

Esercizi di Informatica Teorica

Linguaggi regolari: automi a stati finiti

a cura di

Luca Cabibbo e Walter Didimo

Sommario

- automi a stati finiti
- automi a stati finiti non deterministici
- automi e grammatiche regolari

notazioni sul livello degli esercizi: (*) facile, (**) non difficile
(***) media complessità, (****) difficile, (*****) quasi impossibile

Automa a stati finiti (ASF)

automa a stati finiti (ASF) : $A = \langle \Sigma, K, \delta, q_0, F \rangle$ dove

- $\Sigma = \{ \sigma_1, \sigma_2, \dots, \sigma_n \}$ è un alfabeto (finito) di input
- $K = \{ q_0, q_1, \dots, q_m \}$ è un insieme (finito e non vuoto) di stati
- q_0 è lo stato iniziale
- $F \subseteq K$ è l'insieme degli stati finali
- $\delta : K \times \Sigma \rightarrow K$ è la funzione (totale) di transizione; si rappresenta graficamente attraverso il “grafo di transizione” o “diagramma di stato”

Linguaggio riconosciuto da un ASF

si definisce ricorsivamente la funzione di transizione estesa alle stringhe $\underline{\delta} : K \times \Sigma^* \rightarrow K$ al modo:

- $\underline{\delta}(q, \varepsilon) = q$
- $\underline{\delta}(q, aw) = \underline{\delta}(\delta(q, a), w)$ dove $a \in \Sigma$ e $w \in \Sigma^*$

linguaggio riconosciuto da $A : L = \{w \in \Sigma^* : \underline{\delta}(q_0, w) \in F\}$

Configurazioni e computazioni di un ASF

configurazione (istantanea) di un ASF: $\langle q, w \rangle$ dove

- q è lo **stato corrente** dell'ASF
- w è la porzione di stringa che l'ASF deve ancora leggere

transizione tra configurazioni: $\langle q, w \rangle \vdash \langle q', w' \rangle \Leftrightarrow$

- $w = aw'$ $a \in \Sigma$
- $\delta(q, a) = q'$

una configurazione $\langle q, w \rangle$ è:

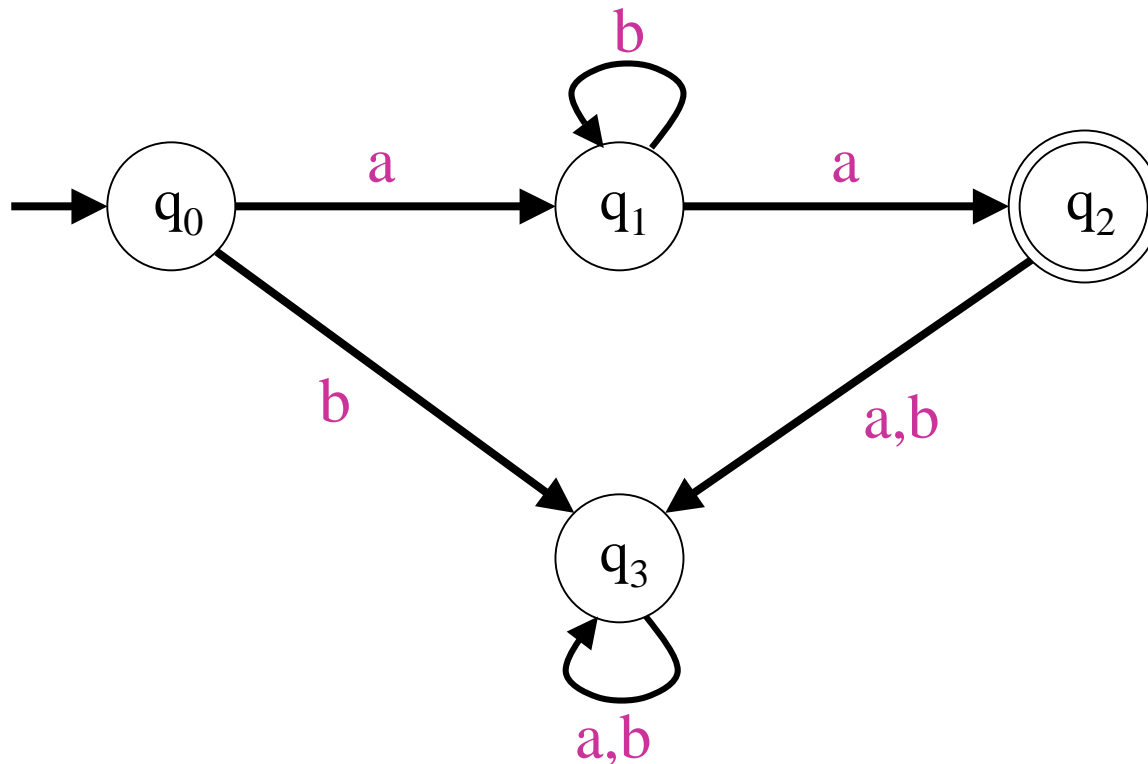
- **iniziale** se $q = q_0$
- **finale** se $w = \varepsilon$
- **accettante** se $w = \varepsilon$ e $q \in F$

computazione: \vdash^* è la chiusura transitiva e riflessiva di \vdash

computazione accettante: $c_0 \vdash^* c_n$ dove c_0 è iniziale e c_n è finale

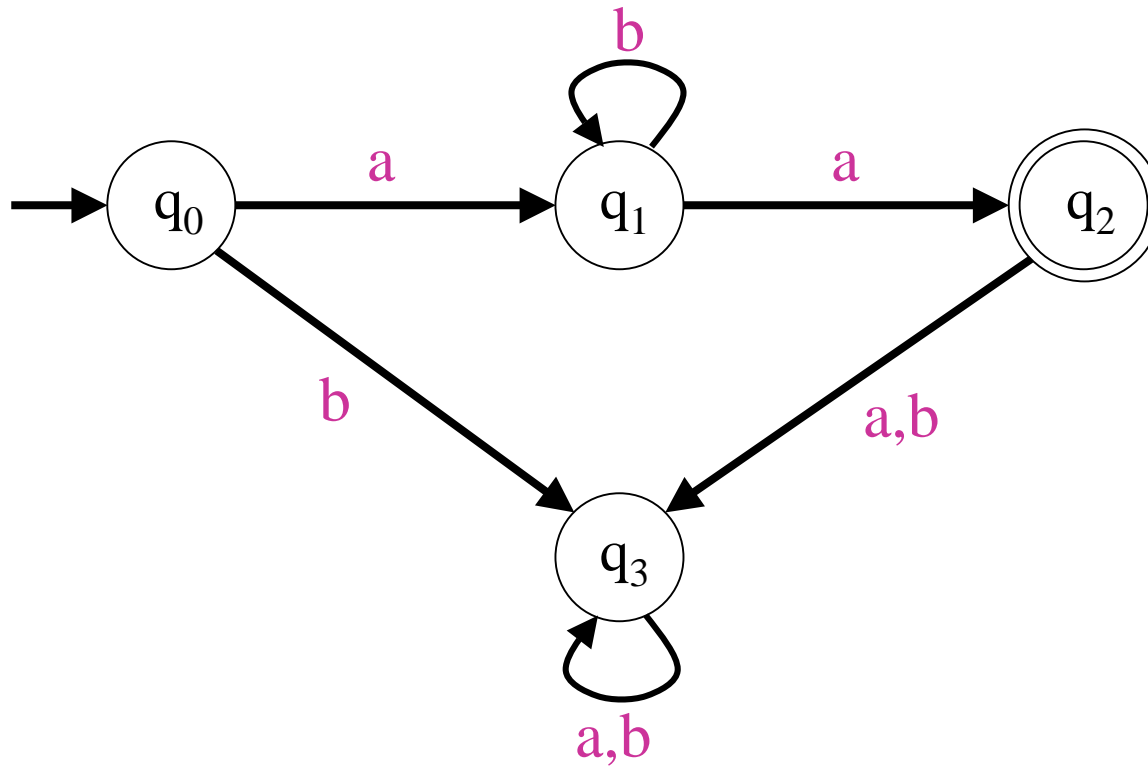
Esercizi svolti sugli ASF

Esercizio 1(**) dire qual'è il linguaggio riconosciuto dal seguente ASF e scrivere la corrispondente espressione regolare



Esercizi svolti sugli ASF

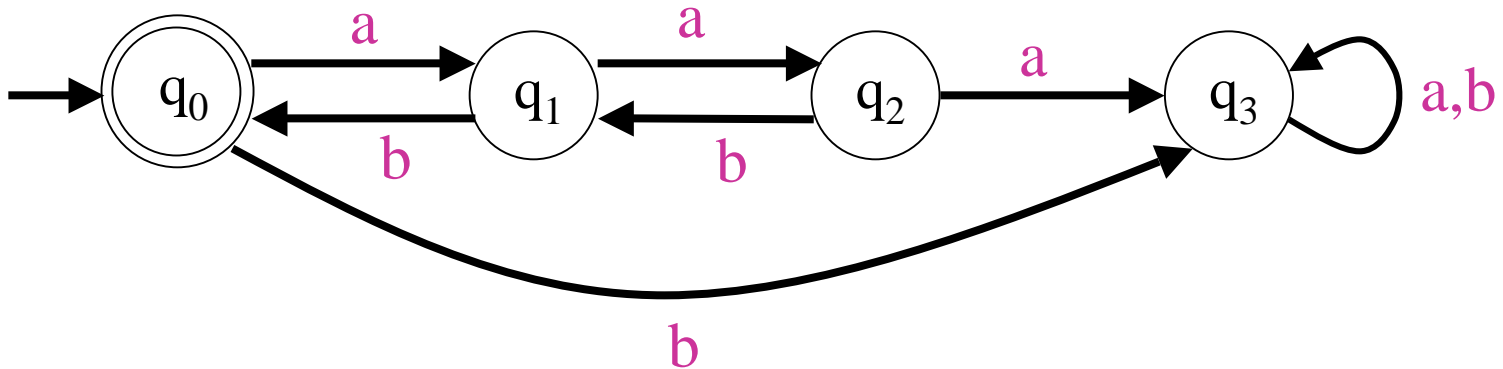
Soluzione $L = \{ab^n a : n \geq 0\}$ cioè $a b^* a$



Esercizi svolti sugli ASF

Esercizio 2(***) si consideri il seguente AFS:

- mostrare le computazioni sulle stringhe “aaab” e “abaabb”
- dire qual’è il linguaggio riconosciuto dall’automa
- descrivere il linguaggio attraverso una espressione regolare



Esercizi svolti sugli ASF

Soluzione

- computazioni sulle stringhe “aaab” e “abaabb”

$\langle q_0, aaab \rangle \vdash \langle q_1, aab \rangle \vdash \langle q_2, ab \rangle \vdash \langle q_3, b \rangle \vdash \langle q_3, \epsilon \rangle$ (non accettante)

$\langle q_0, abaabb \rangle \vdash \langle q_1, baabb \rangle \vdash \langle q_0, aabb \rangle \vdash \langle q_1, abb \rangle \vdash \langle q_2, bb \rangle \vdash \langle q_1, b \rangle \vdash \langle q_0, \epsilon \rangle$ (accettante)

- **linguaggio riconosciuto dall’automa:** stringhe su $\{a,b\}$ tali che:
 - numero di ‘a’ = numero di ‘b’
 - sottosequenze massimali di sole ‘a’ o di sole ‘b’ di lunghezza al più 2
 - iniziano per ‘a’ e finiscono per ‘b’

più la stringa vuota

- **espressione regolare:** $(a(ab)^*b)^*$

Esercizi svolti sugli ASF

Esercizio 3(***) costruire un AFS che riconosce il linguaggio dei numeri naturali pari in base 3, compresa la stringa vuota; si modifichi poi l'automa in modo che non accetti la stringa vuota.

Soluzione

• AFS che riconosce il linguaggio dei numeri naturali in base 3, compresa la stringa vuota

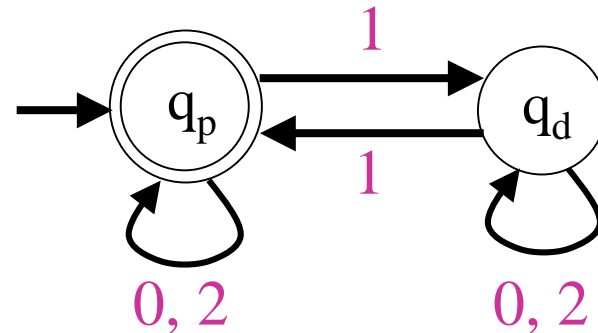
$$\Sigma = \{0, 1, 2\}$$

$$K = \{q_p, q_d\}$$

$$F = \{q_p\}$$

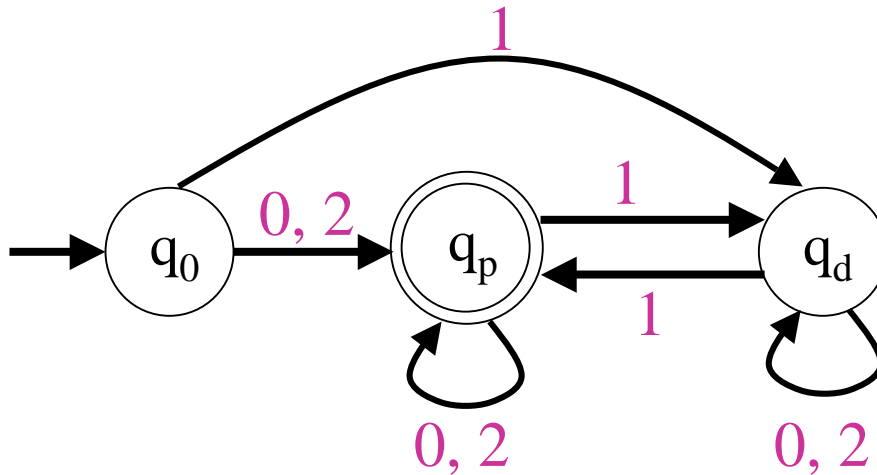
$$q_0 = q_p$$

δ	0	1	2
q_p	q_p	q_d	q_p
q_d	q_d	q_p	q_d



Esercizi svolti sugli ASF

AFS che riconosce il linguaggio dei numeri naturali pari in base 3, esclusa la stringa vuota



esercizio scrivere la funzione di transizione in forma tabellare per questo automa

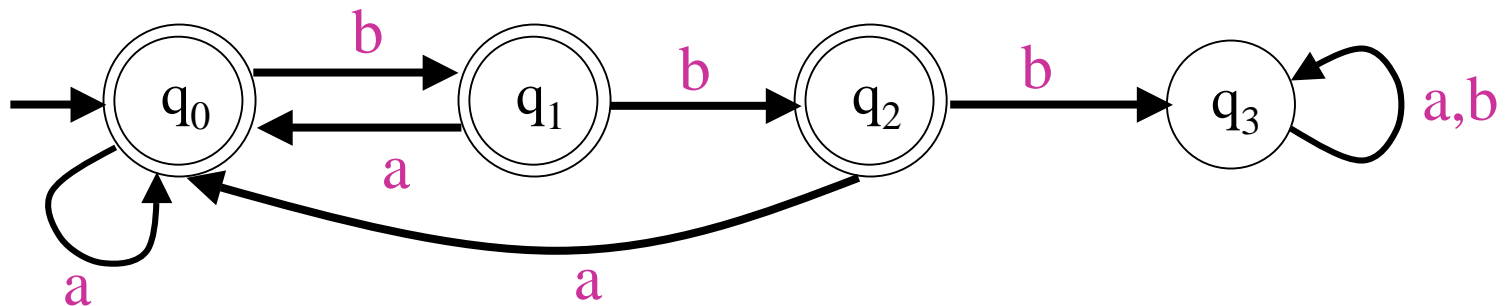
Esercizi svolti sugli ASF

Esercizio 4(***) per ciascuno dei seguenti linguaggi costruire un ASF che lo riconosca.

- $L1 = \{w \in \{a,b\}^* : w \text{ non contiene mai tre 'b' consecutive}\}$
- $L2 = \{w \in \{a,b\}^* : w \text{ contiene tre 'b' consecutive}\}$
- $L3 = \{w \in \{a,b\}^* : w \text{ contiene almeno tre 'b'}\}$

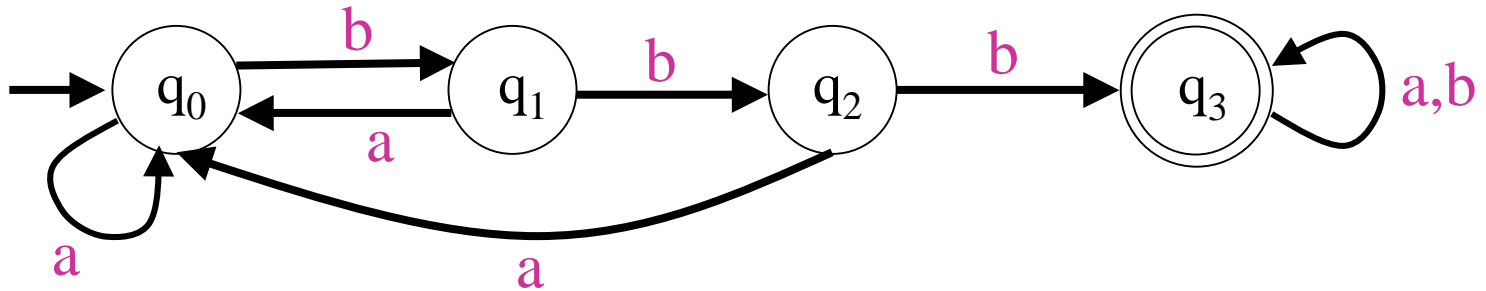
Soluzione

- ASF che riconosce L1

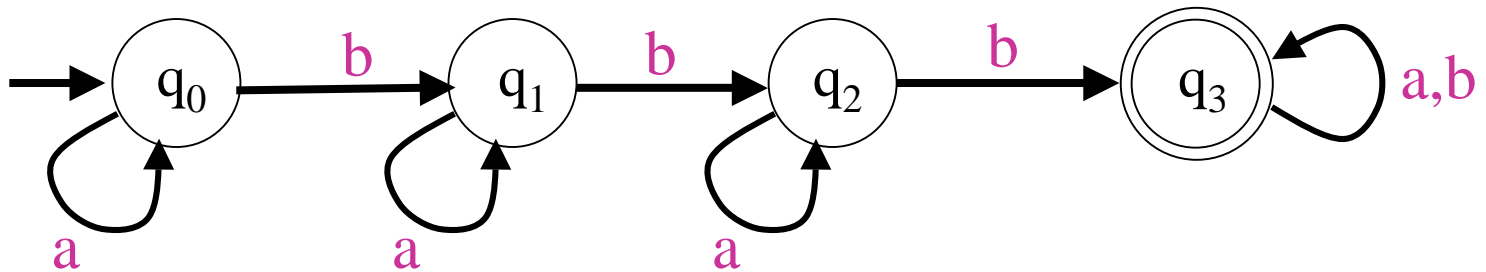


Esercizi svolti sugli ASF

- ASF che riconosce L2



- ASF che riconosce L3



Automa a stati finiti non deterministico (ASFND)

automa a stati finiti non deterministico (ASFND) :

$A = \langle \Sigma, K, \delta_N, q_0, F \rangle$ dove

- $\Sigma = \{ \sigma_1, \sigma_2, \dots, \sigma_n \}$ è un alfabeto (finito) di input
- $K = \{ q_0, q_1, \dots, q_m \}$ è un insieme (finito e non vuoto) di stati
- q_0 è lo stato iniziale
- $F \subseteq K$ è l'insieme degli stati finali
- $\delta_N : K \times \Sigma \rightarrow P(K)$ è la funzione (totale) di transizione;

Linguaggio riconosciuto da un ASFND

si definisce la funzione di transizione estesa alle stringhe

$\underline{\delta}_N : K \times \Sigma^* \rightarrow P(K)$ al modo:

- $\underline{\delta}_N(q, \varepsilon) = q$

- $\underline{\delta}_N(q, aw) = \bigcup_{p \in \delta_N(q, a)} \underline{\delta}_N(p, w)$

linguaggio riconosciuto da $A : L = \{w \in \Sigma^* : \underline{\delta}_N(q_0, w) \cap F \neq \emptyset\}$

Configurazioni e computazioni di un ASFND

configurazione (istantanea) di un ASFND: $\langle Q, w \rangle$ dove

- $Q \subseteq K$ è l'insieme degli **stati correnti** dell'ASFND
- w è la porzione di stringa che l'ASFND deve ancora leggere

transizione tra configurazioni: $\langle Q, w \rangle \vdash \langle Q', w' \rangle \Leftrightarrow$

- $w = a w' \quad a \in \Sigma$
- $Q' = \bigcup_{q \in Q} \delta_N(q, a)$

una configurazione $\langle Q, w \rangle$ è:

- **iniziale** se $Q = \{q_0\}$
- **finale** se $w = \varepsilon$
- **accettante** se $w = \varepsilon$ e $Q \cap F \neq \emptyset$

computazione: \vdash^* è la chiusura transitiva e riflessiva di \vdash

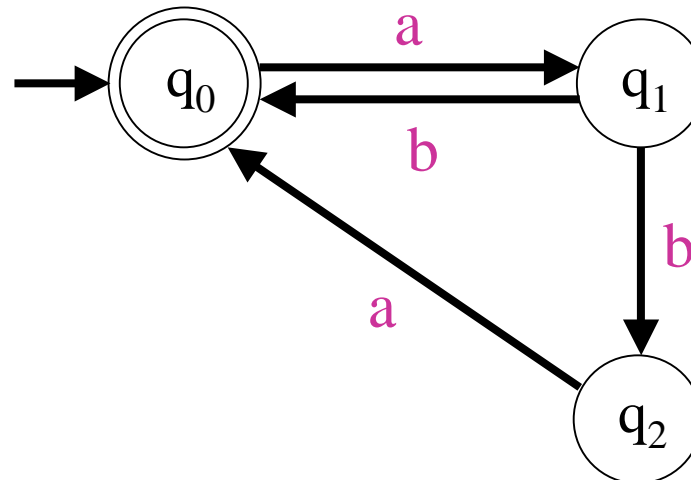
computazione accettante: $c_0 \vdash^* c_n$ dove c_0 è iniziale e c_n è finale

Esercizi svolti sugli ASFND

Esercizio 5(**) costruire un ASFND che riconosce il linguaggio descritto dall'espressione regolare $(ab + aba)^*$

Soluzione

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



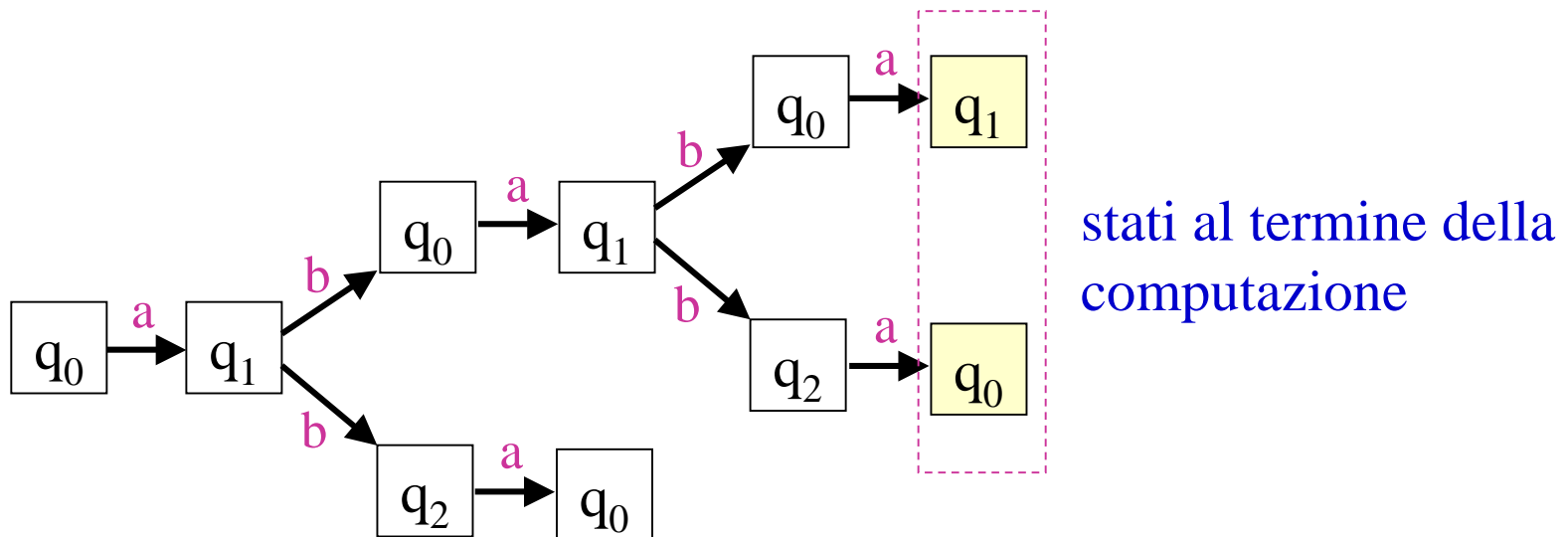
esercizio costruire un ASF che riconosce lo stesso linguaggio

Esercizi svolti sugli ASFND

esempio di computazione sulla stringa “ababa”

$\langle \{q_0\}, ababa \rangle \vdash \langle \{q_1\}, baba \rangle \vdash \langle \{q_0, q_2\}, aba \rangle \vdash \langle \{q_0, q_1\}, ba \rangle \vdash$
 $\langle \{q_0, q_2\}, a \rangle \vdash \langle \{q_0, q_1\}, \varepsilon \rangle$

albero delle transizioni per la stringa “ababa”

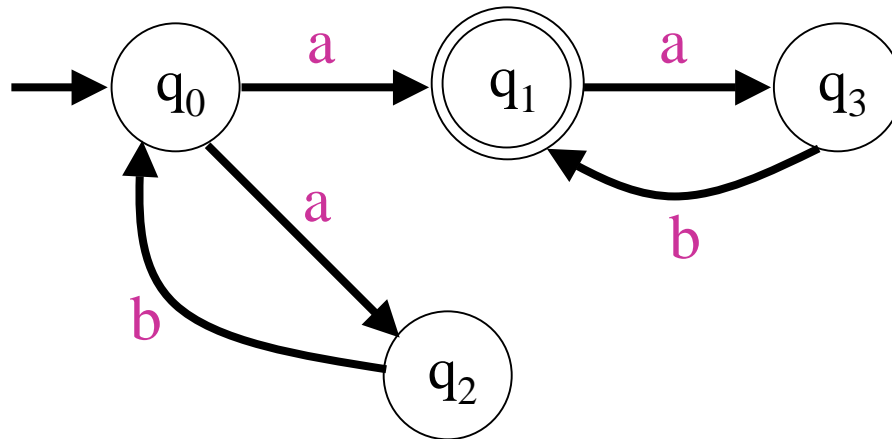


Esercizi svolti sugli ASFND

Esercizio 6(***) costruire un ASFND che riconosce il linguaggio
 $L = (ab)^*a(ab)^*$

Soluzione

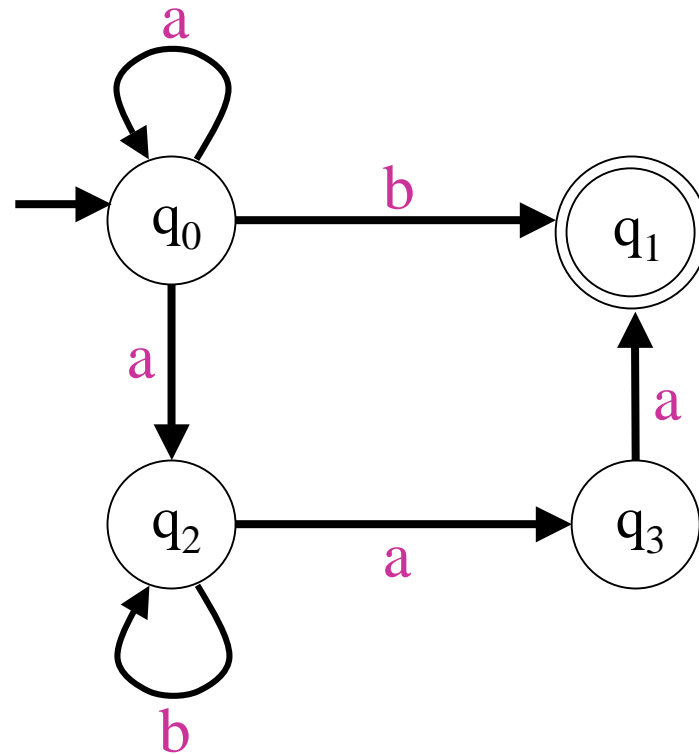
δ_N	a	b
q_0	$\{q_1, q_2\}$	\emptyset
q_1	$\{q_3\}$	\emptyset
q_2	\emptyset	$\{q_0\}$
q_3	\emptyset	$\{q_1\}$



esercizio mostrare la computazione sulla stringa “ababaab”

Esercizi svolti sugli ASFND

Esercizio 7(**) quali stringhe tra “aabb”, “ab”, “abbaa” ed “aabbaa” sono riconosciute dal seguente ASFND?; quale linguaggio riconosce?



Soluzione $L = (a^*b + a^*ab^*aa) = a^* (b + ab^*aa)$

Esercizi svolti sugli ASFND

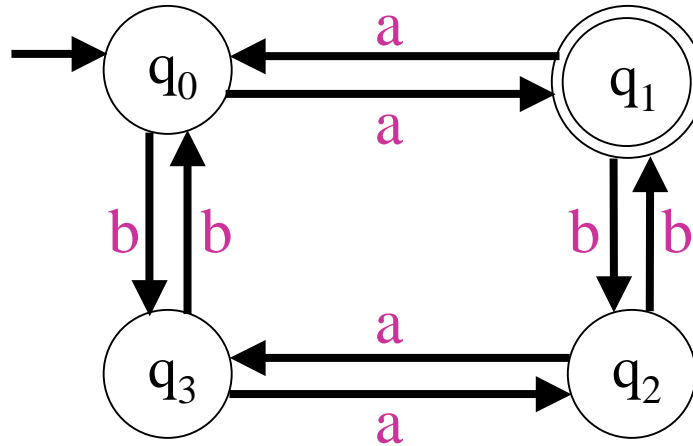
Esercizio 8(***) costruire un ASFND che riconosce il linguaggio delle stringhe su $\{a,b\}$ con un numero dispari di 'a' e un numero pari di 'b'

Soluzione

logica costruttiva:

- si usano quattro stati con i seguenti significati q_0 = pari 'a' e pari 'b', q_1 = dispari 'a' e pari 'b', q_2 = dispari 'a' e dispari 'b', q_3 = pari 'a' e dispari 'b';
- si costruisce la funzione di transizione, osservando che da ciascuno stato si può passare direttamente solo a stati adiacenti;
- si decidono gli stati accettanti sulla base della classificazione fatta e delle stringhe che si vogliono riconoscere

Esercizi svolti sugli ASFND



esercizio modificare l'automa in modo che riconosca le stringhe con un numero dispari di 'a' o un numero pari di 'b'

Esercizi da svolgere sugli automi

Esercizio 9(***) costruire degli automi a stati finiti (deterministici o non deterministici) che riconoscono i seguenti linguaggi:

1) $L = (ab^*a)^*$

2) $L = (ab^*a^*b)^*$

3) $L = a^*b^*(aa + bb)$

4) $L = a(bc)^*a$

5) stringhe su $\{a,b\}$ terminanti con “baa” o con “abb”

6) stringhe su $\{a,b\}$ terminanti con un numero dispari di “a”

Algoritmo: ASFND \rightarrow ASF

input: un ASFND $\langle \Sigma, K, \delta_N, q_0, F \rangle$

output: un ASF $\langle \Sigma', K', \delta', q'_0, F' \rangle$

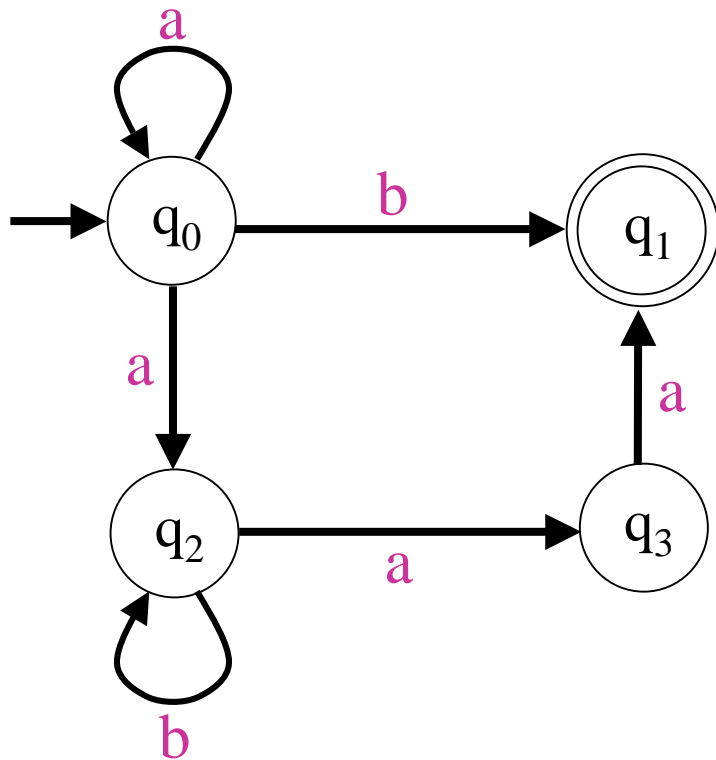
costruzione:

- $\Sigma' = \Sigma$
- K' = contiene un superstato $[q_i \dots q_j]$ per ciascun elemento $\{q_i, \dots, q_j\}$ di $P(K)$
- $q'_0 = [q_0]$
- $F' \subseteq K'$ è l'insieme dei superstati che contengono almeno uno stato di F
- $\delta'([q_i \dots q_j], a) = [q_h \dots q_k]$ dove $\{q_h \dots q_k\} = \delta_N(q_i, a) \cup \dots \cup \delta_N(q_j, a)$

semplificazione: per costruire K' si considerano solo superstati raggiungibili a partire dal superstato $[q_0]$

Esercizi svolti su ASFND \rightarrow ASF

Esercizio 10(***) costruire un ASF che riconosce lo stesso linguaggio del seguente ASFND:



δ_N	a	b
q ₀	{q ₀ ,q ₂ }	{q ₁ }
q ₁	\emptyset	\emptyset
q ₂	{q ₃ }	{q ₂ }
q ₃	{q ₁ }	\emptyset

Esercizi svolti su ASFND \rightarrow ASF

Soluzione

- costruzione incrementale della funzione di transizione δ dell'ASF

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

- $\delta([q_0], a) = [q_0 q_2]$ $\delta([q_0], b) = [q_1]$
- $\delta([q_0 q_2], a) = [q_0 q_2 q_3]$ $\delta([q_0 q_2], b) = [q_1 q_2]$
- $\delta([q_1], a) = []$ $\delta([q_1], b) = []$
- $\delta([q_0 q_2 q_3], a) = [q_0 q_1 q_2 q_3]$ $\delta([q_0 q_2 q_3], b) = [q_1 q_2]$
- $\delta([q_1 q_2], a) = [q_3]$ $\delta([q_1 q_2], b) = [q_2]$

- $\delta([q_0 q_1 q_2 q_3], a) = [q_0 q_1 q_2 q_3]$ $\delta([q_0 q_1 q_2 q_3], b) = [q_1 q_2]$
- $\delta([q_3], a) = [q_1]$ $\delta([q_3], b) = []$
- $\delta([q_2], a) = [q_3]$ $\delta([q_2], b) = [q_2]$

Esercizi da svolgere su ASFND \rightarrow ASF

Esercizio 11(***) utilizzando l'algoritmo ASFND \rightarrow ASF, costruire degli ASF che riconoscono gli stessi linguaggi dei seguenti automi:

- AFSND dell'esercizio 5
- AFSND dell'esercizio 6
- AFSND dell'esercizio 8

Algoritmo: ASFND \rightarrow grammatica regolare

input: un ASFND $\langle \Sigma, K, \delta_N, q_0, F \rangle$ (o un ASF $\langle \Sigma, K, \delta, q_0, F \rangle$)

output: una grammatica regolare $\langle V_T, V_N, P, S \rangle$

costruzione:

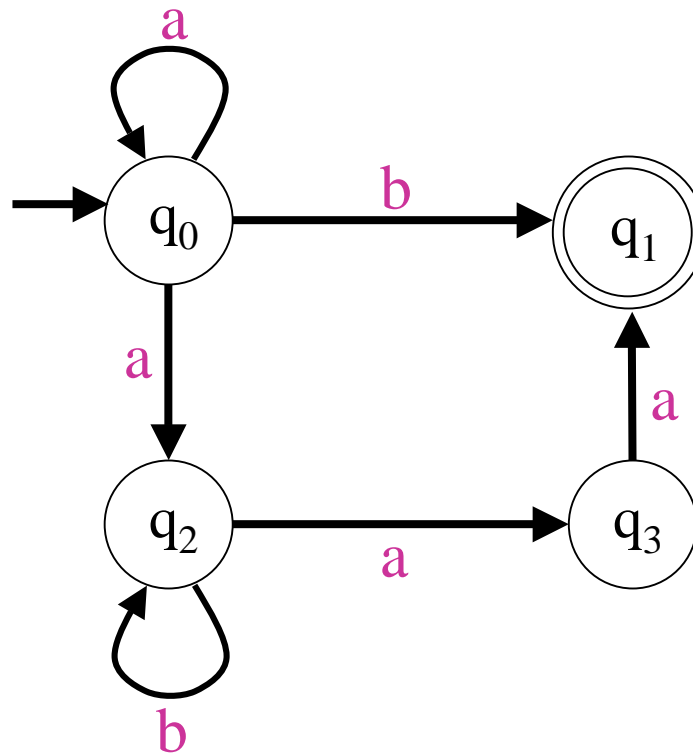
- $V_T = \Sigma$
- V_N contiene un non terminale A_i per ogni stato $q_i \in K$; se q_0 è uno stato finale, si aggiunge un ulteriore non terminale A'_0
- $S = A_0$ se q_0 non è uno stato finale, altrimenti $S = A'_0$
- P contiene le seguenti **produzioni** $\forall q_k \in \delta_N(q_i, a)$ (o se $\delta(q_i, a) = q_k$):
 - $A_i \rightarrow aA_k$
 - $A_i \rightarrow a$ se q_k è uno stato finale

inoltre, se q_0 è uno stato finale, P contiene le seguenti **produzioni**:

- $A'_0 \rightarrow \epsilon$
- $A'_0 \rightarrow aA_k \quad \forall A_0 \rightarrow aA_k \quad A'_0 \rightarrow a \quad \forall A_0 \rightarrow a$

Esercizi svolti su :ASFND \rightarrow grammatica regolare

Esercizio 12(**) determinare una grammatica regolare equivalente al seguente ASFND



Esercizi svolti su :ASFND \rightarrow grammatica regolare

Soluzione $V_T = \{a, b\}$ $V_N = \{A_0, A_1, A_2, A_3\}$ $S = A_0$

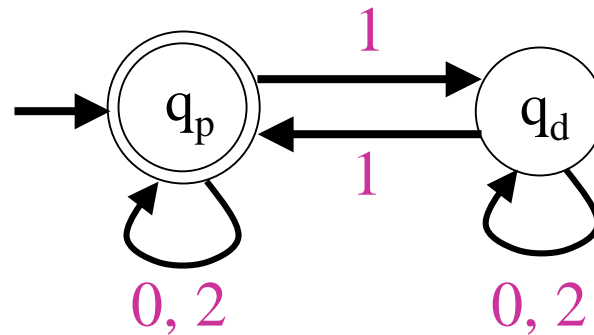
insieme P delle produzioni

- produzioni per A_0 : $A_0 \rightarrow aA_0$ $A_0 \rightarrow aA_2$ $A_0 \rightarrow bA_1$ $A_0 \rightarrow b$
- produzioni per A_1 : nessuna
- produzioni per A_2 : $A_2 \rightarrow aA_3$ $A_2 \rightarrow bA_2$
- produzioni per A_3 : $A_3 \rightarrow aA_1$ $A_3 \rightarrow a$

osservazione: poiché A_1 non ha produzioni, le produzioni $A_0 \rightarrow bA_1$ e $A_3 \rightarrow aA_1$ diventano inutili per la grammatica, e possono quindi essere tolte

Esercizi svolti su :ASFND \rightarrow grammatica regolare

Esercizio 13(**) determinare una grammatica regolare equivalente al seguente ASFND



Soluzione $V_T = \{0, 1, 2\}$ $V_N = \{A_p, A_d, A'\}$ $S = A'$

insieme P delle produzioni

- produzioni per A_p : $A_p \rightarrow 0A_p$ $A_p \rightarrow 2A_p$ $A_p \rightarrow 1A_d$ $A_p \rightarrow 0$ $A_p \rightarrow 2$
- produzioni per A_d : $A_d \rightarrow 0A_d$ $A_d \rightarrow 2A_d$ $A_d \rightarrow 1A_p$ $A_d \rightarrow 1$
- produzioni per A' : $A' \rightarrow 0A_p$ $A' \rightarrow 2A_p$ $A' \rightarrow 1A_d$ $A' \rightarrow 0$ $A' \rightarrow 2$
 $A' \rightarrow \epsilon$

Algoritmo: grammatica regolare \rightarrow ASFND

input: una grammatica regolare $\langle V_T, V_N, P, S \rangle$

output: un ASFND $\langle \Sigma, K, \delta_N, q_0, F \rangle$

costruzione:

- $\Sigma = V_T$
- K contiene uno stato q_A per ogni $A \in V_N$, più uno stato finale q_F
- $q_0 = q_S$
- F contiene q_F , più uno stato q_B per ogni ε -produzione $B \rightarrow \varepsilon$
- $\delta_N(q_B, a)$ contiene:
 - q_C se $B \rightarrow aC$
 - q_F se $B \rightarrow a$

Esercizi svolti su: grammatica regolare \rightarrow ASFND

Esercizio 14(*) dimostrare che per ogni linguaggio regolare L che non contiene la stringa vuota esiste un ASFND con un solo stato finale che riconosce L

Soluzione

se L non contiene la stringa vuota allora esiste per L una grammatica regolare G senza ϵ -produzioni; ma allora, applicando l'algoritmo che da G calcola un AFSND, ricaviamo un automa con un solo stato finale; tale automa, essendo equivalente a G , riconosce L

Esercizi svolti su: grammatica regolare \rightarrow ASFND

Esercizio 15(**) determinare un ASFND equivalente alla seguente grammatica regolare:

$V_T = \{a, b, c\}$ $V_N = \{S, A, C\}$ $S =$ assioma

insieme P delle produzioni:

- $S \rightarrow aA$ $S \rightarrow bC$
- $A \rightarrow aA$ $A \rightarrow bC$
- $C \rightarrow cC$ $C \rightarrow c$

dire inoltre qual'è il linguaggio riconosciuto dall'ASFND

Esercizi svolti su: grammatica regolare \rightarrow ASFND

Soluzione $\Sigma = \{a,b,c\}$, $K = \{q_S, q_A, q_C, q_F\}$, $F = \{q_F\}$

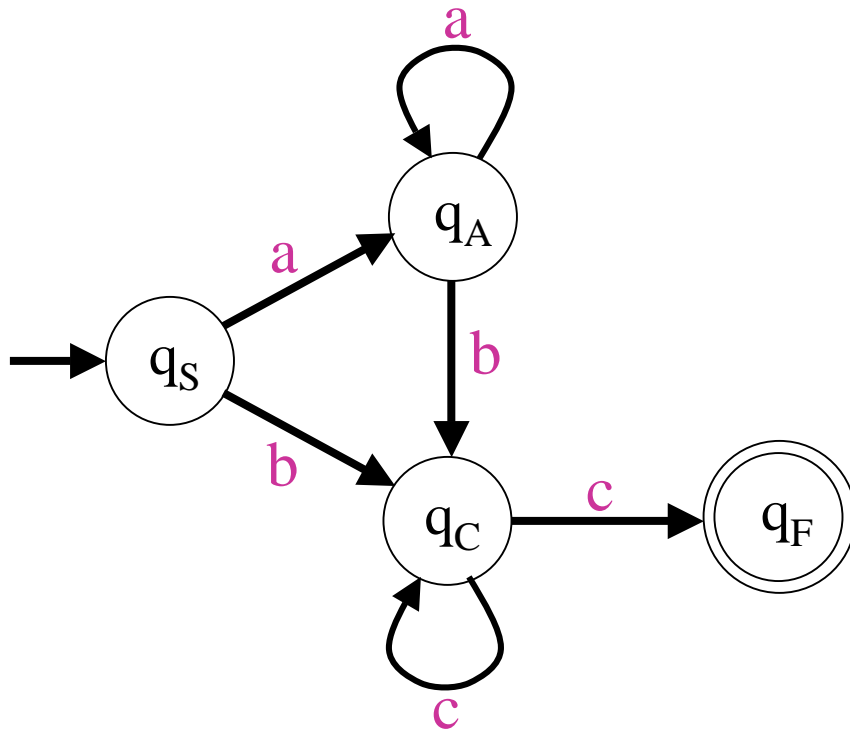
$$\delta_N(q_S, a) = \{q_A\}$$

$$\delta_N(q_S, b) = \{q_C\}$$

$$\delta_N(q_A, a) = \{q_A\}$$

$$\delta_N(q_A, b) = \{q_C\}$$

$$\delta_N(q_C, c) = \{q_C, q_F\}$$



$$L = (aa^*b + b) c^*c$$