subseteq \Sigma^*$  è regolare se e solo se  $R_L$  partiziona  $\Sigma^*$  in un numero finito di classi di equivalenza.

#### Minimizzazione di automi a stati finiti

Dato un linguaggio regolare  $L$ , definiamo un ASF  $\mathcal{A}' = \langle \Sigma, Q', \delta', q'_0, F' \rangle$ , dove:

1.  $Q'$  associa a ogni classe  $[x]$  di  $\Sigma^*$  uno stato  $q_{[x]}$
2.  $q'_0 = q_{[\varepsilon]}$
3.  $F' = \{q_{[x]} \mid x \in L\}$
4.  $\delta'(q_{[x]}, a) = q_{[xa]} \quad \forall a \in \Sigma$

#### Minimizzazione di automi a stati finiti

Si può notare che  $\forall z \in [x] : \bar{\delta}'(q'_0, z) = q_{[x]}$ . Quindi:

- l'insieme delle stringhe che portano l'automa dallo stato iniziale allo stato  $q_{[x]}$  corrisponde all'insieme delle stringhe (equivalenti secondo  $R_L$ ) in  $[x]$ . dalla definizione di  $F'$ , ne consegue immediatamente che  $\mathcal{A}'$  riconosce  $L$ .
- $Q'$ , pari al numero di classi di equivalenza di  $R_L$  è il minimo numero di classi in cui è possibile partizionare  $\Sigma^*$  in modo tale che valga la proprietà  $\forall z \in \Sigma^* \quad xz \in L \iff yz \in L$  se  $x R_L y$

#### Minimizzazione di automi a stati finiti

$Q'$  è minimo.

Se così non fosse, esisterebbero due stringhe  $x, y$  non equivalenti rispetto a  $R_L$  e per cui  $\bar{\delta}'(q'_0, x) = \bar{\delta}'(q'_0, y) = q$ . Ma allora ne seguirebbe che  $\forall z \in \Sigma^* : \bar{\delta}'(q'_0, xz) = \bar{\delta}'(q'_0, yz)$ , il che comporterebbe allora che  $x R_L y$ , contrariamente all'ipotesi.

#### Minimizzazione di automi a stati finiti

Minimizzazione di un ASFD:

- individuazione di tutte le coppie di stati indistinguibili (mediante un algoritmo di marcatura delle coppie distinguibili)
- unificazione degli stati equivalenti, eliminando quelli non raggiungibili e modificando opportunamente la funzione di transizione.

#### Minimizzazione di automi a stati finiti

Ipotesi:

tutti gli stati di  $\mathcal{A}$  sono raggiungibili dallo stato iniziale, altrimenti è necessario un passo preliminare di eliminazione degli stati irraggiungibili.

#### Minimizzazione di automi a stati finiti

- Per marcare le coppie di stati distinguibili si utilizza una tabella contenente una casella per ciascuna coppia (non ordinata) di stati di  $Q$
- Le caselle vengono usate per marcare le coppie di stati distinguibili e per elencare, in una lista associata, tutte le coppie che dovranno essere marcate qualora la coppia a cui è associata la casella venga marcata.

### Minimizzazione di automi a stati finiti

- La procedura inizia con la marcatura delle coppie distinguibili tramite la stringa  $\varepsilon$  (tutte e sole le coppie costituite da uno stato finale e da uno non finale)
- Per ogni coppia  $(p, q)$  non ancora marcata, si considerano, per ogni  $a \in \Sigma$ , tutte le coppie  $(r, s)$ , con  $r = \delta(p, a)$  e  $s = \delta(q, a)$ .
  - Se nessuna delle coppie  $(r, s)$  è marcata come distinguibile allora si inserisce  $(p, q)$  nella lista associata ad ognuna di esse
  - Altrimenti  $p$  e  $q$  vengono riconosciuti distinguibili e la corrispondente casella viene marcata; qualora questa contenga una lista di coppie si procede (ricorsivamente) con la marcatura delle relative caselle

### Minimizzazione di automi a stati finiti

**input** automa a stati finiti  $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ ;

**output** coppie di stati distinguibili di  $Q$ ;

**begin**

**for**  $p \in F$  and  $q \in Q - F$  **do**

    marca  $(p, q)$  e  $(q, p)$ ;

**for each** coppia non marcata di stati distinti **do**

**if**  $\exists a \in \Sigma : (\delta(p, a), \delta(q, a))$  distinguibili **then**

**begin**

        marca  $(p, q)$ ;

        marca ricorsivamente tutte le coppie non ancora marcate  
        sulla lista di  $(p, q)$  e su quelle delle coppie marcate  
        a questo passo

**end**

**else**

**for**  $a \in \Sigma$  **do**

**if**  $\delta(p, a) \neq \delta(q, a)$  **and**  $(p, q) \neq (\delta(p, a), \delta(q, a))$  **then**  
          aggiungi  $(p, q)$  alla lista di  $(\delta(p, a), \delta(q, a))$

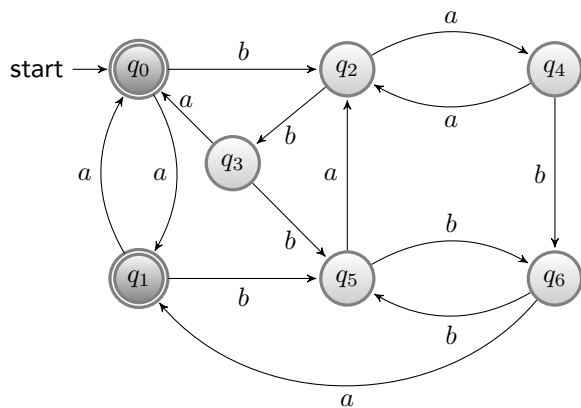
**end.**

### Minimizzazione di automi a stati finiti

Una volta identificate le coppie di stati indistinguibili, ricordando che la relazione di indistinguibilità è una relazione di equivalenza, l'automata equivalente con il minimo numero di stati è dato evidentemente da  $\mathcal{A}' = \langle \Sigma, Q', \delta', q'_0, F' \rangle$ , in cui:

1.  $Q'$  è costruito selezionando, per ogni insieme di stati indistinguibili, uno ed un solo stato di  $Q$  (rappresentante);
2.  $F'$  è costituito da tutti i rappresentanti appartenenti ad  $F$ ;
3.  $\delta'$  è ottenuta da  $\delta$  mediante restrizione al dominio  $Q' \times \Sigma$  ed inoltre, per ogni  $\delta(q_i, a) = q_j$ , con  $q_i \in Q'$  e  $q_j \in Q$ , poniamo  $\delta'(q_i, a) = q_k$ , dove  $q_k \in Q'$  è il rappresentante dell'insieme di stati indistinguibili che include  $q_j$  (chiaramente, se  $q_j \in Q'$  allora è esso stesso un rappresentante e dunque  $q_k = q_j$ ).

### Minimizzazione di automi a stati finiti



### Minimizzazione di automi a stati finiti

Passo iniziale:  $q_0$  e  $q_1$ , finali, distinguibili da tutti gli altri

1						
2	x	x				
3	x	x				
4	x	x				
5	x	x				
6	x	x				
	0	1	2	3	4	5

### Minimizzazione di automi a stati finiti

Consideriamo le coppie di stati scandendo le celle da sinistra verso destra e dall'alto verso il basso.

$(q_0, q_1)$  distinguibile se lo è  $(q_2, q_5)$ , in quanto  $\delta(q_0, b) = q_2$  e  $\delta(q_1, b) = q_5$ : quindi,  $(0, 1)$  inserito nella lista associata a  $(2, 5)$

1						
2	x	x				
3	x	x				
4	x	x				
5	x	x	(0,1)			
6	x	x				
	0	1	2	3	4	5

### Minimizzazione di automi a stati finiti

$(q_2, q_3)$  distinguibile in quanto  $\delta(q_2, a) = q_4$ ,  $\delta(q_3, a) = q_0$ , e la coppia  $(q_0, q_4)$  è distinguibile, (la cella  $(0, 4)$  è marcata): quindi viene marcata anche la cella  $(2, 3)$

1						
2	x	x				
3	x	x	x			
4	x	x				
5	x	x	(0,1)			
6	x	x				
	0	1	2	3	4	5

#### Minimizzazione di automi a stati finiti

Proseguendo, l'algoritmo determina che:

- la coppia  $(q_2, q_4)$  è distinguibile se lo è la coppia  $(q_3, q_6)$ :  $(2, 4)$  è inserito nella lista della cella  $(3, 6)$ ;
- la coppia  $(q_3, q_4)$  è distinguibile: la cella viene marcata;
- la coppia  $(q_2, q_5)$  è distinguibile se lo sono le coppie  $(q_2, q_4)$  e  $(q_3, q_6)$ :  $(2, 5)$  è inserito nelle liste delle celle  $(2, 4)$  e  $(3, 6)$ ;
- la coppia  $(q_3, q_5)$  è distinguibile: la cella viene marcata;
- la coppia  $(q_4, q_5)$  non è distinguibile;
- la coppia  $(q_2, q_6)$  è distinguibile: la cella viene marcata;
- la coppia  $(q_3, q_6)$  è distinguibile se lo è la coppia  $(q_0, q_1)$ :  $(3, 6)$  è inserito nella lista della cella  $(0, 1)$ ;
- la coppia  $(q_4, q_6)$  è distinguibile: la cella viene marcata;
- la coppia  $(q_5, q_6)$  è distinguibile: la cella viene marcata;

#### Minimizzazione di automi a stati finiti

1	(3,6)					
2	x	x				
3	x	x	x			
4	x	x	(2,5)	x		
5	x	x	(0,1)	x		
6	x	x	x	(2,4) (2,5)	x	x
	0	1	2	3	4	5

#### Minimizzazione di automi a stati finiti

Mantenendo le sole indicazioni di coppie distinguibili e non.

1						
2	x	x				
3	x	x	x			
4	x	x		x		
5	x	x		x		
6	x	x	x		x	x
	0	1	2	3	4	5

#### Minimizzazione di automi a stati finiti

Dalla tabella, risulta che gli insiemi di stati indistinguibili sono:

$q_0, q_1$

$q_3, q_6$

$q_2, q_4, q_5$

#### Minimizzazione di automi a stati finiti

L'automata equivalente col minimo numero di stati risultante dal procedimento è mostrato sotto: lo stato  $q_0$  corrisponde alla classe d'equivalenza  $\{q_0, q_1\}$ , lo stato  $q_1$  corrisponde alla classe  $\{q_3, q_6\}$  e lo stato  $q_2$ , infine, corrisponde alla classe  $\{q_2, q_4, q_5\}$ .

