Automi a stati finiti non deterministici (NFA)

Un Automa a stati finiti non deterministico (NFA) è definito dalla quintupla:
$A=<\Sigma,Q,\delta,q_0,F>$ dove
$\Sigma = \{\sigma_1,\ldots,\sigma_n\}$ alfabeto di input
$Q = \{q_0,\ldots,q_n\}$ insieme finito non vuoto di stati
$F \subseteq Q$ insieme di stati finali
$q_0 \in Q$ stato iniziale
$\delta: Q \times \Sigma \rightarrow P(Q)$ funzione di transizione, funzione totale che determina l'insieme (eventualmente vuoto) degli stati successivi

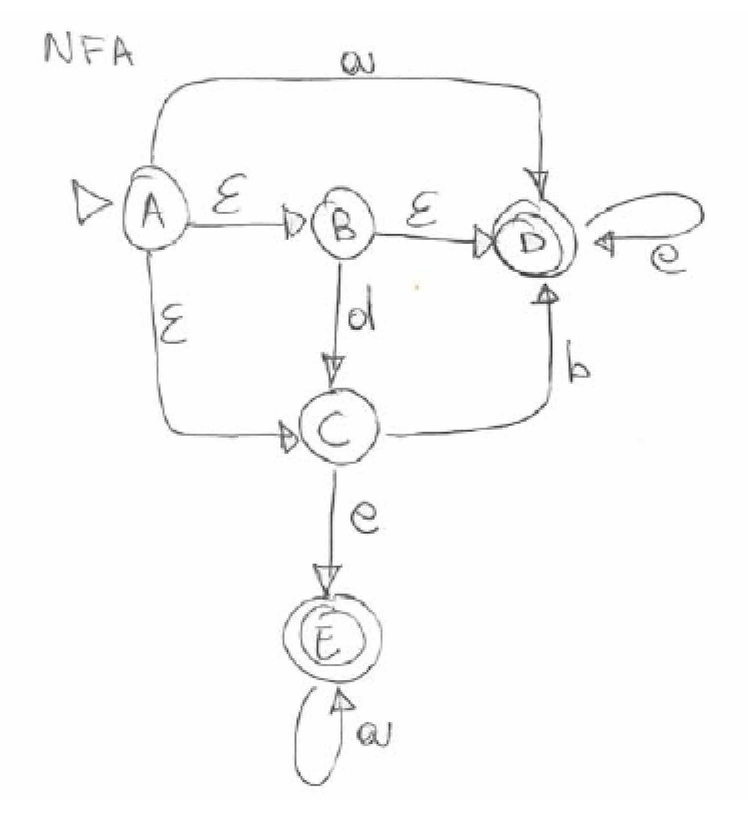
La **ε- chiusura di uno stato q** è l'insieme degli stati raggiungibili da q mediante zero, una o più transazioni ε.

Costruzione di un DFA da un NFA

1. Eliminazione ε-mosse
2. Costruzione DFA

ESERCIZIO 1

Costruire un DFA a partire dal seguente NFA:



ε- chiusura(A)={A,B,C,D}

ε- chiusura(B)={B,D}

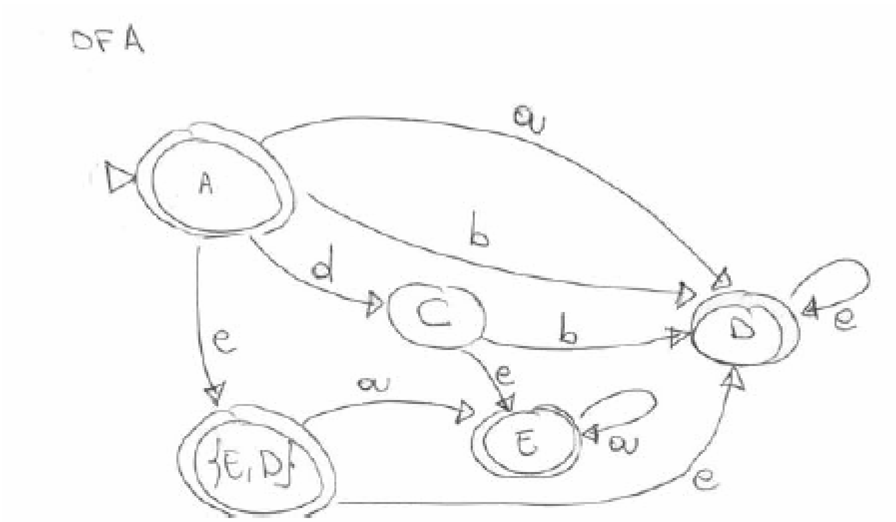
ε- chiusura(C)={C}

ε- chiusura(D)={D}

ε- chiusura(E)={E}

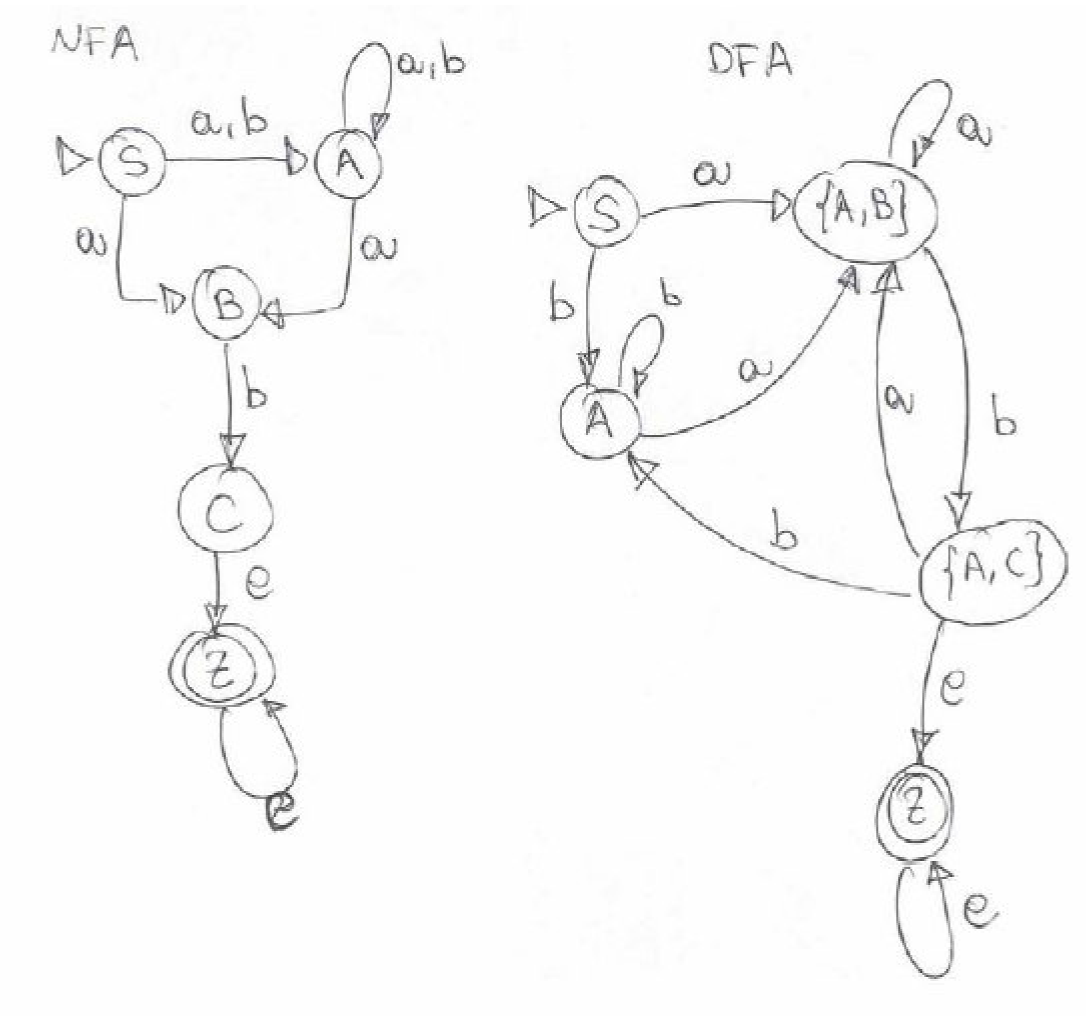
Eliminazione ε-mosse

- $\delta_N (A, a)= \delta^* (A, a)= \varepsilon$- chiusura($\delta (A, a) \cup \delta (B, a) \cup \delta (C, a) \cup \delta (D, a)$)={D}
- $\delta_N (A, b)= \delta^* (A, b)= \varepsilon$- chiusura($\delta (A, b) \cup \delta (B, b) \cup \delta (C, b) \cup \delta (D, b)$)={D}
- $\delta_N (A, c)= \delta^* (A, c)= \varepsilon$- chiusura($\delta (A, c) \cup \delta (B, c) \cup \delta (C, c) \cup \delta (D, c)$)={E,D}
- $\delta_N (A, d)= \delta^* (A, d)= \varepsilon$- chiusura($\delta (A, d) \cup \delta (B, d) \cup \delta (C, d) \cup \delta (D, d)$)={C}
- $\delta_N (B, a)= \delta^* (B, a)= \varepsilon$- chiusura($\delta (B, a) \cup \delta (D, a)$)={}
- $\delta_N (B, b)= \delta^* (B, b)= \varepsilon$- chiusura($\delta (B, b) \cup \delta (D, b)$)={}
- $\delta_N (B, c)= \delta^* (B, c)= \varepsilon$- chiusura($\delta (B, c) \cup \delta (D, c)$)={D}
- $\delta_N (B, d)= \delta^* (B, d)= \varepsilon$- chiusura($\delta (B, d) \cup \delta (D, d)$)={C}
- ….
-



ESERCIZIO 2

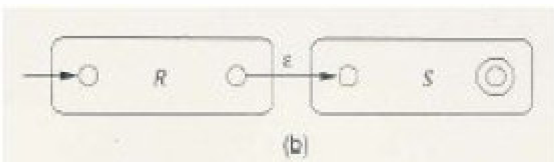
Costruire un DFA a partire dal seguente NFA:



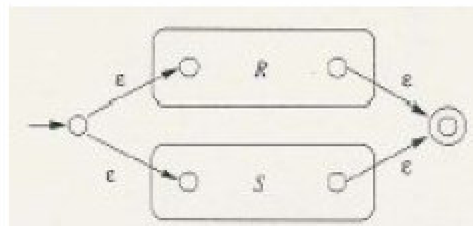
Da espressione regolare a ε- NFA

Un’espressione regolare può essere trasformata in un automa a stati finiti mediante una decomposizione in blocchi.
Sia *r* un’espressione regolare. Esiste un automa *M* che accetta tutte e solo le stringhe generate da *r*:

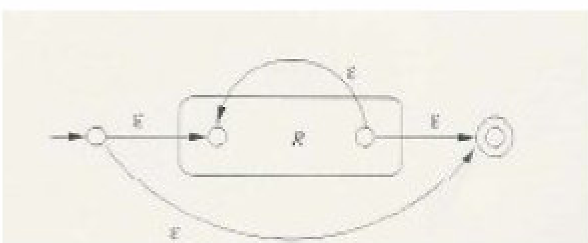
- si individuano i componenti di base che costituiscono *r*;
- si costruisce un NFA per ciascuno di essi;
- guidati dalla struttura sintattica di *r*, si combinano induttivamente i vari NFA prodotti al passo precedente.



E=RS



E=R|S



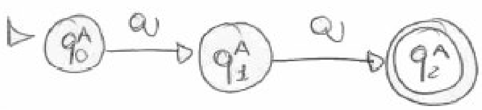
E=R*

ESERCIZIO 3

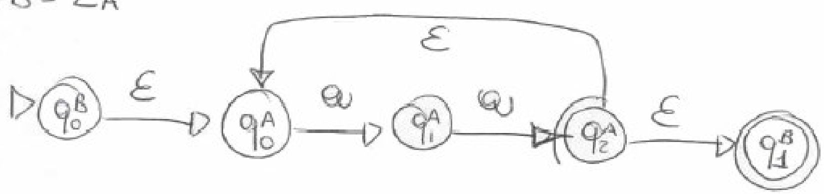
Dato il linguaggio L, descritto mediante espressione regolare, costruire un automa deterministico che riconosca lo stesso linguaggio.

$$L = ((aa)^+ (b^* | ae))^+$$

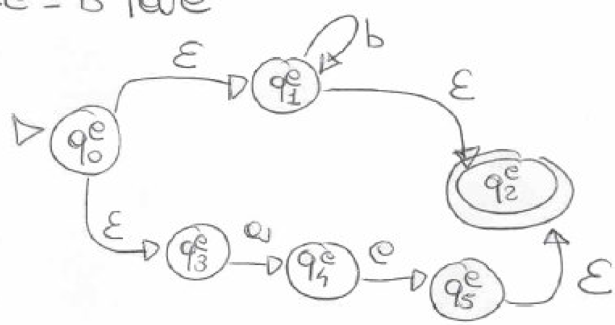
$$L_A = aa$$



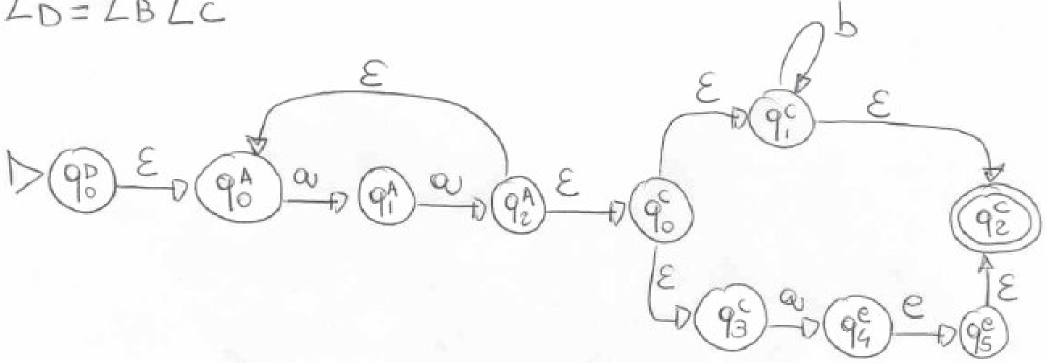
$$L_B = L_A^+$$

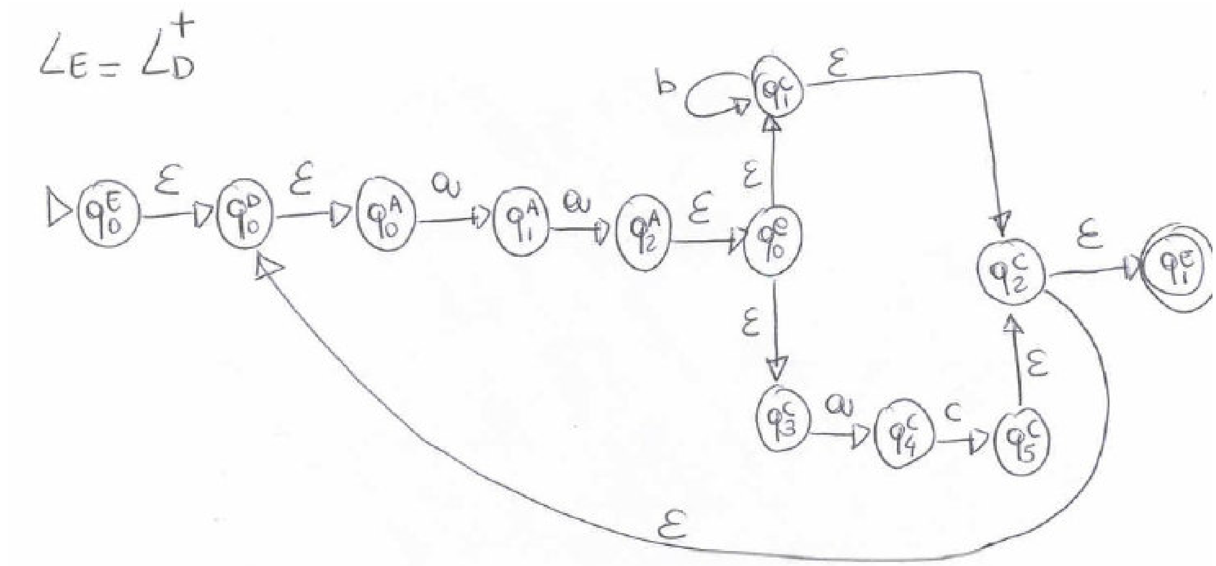


$$L_C = b^* | ae$$

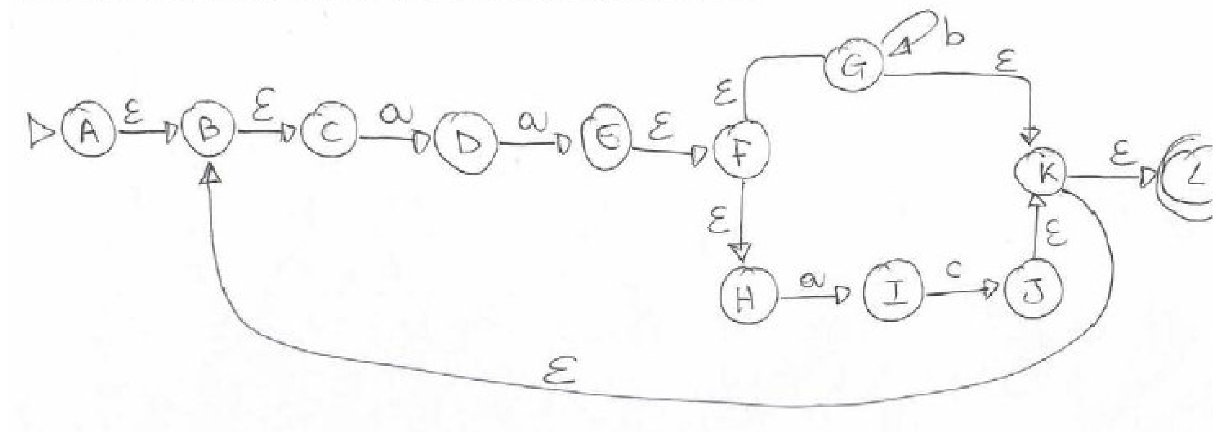


$$L_D = L_B L_C$$





Per comodità rinominiamo gli stati nel seguente modo:



Eliminazione ε -mosse:

ε - chiusura(A)={A,B,C}

- δ_N (A, a)= δ^* (A, a)= ε - chiusura(δ (A, a)$\cup$ δ (B, a) $\cup$ δ (C, a))={D}
- δ_N (A, b)= δ^* (A, b)= { }
- δ_N (A, c)= δ^* (A, c)= { }

ε - chiusura(B)={B,C}

- δ_N (B, a)= δ^* (B, a)= ε - chiusura(δ (B, a) $\cup$ δ (C, a))={D}
- δ_N (B, b)= δ^* (B, b)= { }
- δ_N (B, c)= δ^* (B, c)= { }

ε - chiusura(C)={C}

- δ_N (C, a)= δ^* (C, a)= ε - chiusura(δ (C, a))={D}
- δ_N (C, b)= δ^* (C, b)= { }
- δ_N (C, c)= δ^* (C, c)= { }

$\mathcal{E}$ - chiusura(D)={D}

- δ_N (D, a)= δ^* (D, a)= $\mathcal{E}$ - chiusura(δ (D, a))= $\mathcal{E}$ - chiusura(E)={E,F,G,K,L,B,H,C}
- δ_N (D, b)= δ^* (D, b)= { }
- δ_N (D, c)= δ^* (D, c)= { }

$\mathcal{E}$ - chiusura(E)={E,F,G,K,B,L,H,C}

- δ_N (E, a)= δ^* (E, a)= $\mathcal{E}$ - chiusura({D,I})= {D,I}
- δ_N (E, b)= δ^* (E, b)= $\mathcal{E}$ - chiusura({G})= {G,K,L,B,C}
- δ_N (E, c)= δ^* (E, c)= { }

$\mathcal{E}$ - chiusura(F)={F,G,K,L,B,C,H}

- δ_N (F, a)= δ^* (F, a)= $\mathcal{E}$ - chiusura({D,I})= {D,I}
- δ_N (F, b)= δ^* (F, b)= $\mathcal{E}$ - chiusura({G})= {G,K,L,B,C}
- δ_N (F, c)= δ^* (F, c)= { }

$\mathcal{E}$ - chiusura(G)={G,K,B,L,C}

- δ_N (G, a)= δ^* (G, a)= $\mathcal{E}$ - chiusura({D})= {D}
- δ_N (G, b)= δ^* (G, b)= $\mathcal{E}$ - chiusura({G})= {G,K,L,B,C}
- δ_N (G, c)= δ^* (G, c)= { }

$\mathcal{E}$ - chiusura(H)={H}

- δ_N (H, a)= δ^* (H, a)= $\mathcal{E}$ - chiusura({I})= { I}
- δ_N (H, b)= δ^* (H, b)= { }
- δ_N (H, c)= δ^* (H, c)= { }

$\mathcal{E}$ - chiusura(I)={I}

- δ_N (I, a)= δ^* (I, a)={ }
- δ_N (I, b)= δ^* (I, b)= { }
- δ_N (I, c)= δ^* (I, c)= $\mathcal{E}$ - chiusura({J})={J,K,L,B,C}

$\mathcal{E}$ - chiusura(J)={J,K,L,B,C}

- δ_N (J, a)= δ^* (J, a)= $\mathcal{E}$ - chiusura({D})={D}
- δ_N (J, b)= δ^* (J, b)= { }
- δ_N (J, c)= δ^* (J, c)= { }

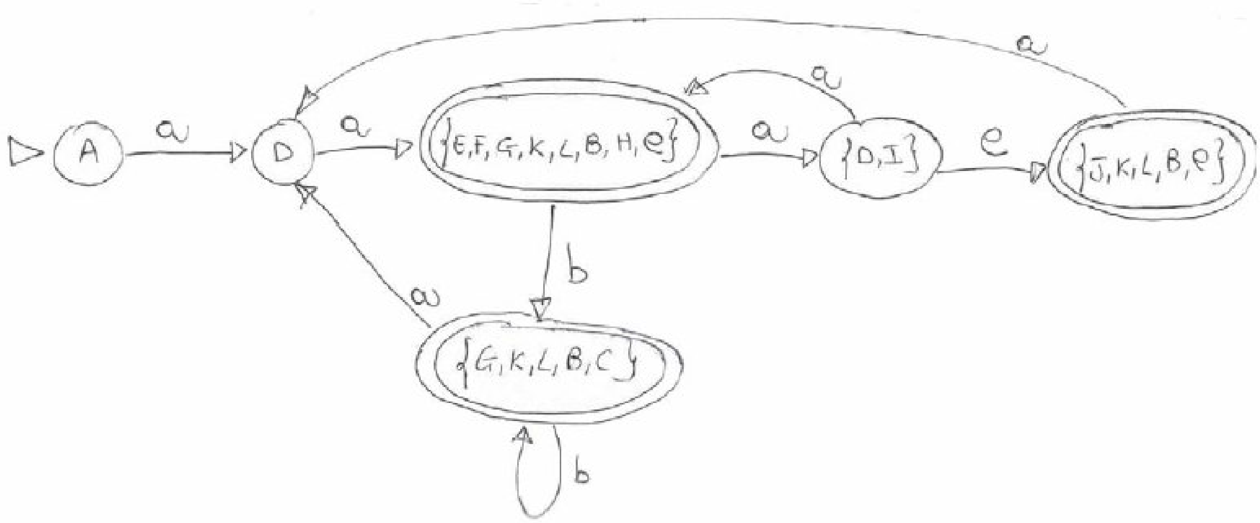
$\mathcal{E}$ - chiusura(K)={K,L,B,C}

- δ_N (K, a)= δ^* (K, a)= $\mathcal{E}$ - chiusura({D})={D}
- δ_N (K, b)= δ^* (K, b)= { }
- δ_N (K, c)= δ^* (K, c)= { }

La matrice di transizione dell'automa ottenuto dall'eliminazione delle ε -mosse è la seguente:

	a	b	c
→ A	{D}	{}	{}
B	{D}	{}	{}
C	{D}	{}	{}
D	{E,F,G,K,L,B,H,C}	{}	{}
E	{D,I}	{G,K,L,B,C}	{}
F	{D,I}	{G,K,L,B,C}	{}
G	{D}	{G,K,L,B,C}	{}
H	{I}	{}	{}
I	{}	{}	{J,K,L,B,C}
J	{D}	{}	{}
K	{D}	{}	{}
*L	{}	{}	{}

Costruiamo quindi il DFA:



Utilizzando questo automa è immediato scrivere il seguente metodo di riconoscimento in Java che restituisce true se la stringa *st* appartiene al linguaggio:

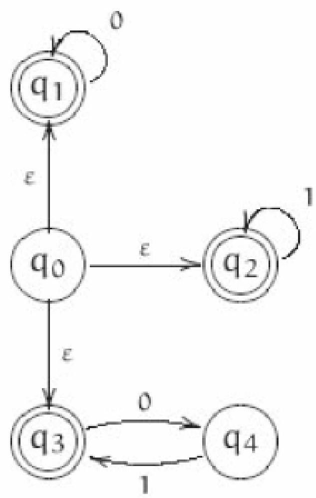
indichiamo con q0=A;q1=D;q2={E,F,G,K,L,B,H,C}:q3={G,K,L,B,C};q4={D,I};q5={J,K,L,B,C}

```
boolean riconosciStringa ( char [] st, int n ) {
    boolean errore = false; int q = 0; int i = 0;
    for (int i=0; i<n && !errore; i++)
        switch (q) {
            0: if ( st[i] == "a") q=1;
```

```
                else errore=true; break;
            1: if ( st[i] == "a" ) q=2;
               else errore=true; break;
            2: if (st[i] == "a") q=4;
                else if (st[i] == "b") q=3; else errore=true; break;
            3: if (st[i] == "a") q=1;
                else if (st[i] == "b") q=3; else errore=true; break;
            4: if (st[i] == "a") q=2;
                else if (st[i] == "c") q=5; else errore=true; break;
            5: if (st[i] == "a") q=1;
                else errore=true; break;
        }
    return !errore && (q==2 || q==3 || q==5)
}
```

ESERCIZIO 4

Si costruisca un automa deterministico per il linguaggio definito dalla seguente espressione regolare:
L=(0*+1*+(01)*)



$\mathcal{E}$ - chiusura(q_0)={q0,q1,q2,q3}

- δ_N (q0,0)= δ^* (q0,0)= $\mathcal{E}$ - chiusura(δ (q0,0)$\cup$ δ (q1,0) $\cup$ δ (q2, 0) $\cup$ δ (q3,0)) ={q1,q4}
- δ_N (q0,1)= δ^* (q0,1)= $\mathcal{E}$ - chiusura(q2)={q2}

$\mathcal{E}$ - chiusura(q1)={q1}

- δ_N (q1,0)= δ^* (q1,0)= $\mathcal{E}$ - chiusura(q1)={q1}
- δ_N (q1,1)= δ^* (q1,1)= { }

$\mathcal{E}$ - chiusura(q_2)={q2}

- $\delta_N(q2,0) = \delta^*(q2,0) = \{\}$
- $\delta_N(q2,1) = \delta^*(q2,1) = \varepsilon$ - chiusura(q2)={q2}

ε - chiusura(q_3)={q3}

- $\delta_N(q3,0) = \delta^*(q3,0) = \{q4\}$
- $\delta_N(q3,1) = \delta^*(q3,1) = \{\ \}$

ε - chiusura(q_4)={q4}

- $\delta_N(q4,0) = \delta^*(q4,0) = \{\ \}$
- $\delta_N(q4,1) = \delta^*(q4,1) = \{q3\}$

Tabella di transizione dell'automa indeterministico senza ε -mosse:

	0	1
→ *q0	{q1,q4}	{q2}
*q1	{q1}	{}
*q2	{}	{q2}
*q3	{q4}	{}
q4	{}	{q3}

Il DFA mostrato è quindi ottenuto sostituendo per comodità:
p0=q0; p1=q1; p2=q2; p3=q3; p4=q4; p5={q1,q4};