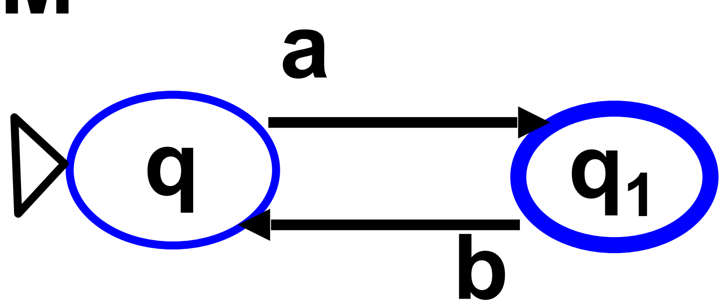
CURIOSITÀ

CURIOSITÀ

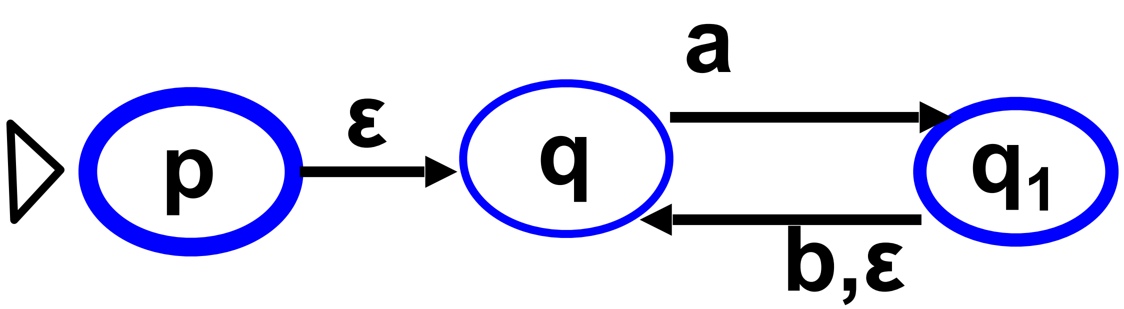
1. **Perchè è necessario aggiungere un nuovo stato nella costruzione dell'automa per la chiusura rispetto alla stella di Kleene?**

aggiunta nuovo stato per la *

M



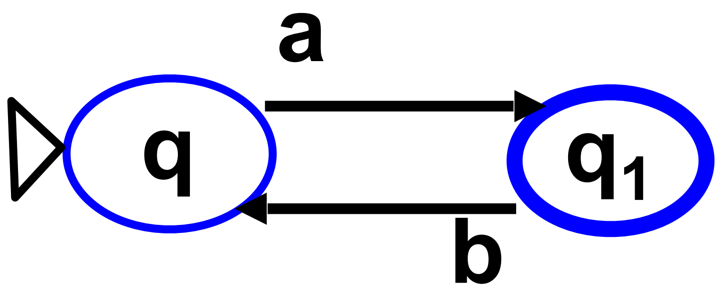
$L(M) = \{a\}\{ba\}^*$



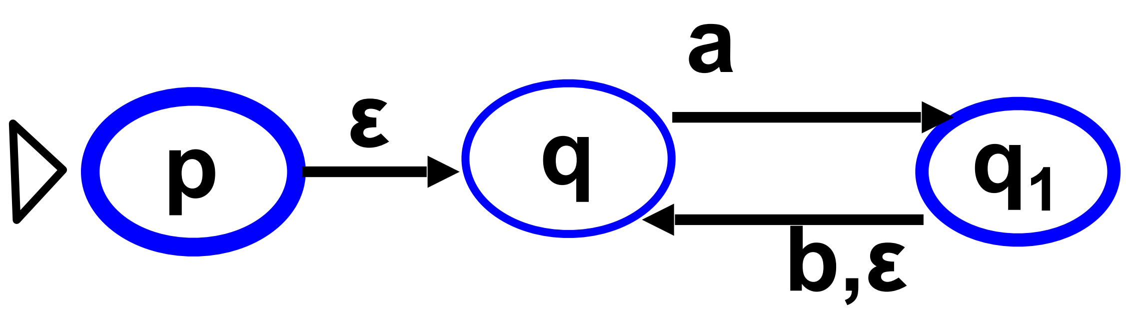
$L(M)^* = (\{a\}\{ba\}^*)^*$

aggiunta nuovo stato per la *

M

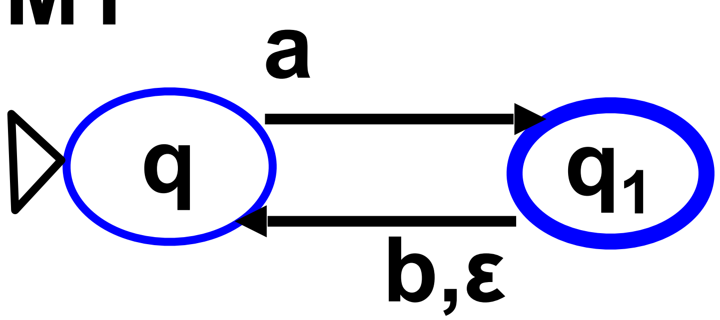


L(M) = {a}{ba}*



L(M)* = ({a}{ba}*)*

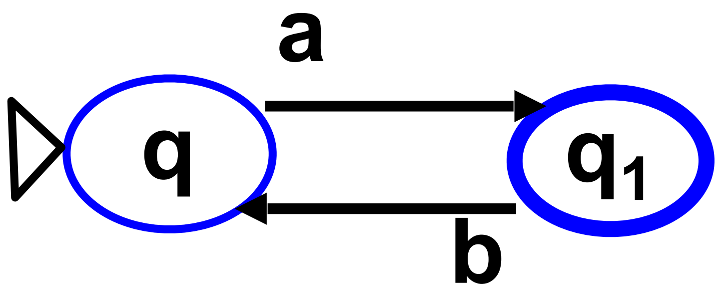
M1



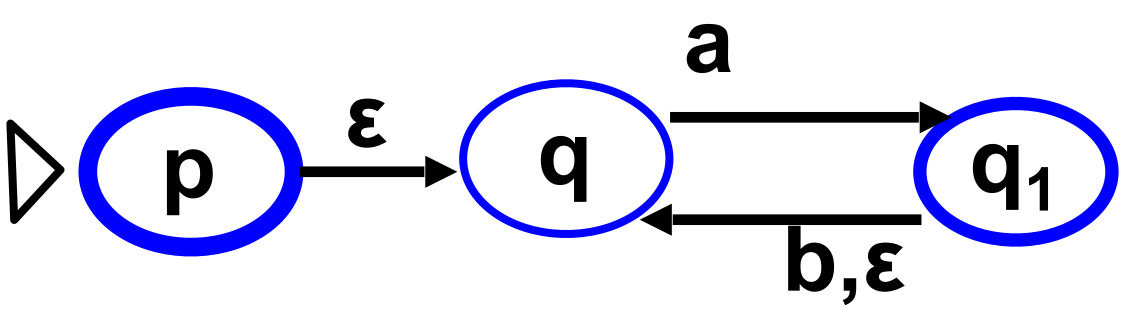
ma ε non è in L(M1),
quindi L(M1) ≠ L(M)*

aggiunta nuovo stato per la *

M

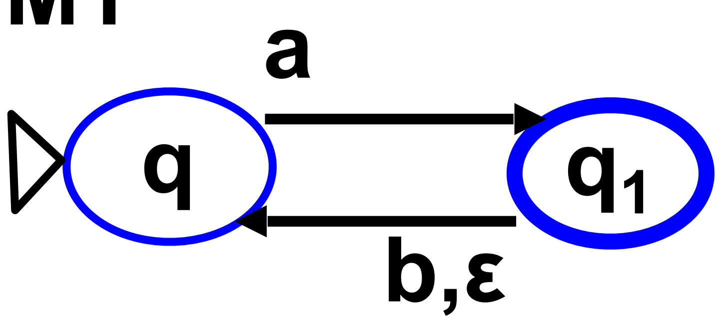


L(M) = {a}{ba}*



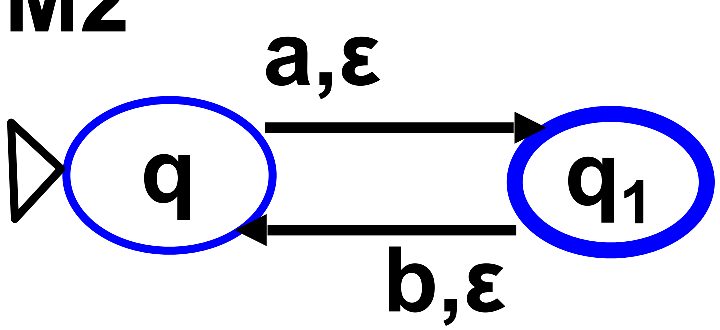
L(M)* = ({a}{ba}*)*

M1



ma ε non è in L(M1),
quindi L(M1) ≠ L(M)*

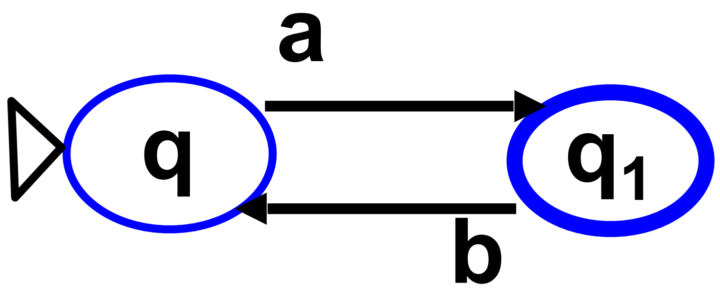
M2



ora ε è in L(M2),
ma anche b, e quindi L(M2) ≠ L(M)*

aggiunta nuovo stato per la *

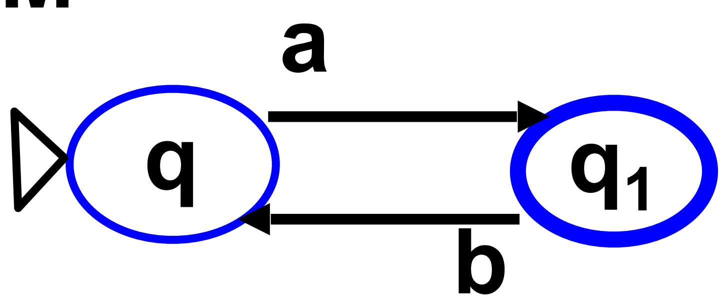
M



L(M) = {a}{ba}*

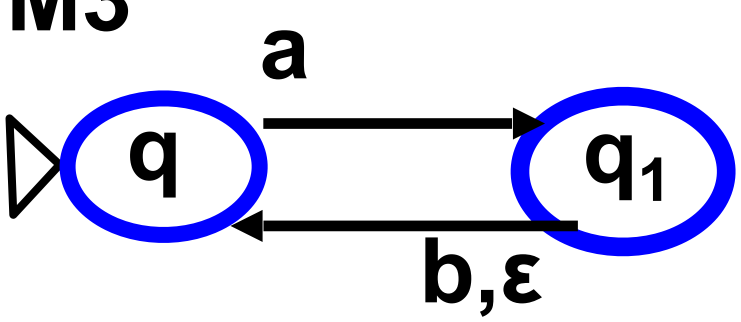
aggiunta nuovo stato per la *

M

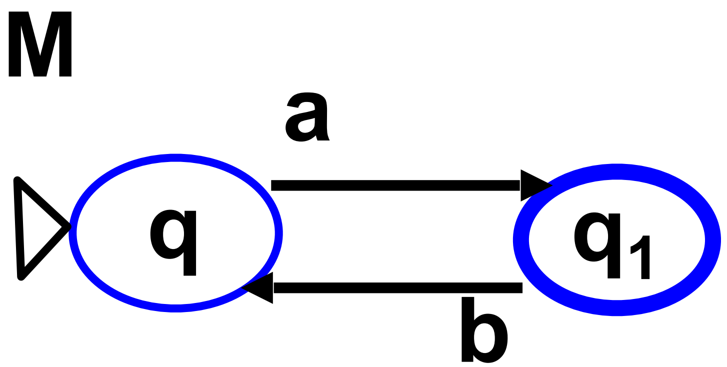


L(M) = {a}{ba}*

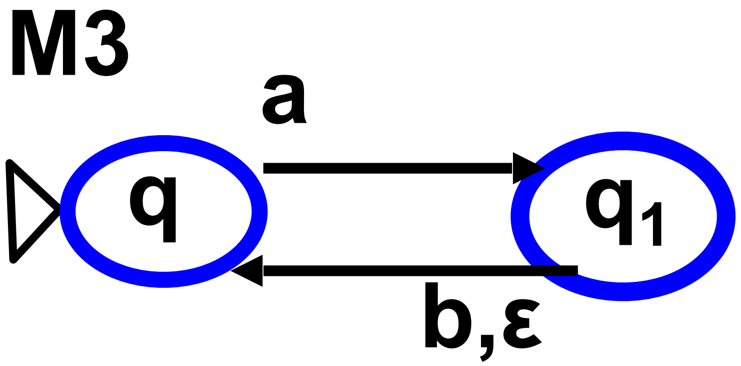
M3



aggiunta nuovo stato per la *



$L(M) = \{a\}\{ba\}^*$



ma ab è in L(M3),
quindi $L(M3) \neq L(M)^*$

Le epsilon mosse (da sole) non fanno nondeterminismo

Un **automa a stati finiti deterministico con epsilon mosse**, in breve ε-DFA , è una quintupla $M = (Q,\Sigma,\delta,q_0,F)$ dove

Le epsilon mosse (da sole) non fanno nondeterminismo

Un **automa a stati finiti deterministico con epsilon mosse**, in breve ε-DFA , è una quintupla $M = (Q,\Sigma,\delta,q_0,F)$ dove

- Q e Σ sono l' insieme degli stati e dei simboli di input;

Le epsilon mosse (da sole) non fanno nondeterminismo

Un **automa a stati finiti deterministico con epsilon mosse**, in breve ε-DFA , è una quintupla $M = (Q,\Sigma,\delta,q_0,F)$ dove

- Q e Σ sono l'insieme degli stati e dei simboli di input;
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow P(Q)$ è la funzione di transizione, tale che
  - $|\delta(q,a)| \leq 1$ per ogni a in $\Sigma \cup \{\varepsilon\}$ e ogni q in Q
  - se $|\delta(q,\varepsilon)| = 1$ allora $\delta(q,a) = \emptyset$

Le epsilon mosse (da sole) non fanno nondeterminismo

Un **automa a stati finiti deterministico con epsilon mosse**, in breve ε-DFA , è una quintupla $M = (Q,\Sigma,\delta,q_0,F)$ dove

- Q e Σ sono l' insieme degli stati e dei simboli di input;
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to P(Q)$ è la funzione di transizione, tale che
  - $|\delta(q,a)| \leq 1$ per ogni a in $\Sigma \cup \{\varepsilon\}$ e ogni q in Q
  - se $|\delta(q,\varepsilon)| = 1$ allora $\delta(q,a) = \emptyset$
- q_0 è lo stato iniziale e $F \subseteq Q$ è l' insieme degli stati finali;

Esempi di NFA

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

L = {x010y | x,y in {0,1}*}

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

L = {x010y | x,y in {0,1}*}

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

L = {x010y | x,y in {0,1}*}

e il DFA equivalente

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

L = {x010y | x,y in {0,1}*}

e il DFA equivalente

Esempi di NFA

Si costruisca un NFA che accetta il linguaggio:

L = {x010y | x,y in {0,1}*}

e il DFA equivalente

Esercizio

Esercizio

Si costruisca un DFA che accetta il linguaggio delle stringhe binarie che rappresentano numeri multipli di sei:

Esercizio

Si costruisca un DFA che accetta il linguaggio delle stringhe binarie che rappresentano numeri multipli di sei:

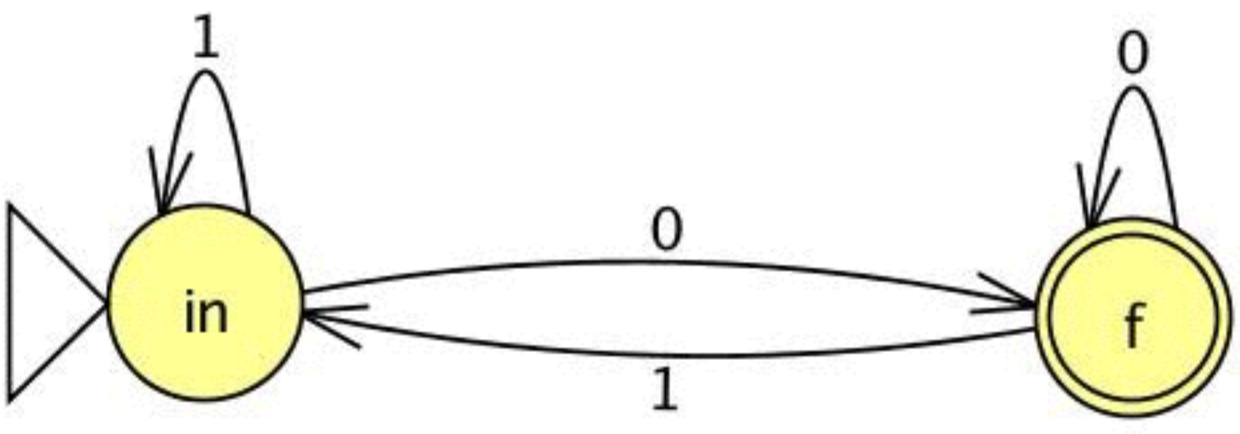
Esercizio

Si costruisca un DFA che accetta il linguaggio delle stringhe binarie che rappresentano numeri multipli di sei:

Esercizio

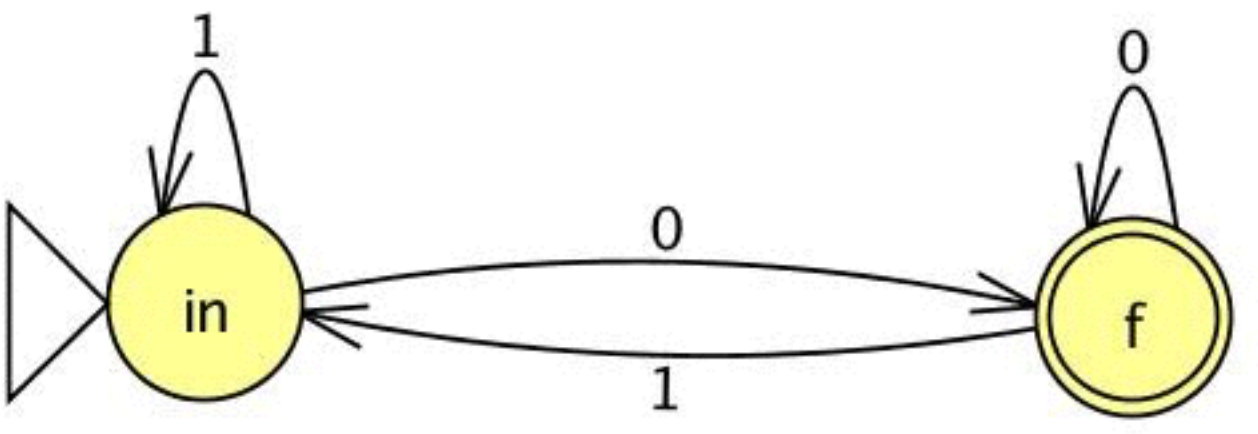
Si costruisca un DFA che accetta il linguaggio delle stringhe binarie che rappresentano numeri multipli di sei:

Esercizio chiusure



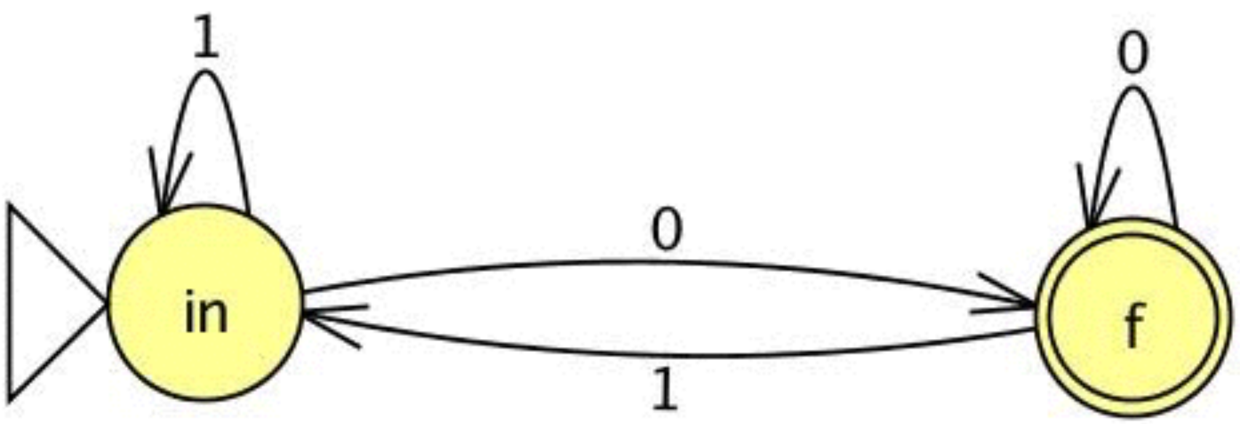
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di 2:



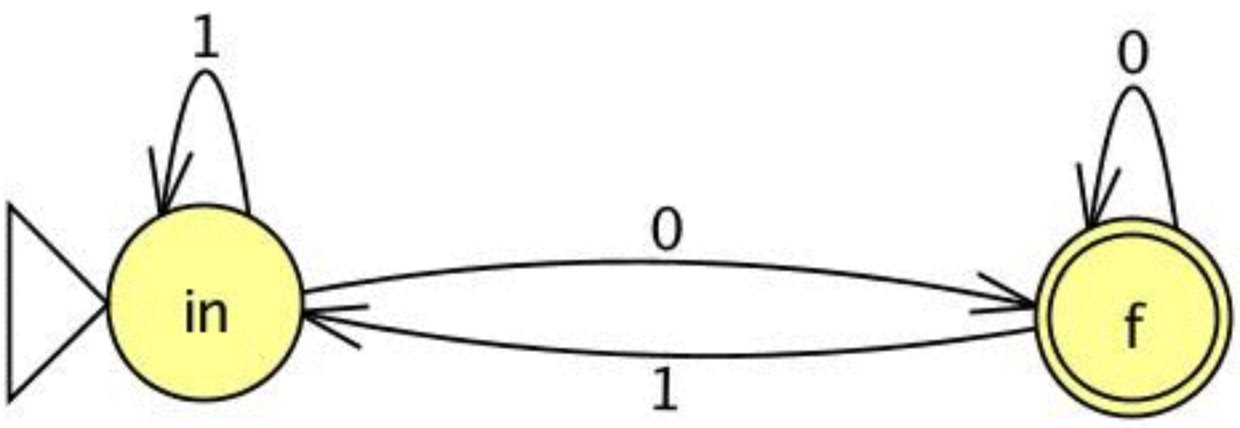
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di 2:



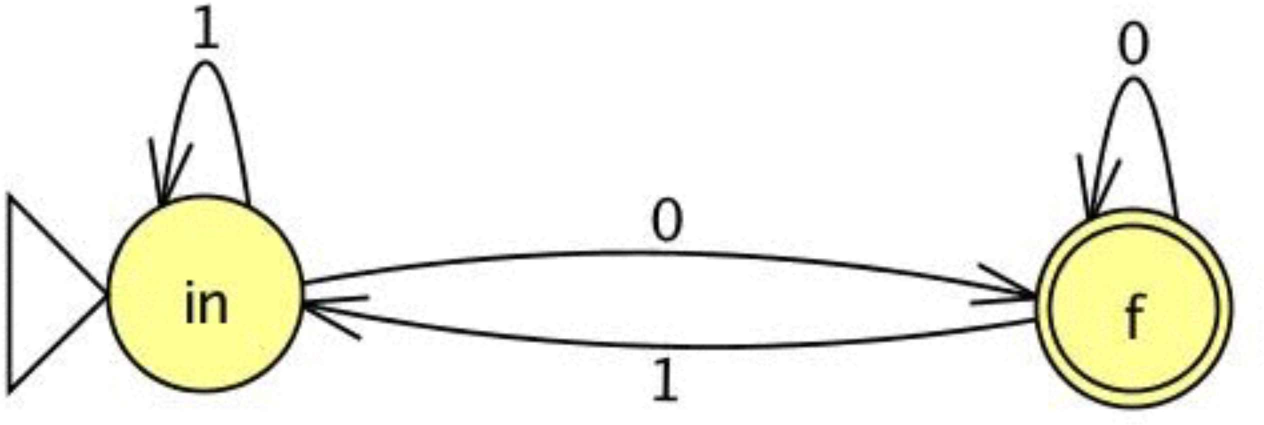
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di 2:

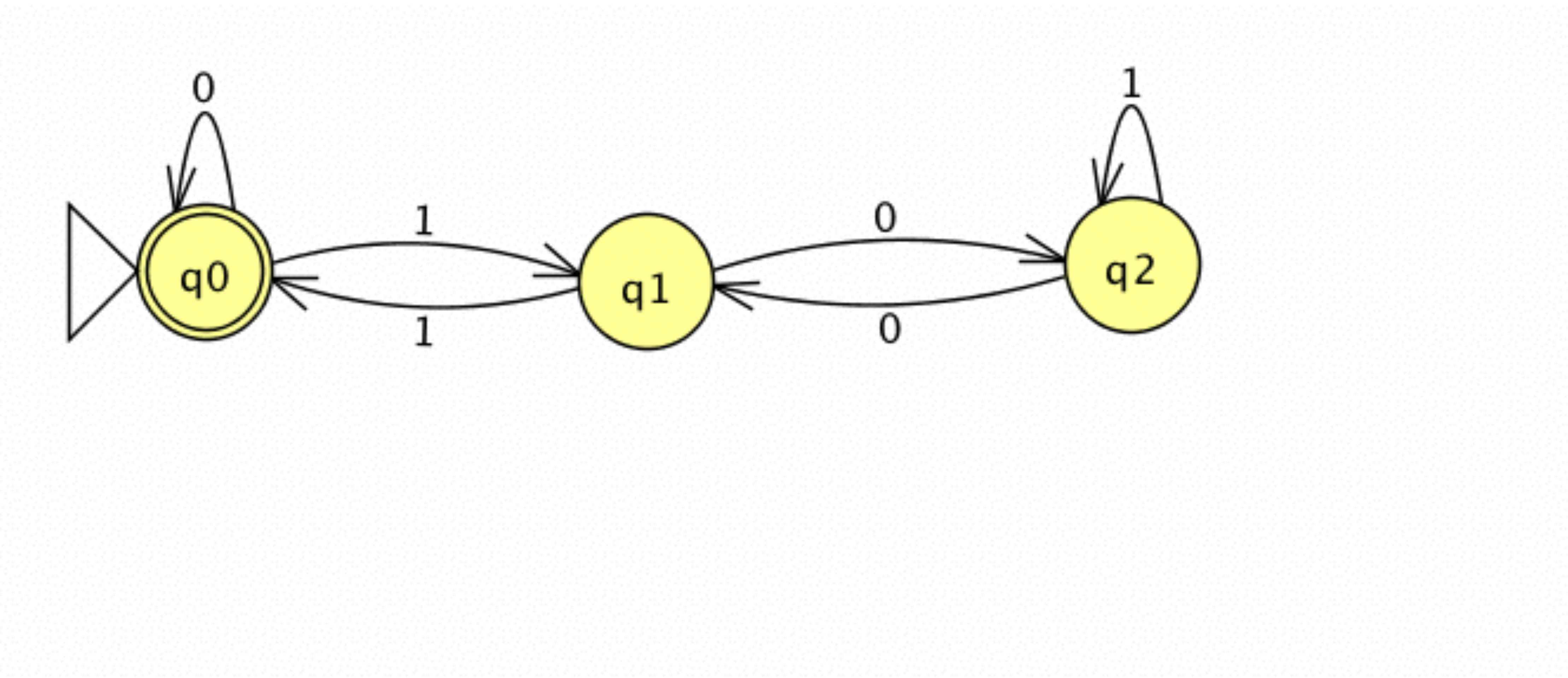


Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di 2:

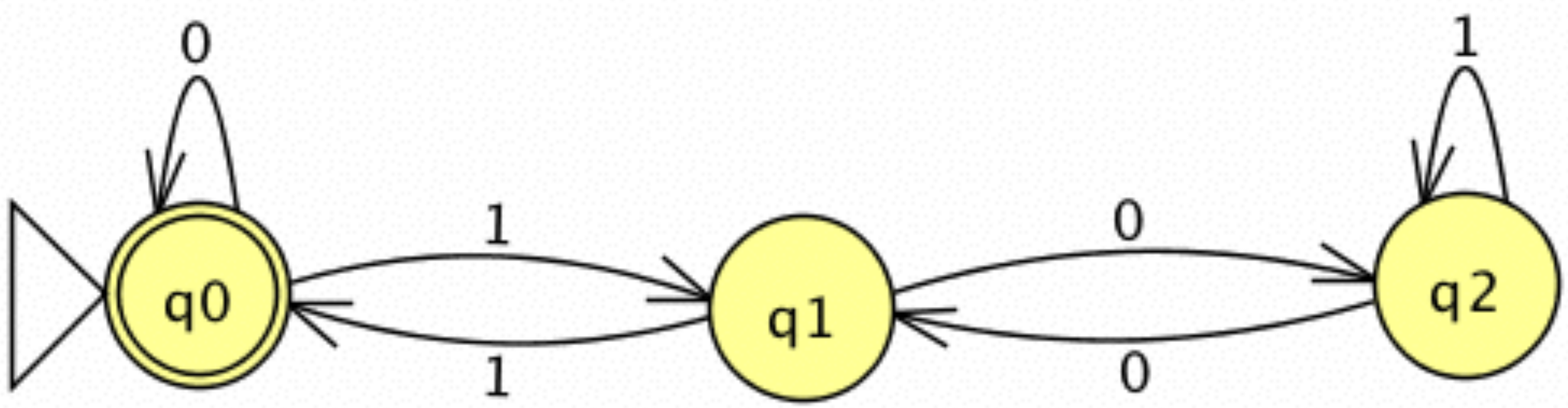


Esercizio chiusure



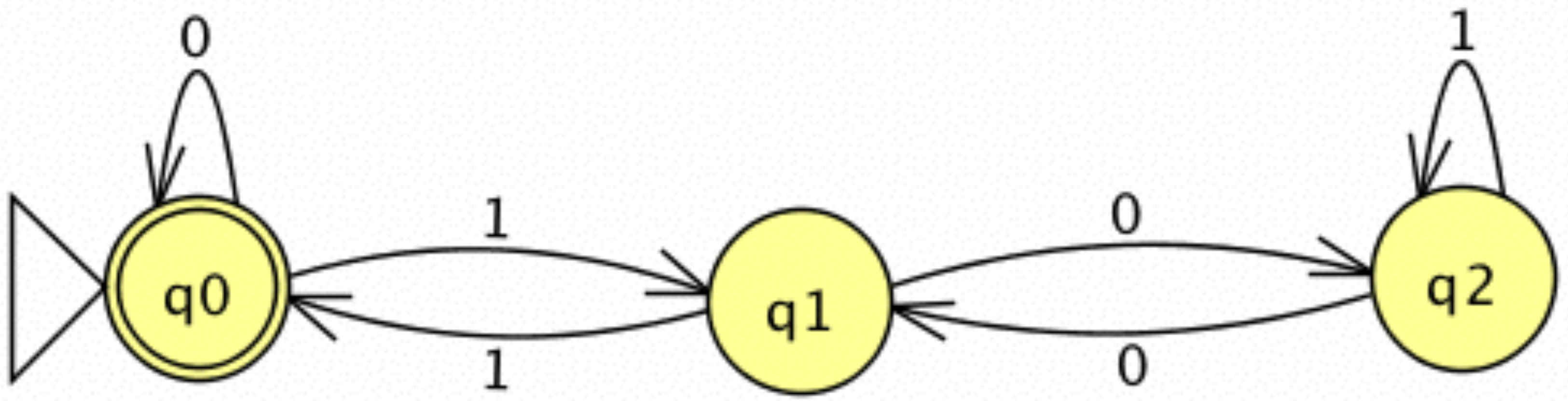
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di tre:



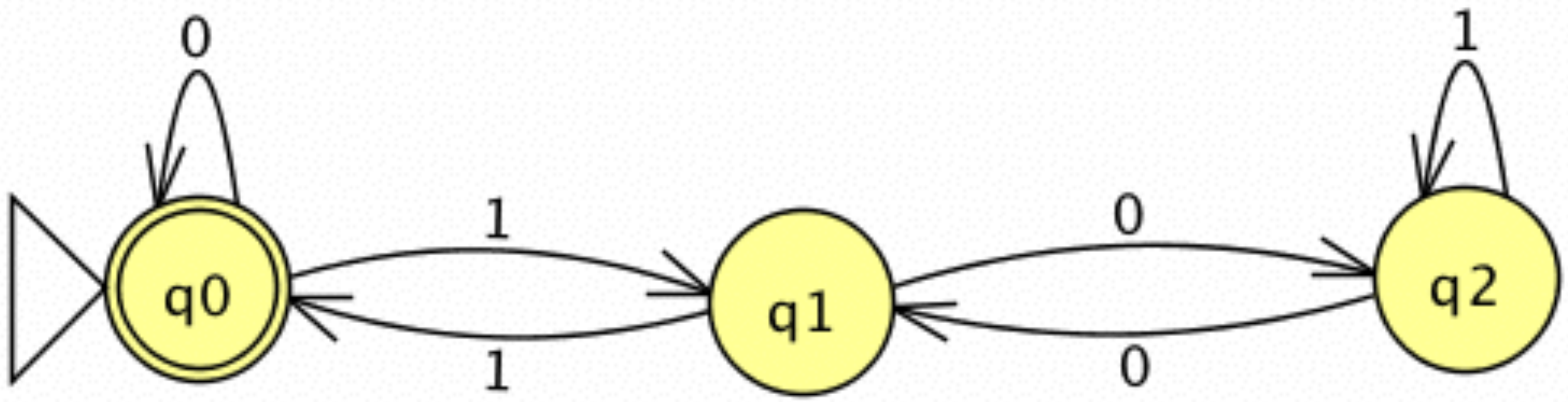
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di tre:



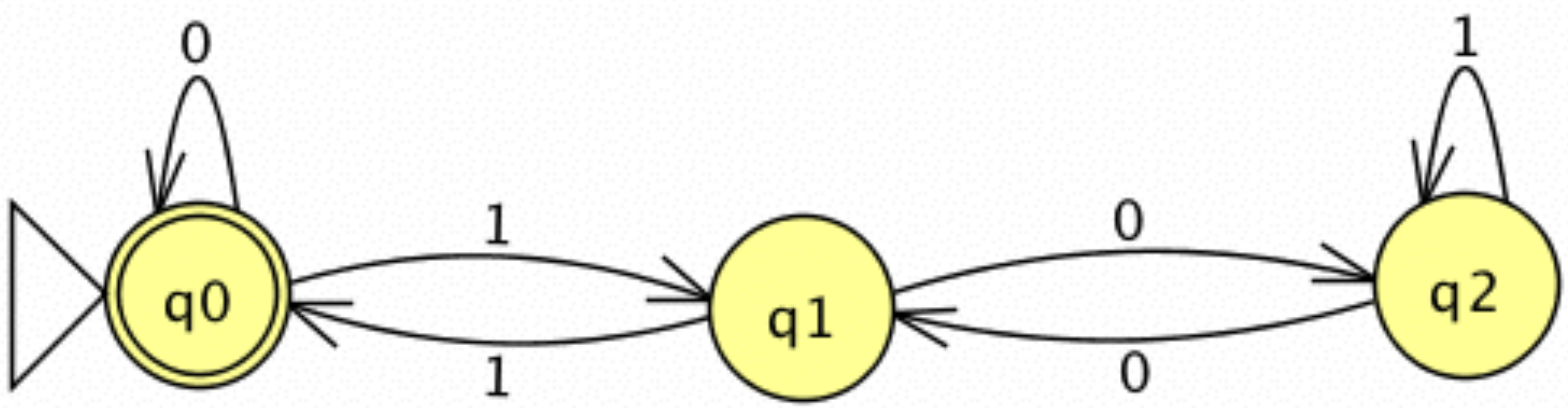
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di tre:



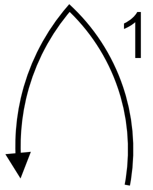
Esercizio chiusure

Questo automa accetta le stringhe binarie che rappresentano numeri multipli di tre:



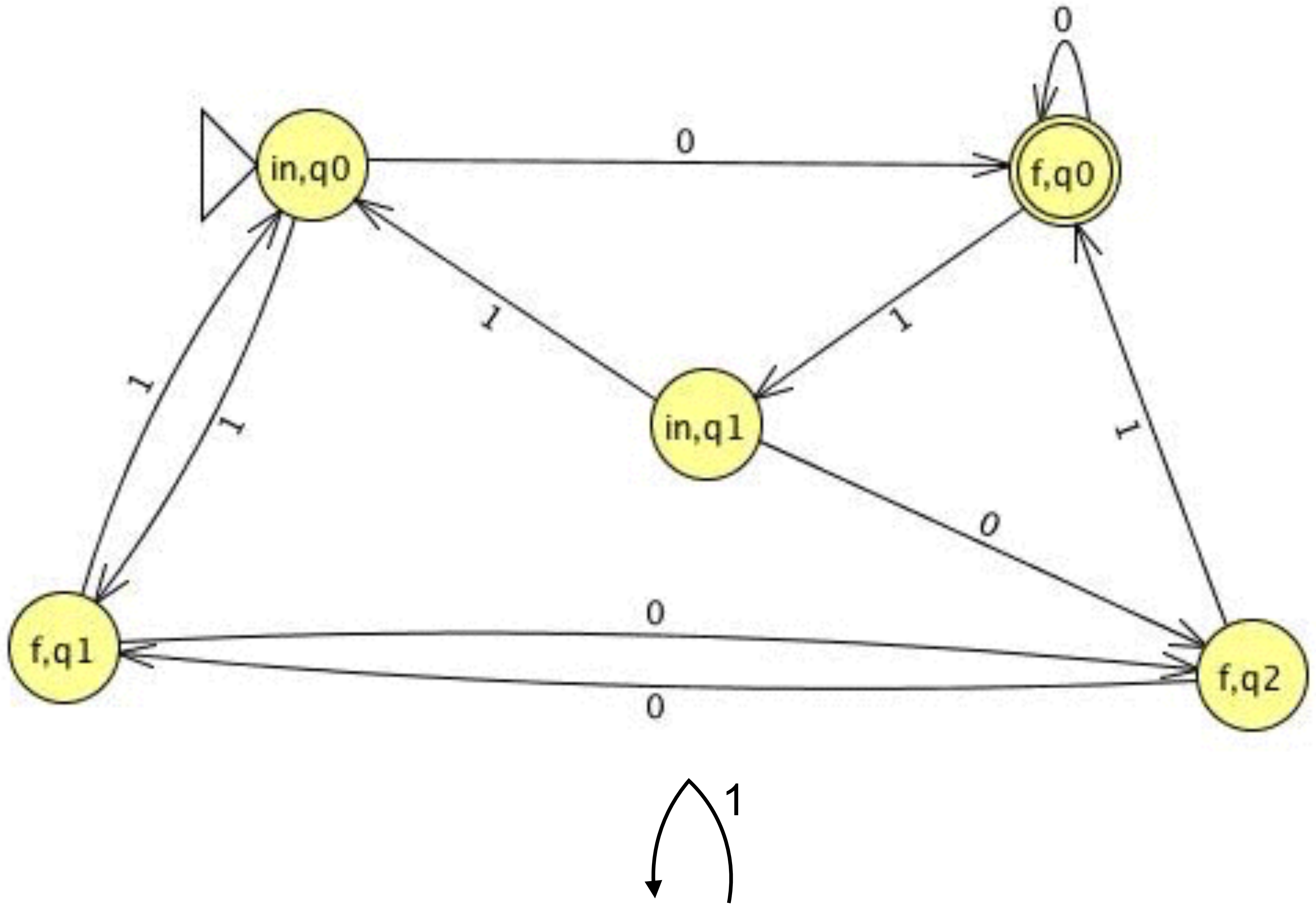
Esercizio chiusure

L'intersezione tra i due accetta solo le stringhe binarie che rappresentano numeri multipli di 6:



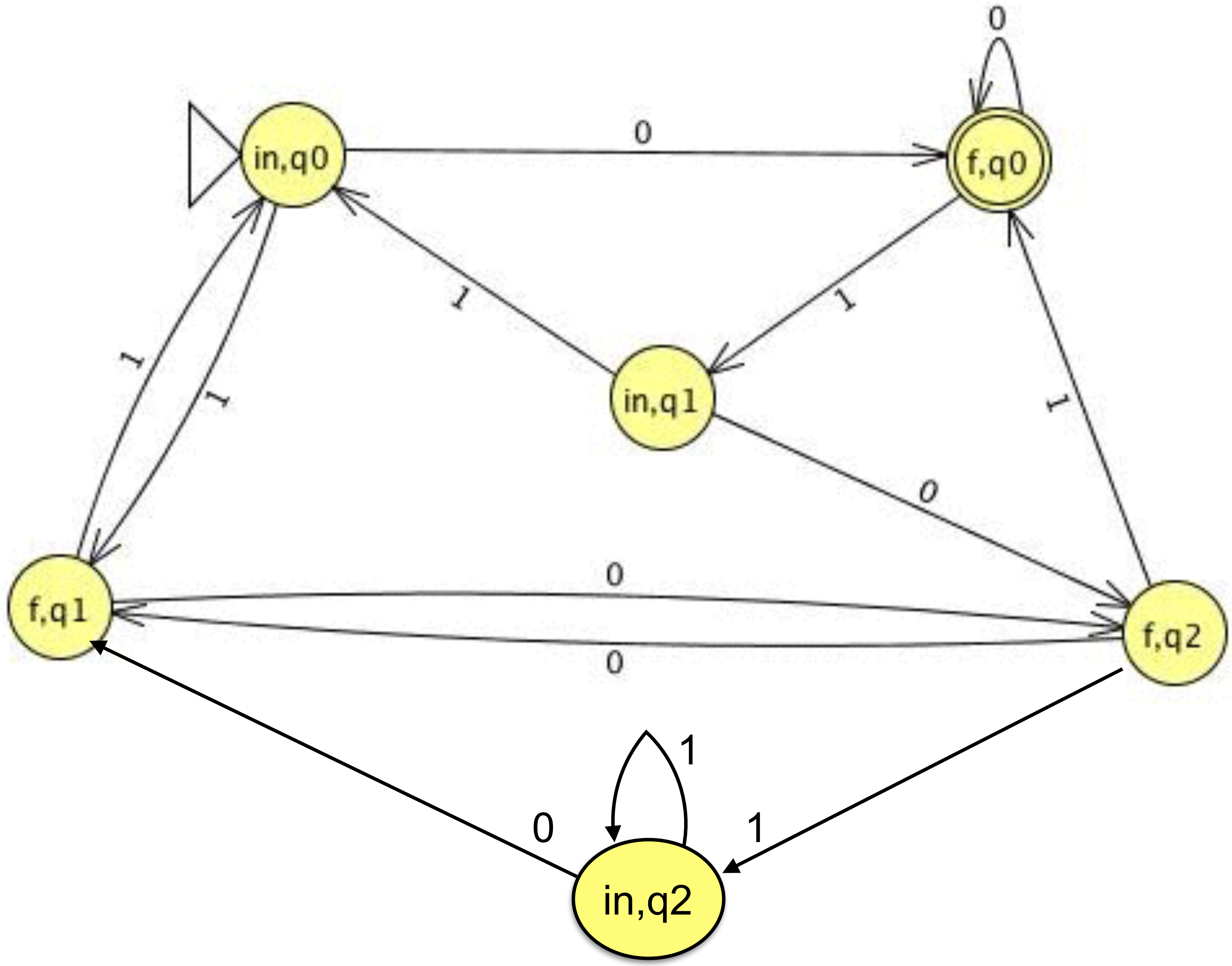
Esercizio chiusure

L'intersezione tra i due accetta solo le stringhe binarie che rappresentano numeri multipli di 6:



Esercizio chiusure

L'intersezione tra i due accetta solo le stringhe binarie che rappresentano numeri multipli di 6:



PUMPING LEMMA

Enunciato definitivo: Se L è un linguaggio regolare, allora esiste un n tale che tutte le parole w in L di lunghezza maggiore o uguale a n si possono ottenere come concatenazione di tre sottoparole, x,y e z, w = xyz, e

PUMPING LEMMA

Enunciato definitivo: Se L è un linguaggio regolare, allora esiste un n tale che tutte le parole w in L di lunghezza maggiore o uguale a n si possono ottenere come concatenazione di tre sottoparole, x,y e z, w = xyz, e

1. $|y|>0$
2. $|xy|\leq n$
3. le parole $w_i = xy^iz$, per ogni $i\geq 0$ sono in L

PUMPING LEMMA

Enunciato definitivo: Se L è un linguaggio regolare, allora esiste un n tale che tutte le parole w in L di lunghezza maggiore o uguale a n si possono ottenere come concatenazione di tre sottoparole, x,y e z, w = xyz, e

1. $|y|>0$

é l'etichetta di un ciclo che almeno comprende due stati

2. $|xy| \leq n$

3. le parole $w_i = xy^iz$, per ogni $i \geq 0$ sono in L

PUMPING LEMMA

Enunciato definitivo: Se L è un linguaggio regolare, allora esiste un n tale che tutte le parole w in L di lunghezza maggiore o uguale a n si possono ottenere come concatenazione di tre sottoparole, x,y e z, w = xyz, e

1. $|y|>0$

é l'etichetta di un ciclo che almeno comprende due stati

2. $|xy|\leq n$

si assicura scegliendo il primo ciclo!

3. le parole $w_i = xy^iz$, per ogni $i\geq 0$ sono in L

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare. Basta quindi dimostrare che è vera la negazione della proprietà necessaria.

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare. Basta quindi dimostrare che è vera la negazione della proprietà necessaria.

Sia L un linguaggio che si sospetta non regolare.

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare. Basta quindi dimostrare che è vera la negazione della proprietà necessaria.

Sia L un linguaggio che si sospetta non regolare.

Si tratta di dimostrare che per ogni n siamo in grado di individuare una parola w in L di lunghezza almeno n tale che per ogni sua scomposizione nella concatenazione di tre sottoparole x,y,z con $|y|>0$, e $|xy|\leq n$ non siamo in grado di concludere che $w_i = xy^iz$ è in L per ogni $i\geq 0$. Questo vuol dire che siamo in grado di individuare almeno un indice j tale che $w_j = xy^jz$ non è in L.

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

- **si prende un qualsiasi intero positivo k,**

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

- **si prende un qualsiasi intero positivo k,**

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

- **si prende un qualsiasi intero positivo k,**
- **si sceglie (questa è la parte creativa) z in L tale che $|z| \geq k$**

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

- **si prende un qualsiasi intero positivo k,**
- **si sceglie (questa è la parte creativa) z in L tale che $|z| \geq k$**

Uso Pumping Lemma

Si usa per provare che un linguaggio NON è regolare.

STRATEGIA DI PROVA

Sia L un linguaggio che si sospetta non regolare:

- **si prende un qualsiasi intero positivo k,**
- **si sceglie (questa è la parte creativa) z in L tale che $|z| \geq k$**
- **si mostra che per <u>ogni</u> fattorizzazione $z = uvw$ con $|v| \geq 1$ e $|uv| \leq k$, <u>esiste</u> un $i \geq 0$ tale che $uv^iw \notin L$.**

Esempio

Esempio

$L = \{a^n b^n \mid n \geq 0\}$ non è regolare.

Esempio

$L = \{a^n b^n \mid n \geq 0\}$ non è regolare.

Per ogni n consideriamo le parole $w = a^n b^n$ (dato il linguaggio non c'è molta scelta!). Consideriamo le possibili scomposizioni $w = xyz$. Poiché $|xy| \leq n$, sia x che y sono di sole a quindi

Esempio

$L = \{a^n b^n \mid n \geq 0\}$ non è regolare.

Per ogni n consideriamo le parole $w = a^n b^n$ (dato il linguaggio non c'è molta scelta!). Consideriamo le possibili scomposizioni $w = xyz$. Poiché $|xy| \leq n$, sia x che y sono di sole a quindi

1. $y = a^m$, per $m>0$, allora $w_i=x(a^m)^i z$ ha meno a di b se $i=0$.

Esempio

$L = \{a^n b^n \mid n \geq 0\}$ non è regolare.

Per ogni n consideriamo le parole $w = a^n b^n$ (dato il linguaggio non c'è molta scelta!). Consideriamo le possibili scomposizioni $w = xyz$. Poiché $|xy| \leq n$, sia x che y sono di sole a quindi

1. $y = a^m$, per $m>0$, allora $w_i = x(a^m)^i z$ ha meno a di b se $i=0$.

E abbiamo considerato tutte le scomposizioni che rispettano il vincolo $|xy| \leq n$

Esempio

Esempio

Se volessimo dimostrare che $L = \{a^n b^n \mid n \geq 0\}$ non è regolare applicando il pumping lemma ma ignorando il vincolo 2, dovremmo considerare per ogni n le parole $w = a^n b^n$ e tutte le possibili scomposizioni $w = xyz$:

Esempio

Se volessimo dimostrare che L = $\{a^n b^n \mid n \geq 0\}$ non è regolare applicando il pumping lemma ma ignorando il vincolo 2, dovremmo considerare per ogni n le parole w = $a^n b^n$ e tutte le possibili scomposizioni w = xyz:

1. y = a^m, per m>0, allora $w_i=x(a^m)^i z$ ha meno a di b se i=0.

Esempio

Se volessimo dimostrare che $L = \{a^n b^n \mid n \geq 0\}$ non è regolare applicando il pumping lemma ma ignorando il vincolo 2, dovremmo considerare per ogni n le parole $w = a^n b^n$ e tutte le possibili scomposizioni $w = xyz$:

1. $y = a^m$, per $m>0$, allora $w_i=x(a^m)^i z$ ha meno a di b se $i=0$.

2. $y = b^m$, analogo

Esempio

Se volessimo dimostrare che $L = \{a^n b^n \mid n \geq 0\}$ non è regolare applicando il pumping lemma <u>ma ignorando il vincolo 2</u>, dovremmo considerare per ogni n le parole $w = a^n b^n$ e tutte le possibili scomposizioni $w = xyz$:

1. $y = a^m$, per $m>0$, allora $w_i=x(a^m)^i z$ ha meno a di b se $i=0$.

2. $y = b^m$, analogo

3. $y = a^m b^p$, con $m,p \geq 1$, allora $y = x(a^m b^p)^i z$ contiene una a che segue una b per ogni $i>0$.

Esempio

Se volessimo dimostrare che $L = \{a^n b^n \mid n \geq 0\}$ non è regolare applicando il pumping lemma <u>ma ignorando il vincolo 2</u>, dovremmo considerare per ogni n le parole $w = a^n b^n$ e tutte le possibili scomposizioni $w = xyz$:

1. $y = a^m$, per $m>0$, allora $w_i=x(a^m)^i z$ ha meno a di b se $i=0$.

2. $y = b^m$, analogo

3. $y = a^m b^p$, con $m,p \geq 1$, allora $y = x(a^m b^p)^i z$ contiene una a che segue una b per ogni $i>0$.

Molto lavoro in più anche in questo semplice caso!

Esempio

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

Esempio

$L = \{a^n b^m \mid n > m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

con $r+s+t = n+1$ e $t \geq 1$

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

con $r+s+t = n+1$ e $t\geq 1$

(in questo modo prendiamo tutte le scomposizioni che soddisfano la condizione $|xy|\leq n$)

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

con $r+s+t = n+1$ e $t \geq 1$

(in questo modo prendiamo tutte le scomposizioni che soddisfano la condizione $|xy| \leq n$)

Poi basta osservare che $xy^0 = a^r$, e

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

con $r+s+t = n+1$ e $t \geq 1$

(in questo modo prendiamo tutte le scomposizioni che soddisfano la condizione $|xy| \leq n$)

Poi basta osservare che $xy^0 = a^r$, e

allora $w_0=xz = a^r a^t b^n$,ma $r+t \leq n$

Esempio

$L = \{a^n b^m \mid n>m \geq 0\}$ non è regolare.

Per ogni n consideriamo (la parte creativa!) le parole $w = a^{n+1}b^n$. Consideriamo le possibili scomposizioni $w = xyz$, con $|xy| \leq n$.

Allora $x = a^r$, per $r \geq 0$ e $y = a^s$, per $s>0$ e $z= a^t b^n$

con $r+s+t = n+1$ e $t \geq 1$

(in questo modo prendiamo tutte le scomposizioni che soddisfano la condizione $|xy| \leq n$)

Poi basta osservare che $xy^0 = a^r$, e

allora $w_0=xz = a^r a^t b^n$,ma $r+t \leq n$

perché $s \geq 1$ e $r+s+t = n+1$, quindi w_0 non è in L.

Esercizi

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

$$L = \{a^p \mid p \text{ è un numero primo}$$

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

$$L = \{a^p \mid p \text{ è un numero primo}$$

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

L = {a^p | p è un numero primo

L = {w#x | w,x in {0,1}* e x è una

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

L = {a^p | p è un numero primo

L = {w#x | w,x in {0,1}* e x è una sottostringa di w}

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

$$L = \{a^p \mid p \text{ è un numero primo}$$

$$L = \{w\#x \mid w,x \text{ in } \{0,1\}^* \text{ e } x \text{ è una sottostringa di } w\}$$

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

$$L = \{a^p \mid p \text{ è un numero primo}$$

$$L = \{w\#x \mid w,x \text{ in } \{0,1\}^* \text{ e } x \text{ è una sottostringa di } w\}$$

$$L = \{ww^R \mid w \text{ in } \{0,1\}^* \text{ e } w^R$$

Esercizi

Si dimostri, utilizzando il pumping lemma che i seguenti linguaggi non sono regolari.

$$L = \{a^{2^n} \mid n \geq 0\}$$

L = {a^p | p è un numero primo

L = {w#x | w,x in {0,1}* e x è una sottostringa di w}

L = {ww^R | w in {0,1}* e w^R è w rovesciata}

Esercizi

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio L = {00,11}*

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio L = {00,11}*

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio L = {00,11}*

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio L = {00,11}*

Sia L $\subseteq$ Σ* un linguaggio, definiamo la sua chiusura rispetto al prefisso come il linguaggio che contiene ogni prefisso di una parola di L.

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio $L = \{00,11\}^*$

Sia $L \subseteq \Sigma^*$ un linguaggio, definiamo la sua chiusura rispetto al prefisso come il linguaggio che contiene ogni prefisso di una parola di L.
Il prefisso di una parola w è una parola x tale che w=xy, per un qualche y in Σ^*.

Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio L = {00,11}*

Sia $L \subseteq \Sigma^*$ un linguaggio, definiamo la sua chiusura rispetto al prefisso come il linguaggio che contiene ogni prefisso di una parola di L.
Il prefisso di una parola w è una parola x tale che $w=xy$, per un qualche y in Σ^*.
$pref(L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \text{ e } xy \in L\}$

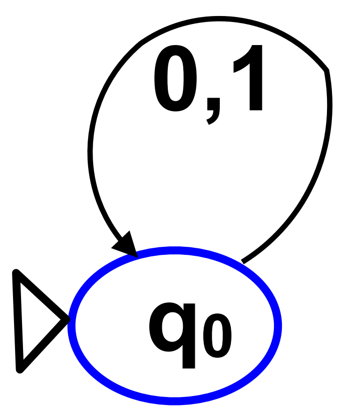
Esercizi

Si spieghi dove fallisce la prova di non regolarità, basata sul pumping lemma, per il linguaggio $L = \{00,11\}^*$

Sia $L \subseteq \Sigma^*$ un linguaggio, definiamo la sua chiusura rispetto al prefisso come il linguaggio che contiene ogni prefisso di una parola di L.
Il prefisso di una parola w è una parola x tale che $w=xy$, per un qualche y in Σ^*.
$\mathrm{pref}(L) = \{x \in \Sigma^* \mid \exists y \in \Sigma^* \text{ e } xy \in L\}$
Si tratta di dimostrare che se L è regolare allora anche pref(L) lo è.

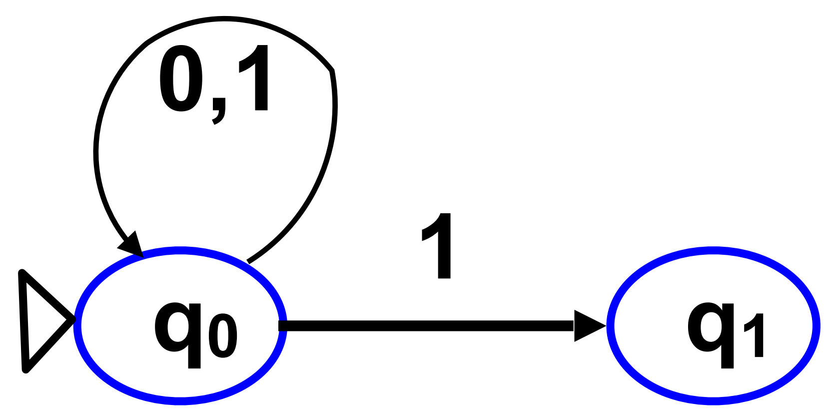
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {0,1}* che contengono un 1 o nella penultima o nella terz'ultima posizione.



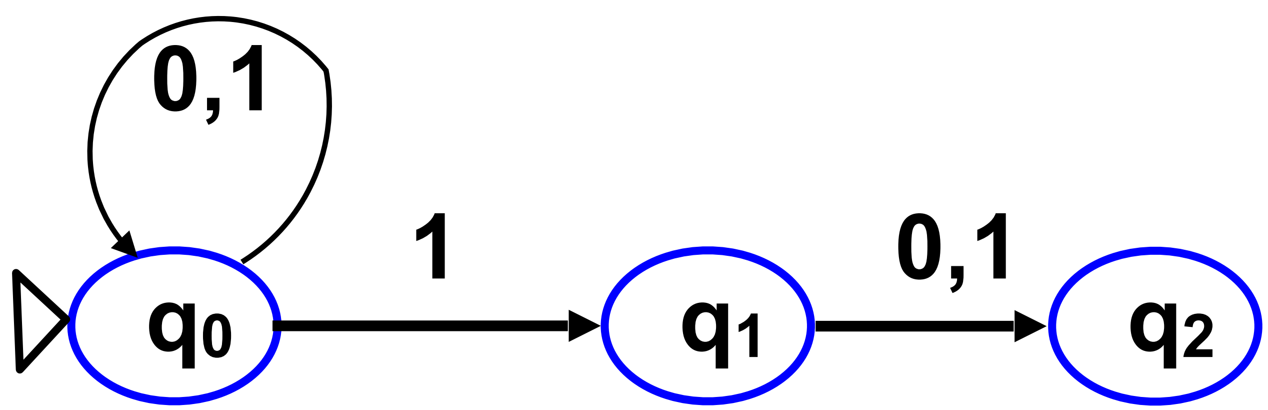
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {0,1}* che contengono un 1 o nella penultima o nella terz'ultima posizione.



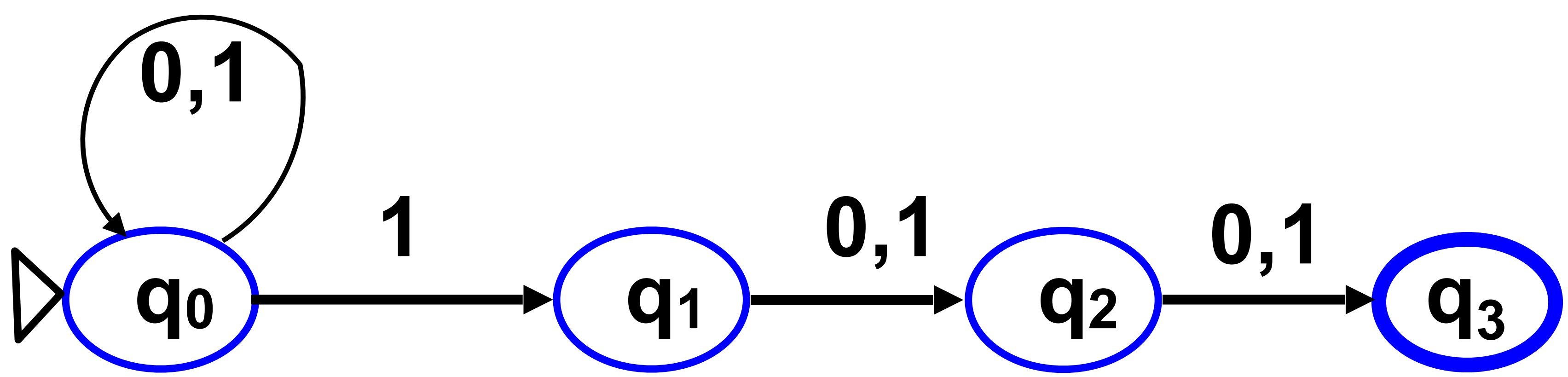
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {0,1}* che contengono un 1 o nella penultima o nella terz'ultima posizione.



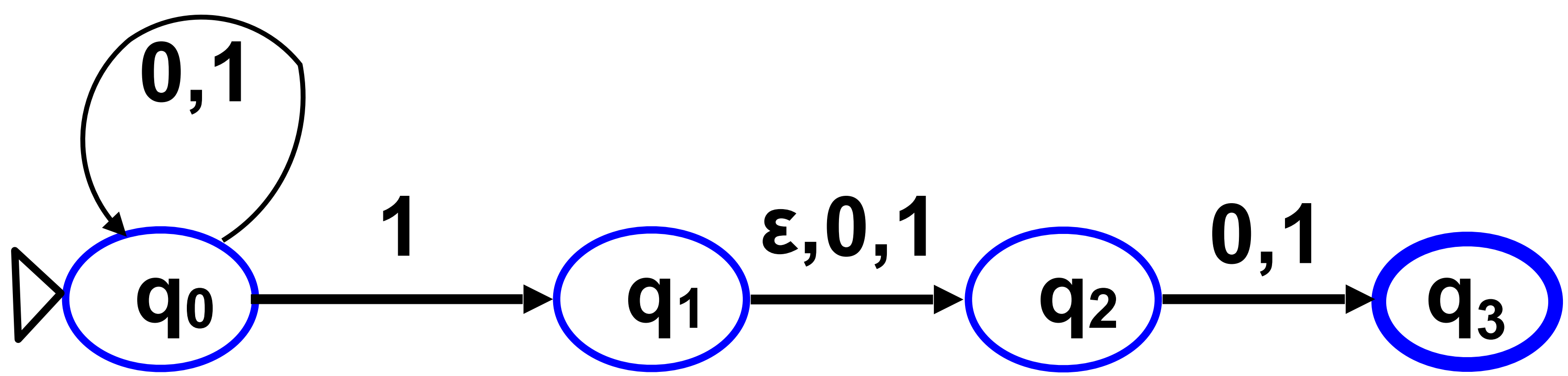
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {0,1}* che contengono un 1 o nella penultima o nella terz'ultima posizione.



Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {0,1}* che contengono un 1 o nella penultima o nella terz'ultima posizione.

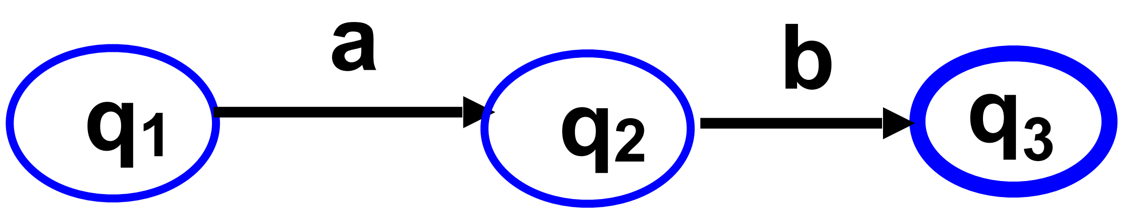


Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.

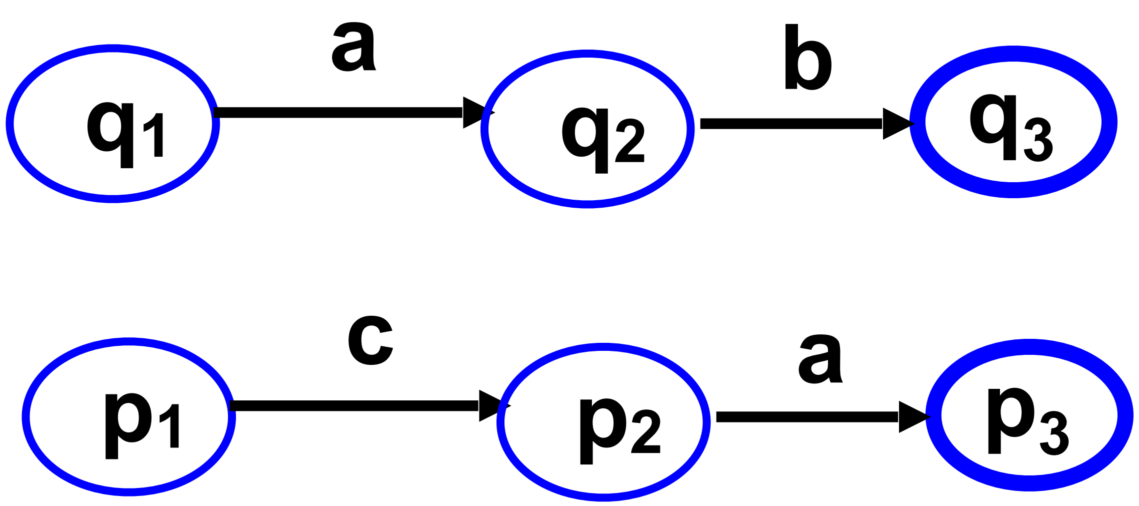
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



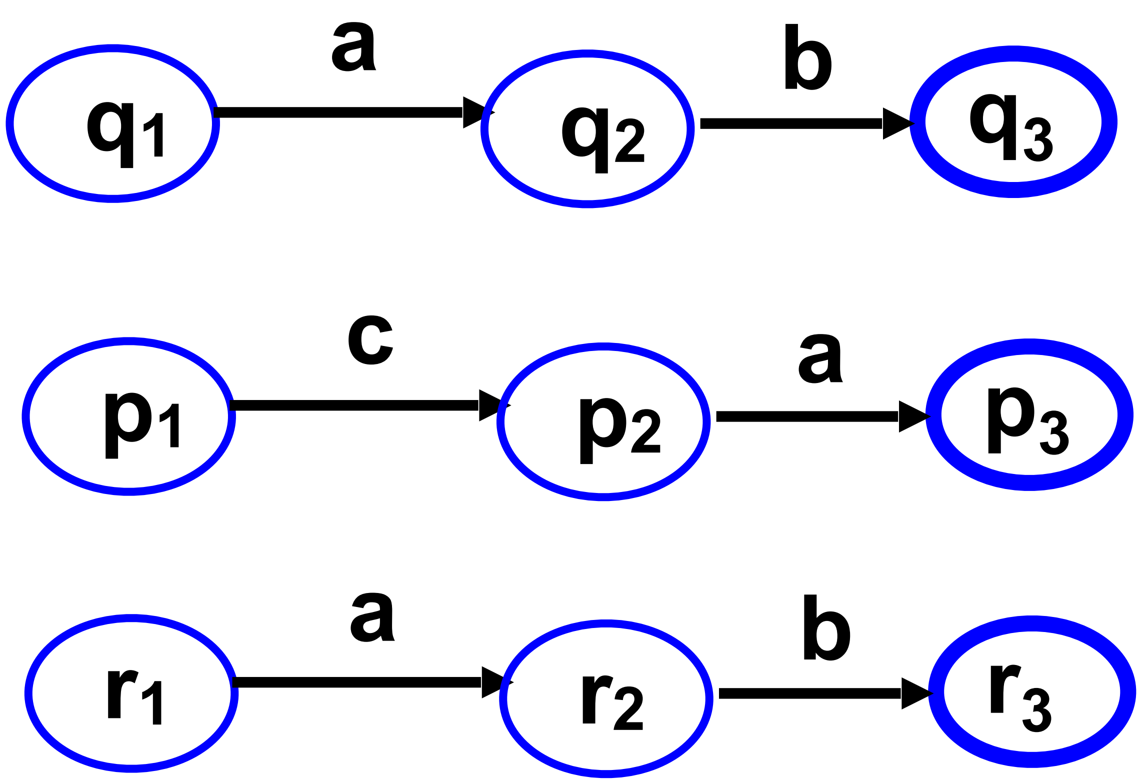
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



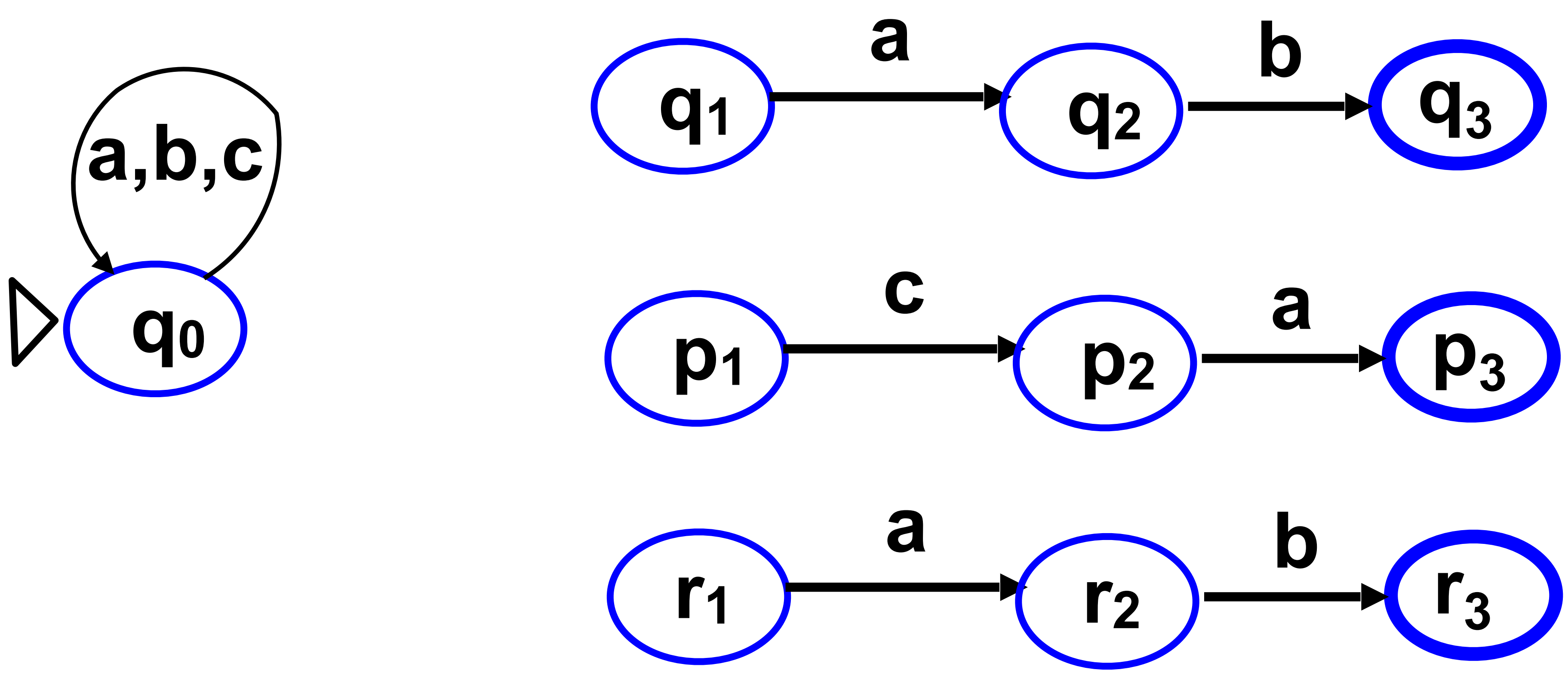
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



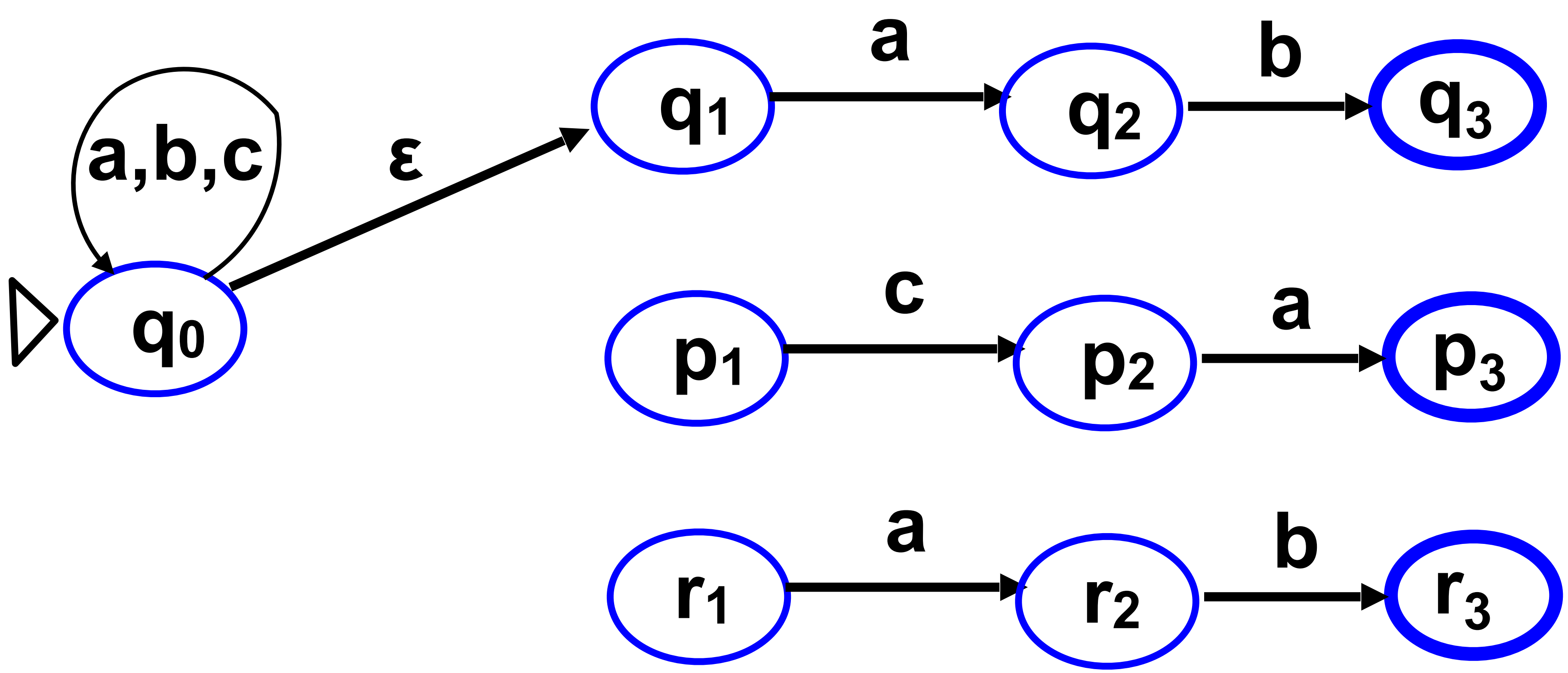
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



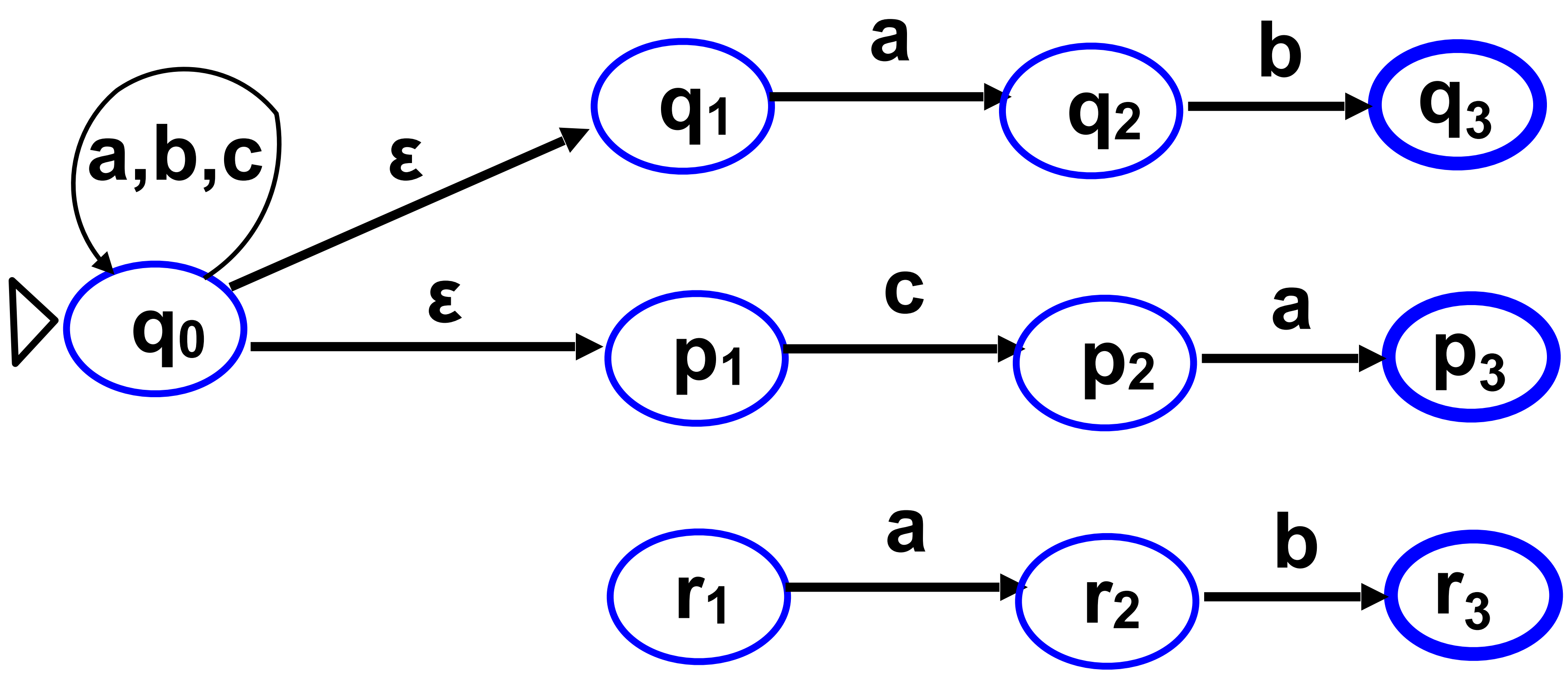
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



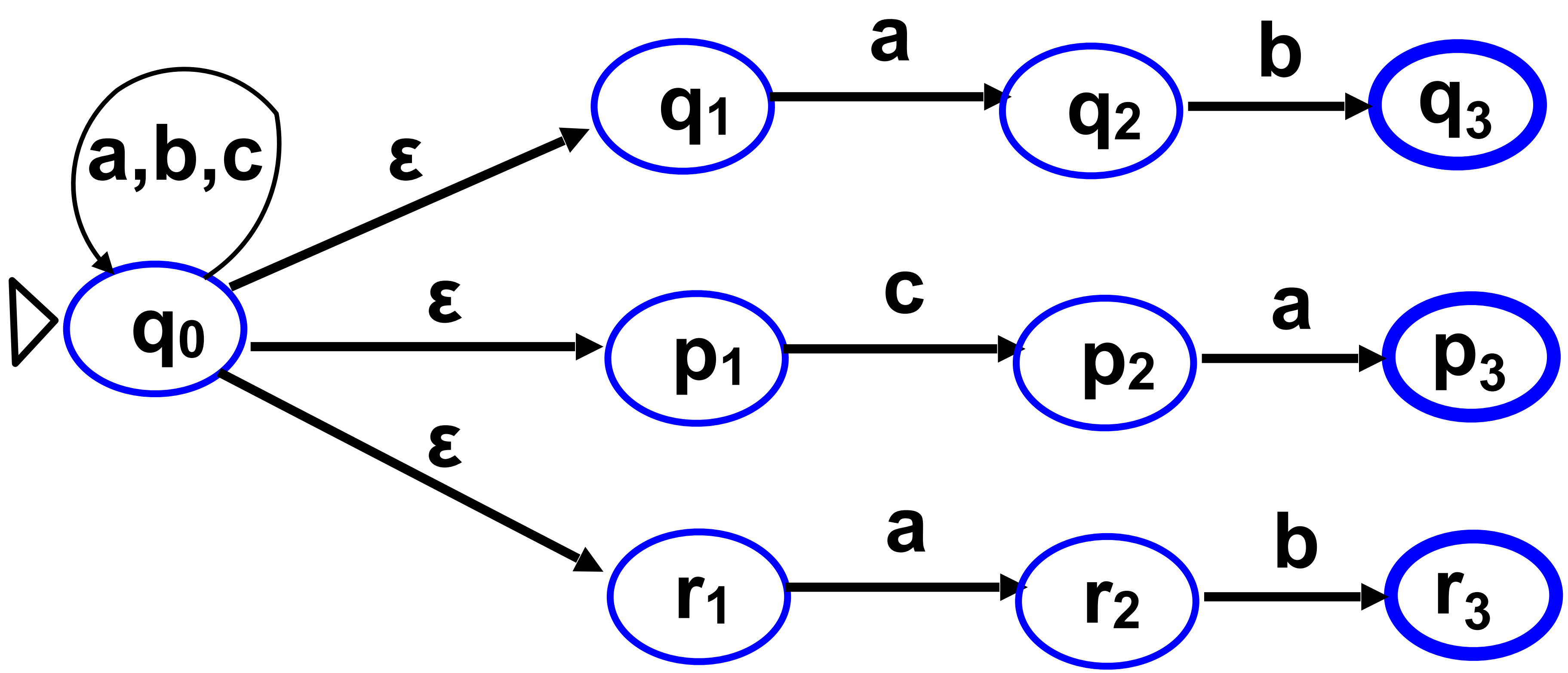
Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



Esercizi

Si costruisca un NFA che accetta tutte le stringhe in {a,b,c}* che terminano con ab,ca oppure ba.



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che non contengono 01 come sottostringa.

Esercizi

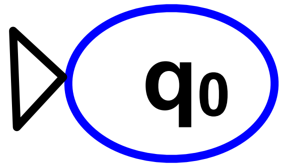
Si costruisca un DFA che accetta tutte le stringhe in $\{0,1\}^*$ che non contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:

Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

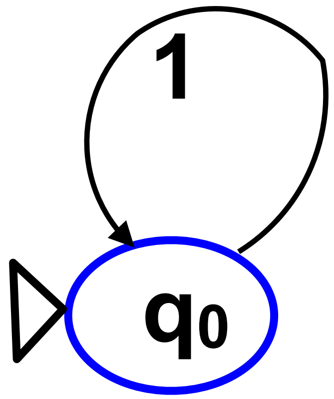
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

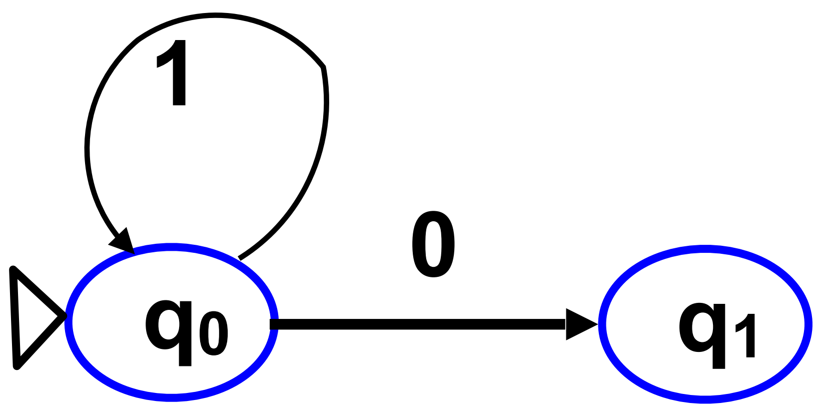
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

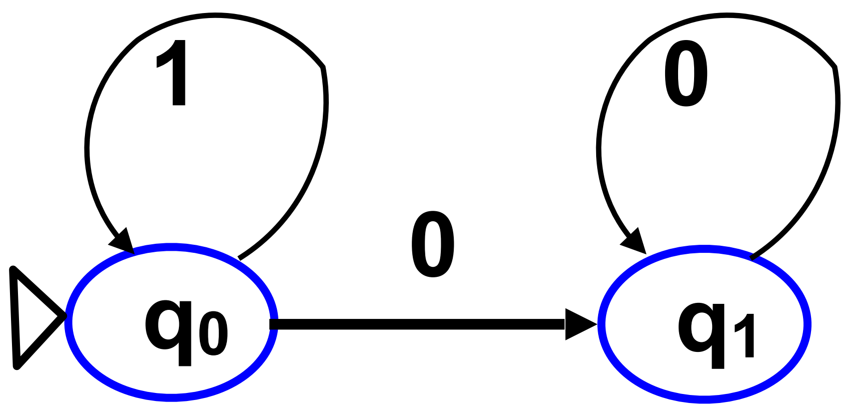
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

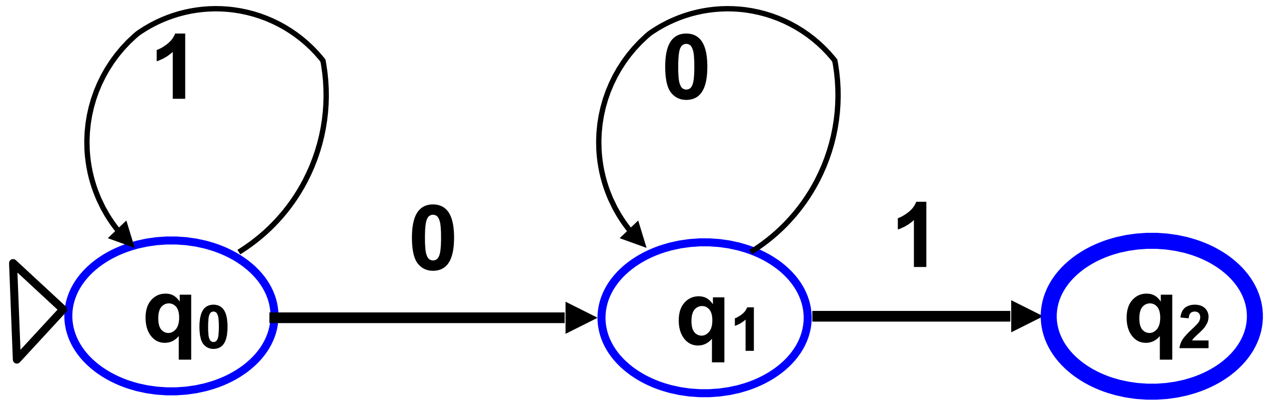
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

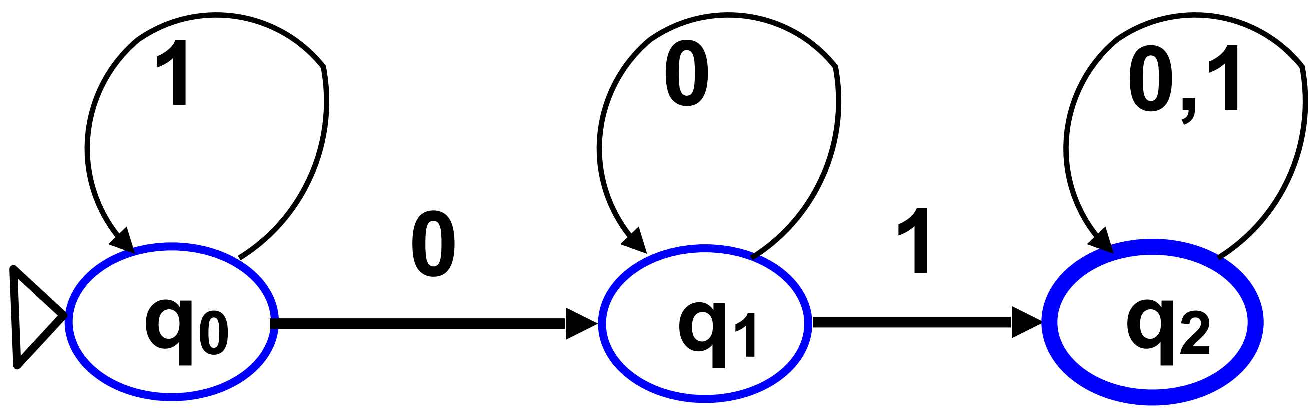
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

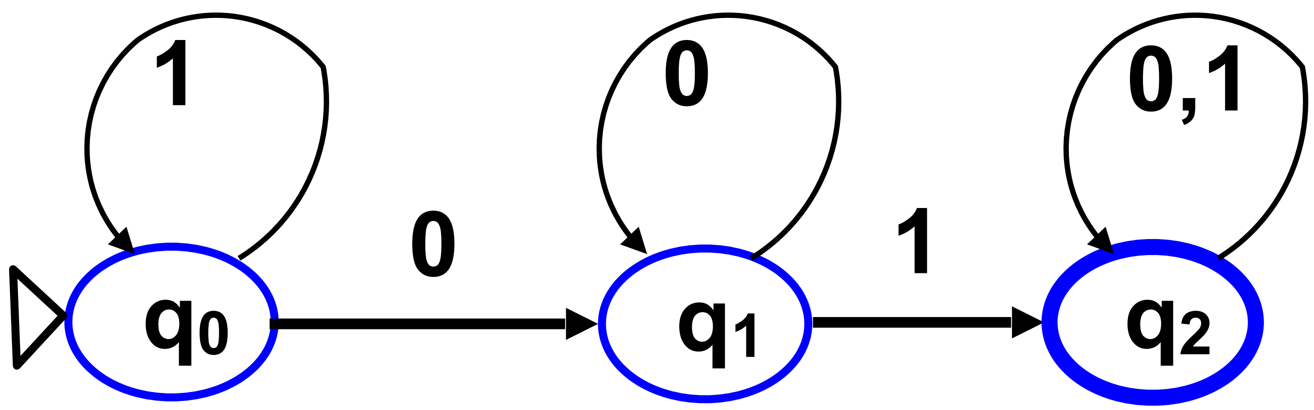
un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:

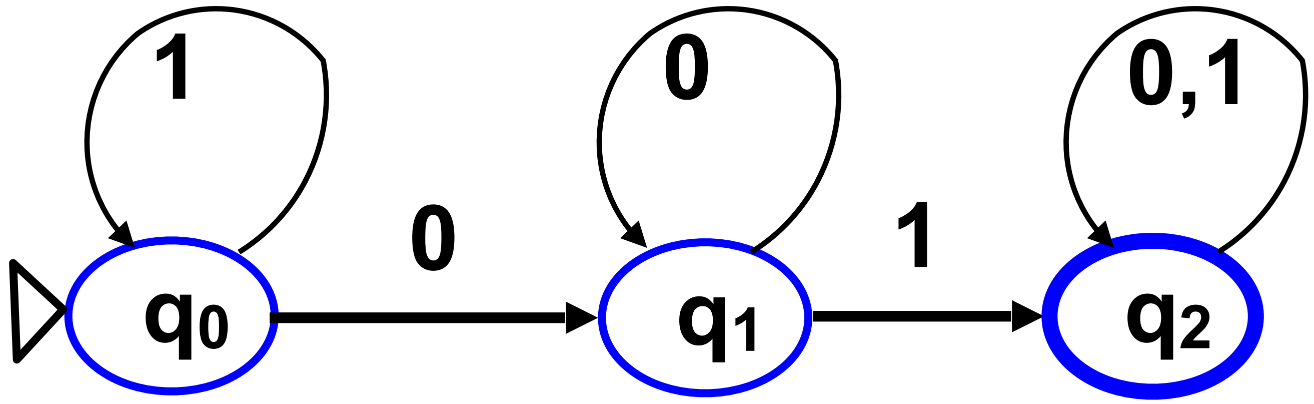


Il DFA per il complemento:

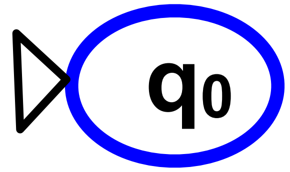
Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che non contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



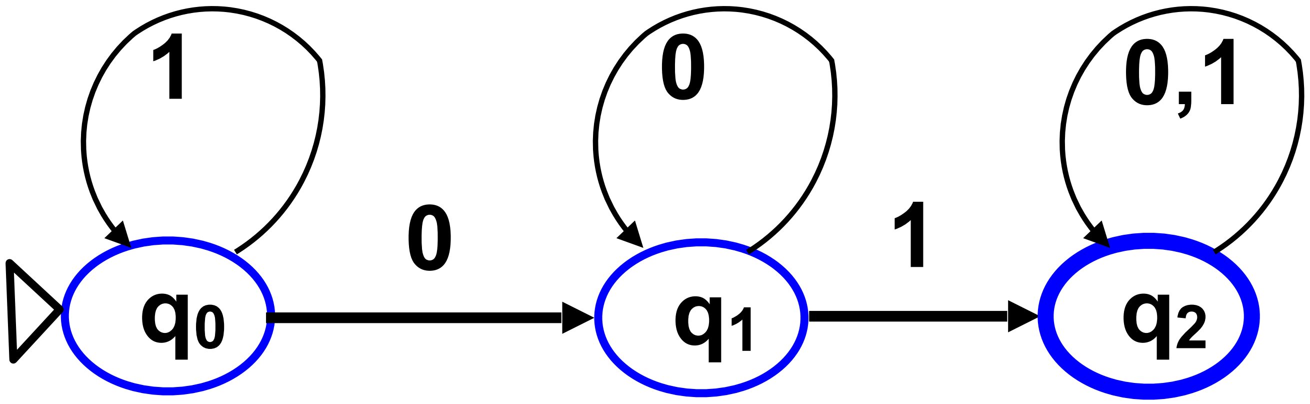
Il DFA per il complemento:



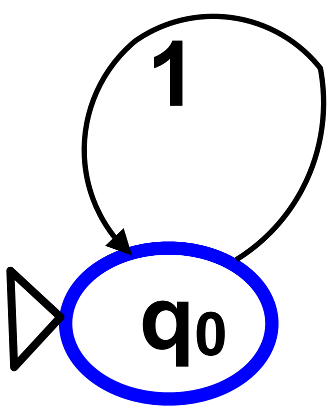
Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



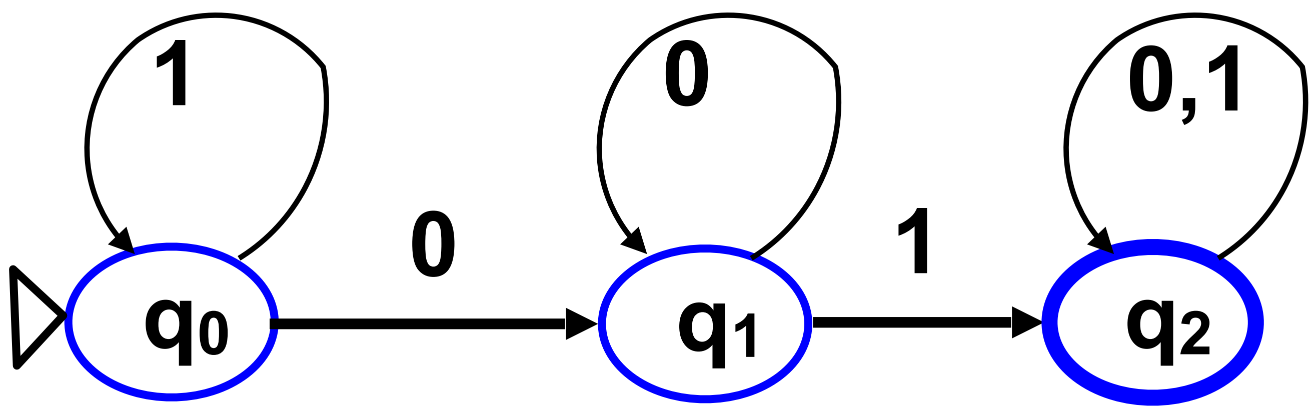
Il DFA per il complemento:



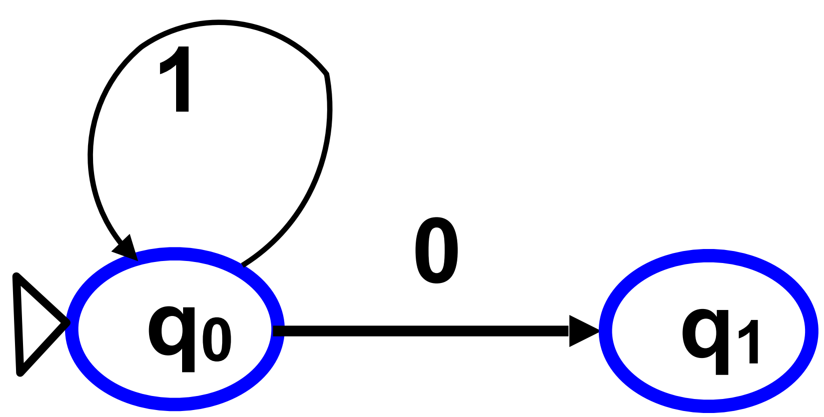
Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



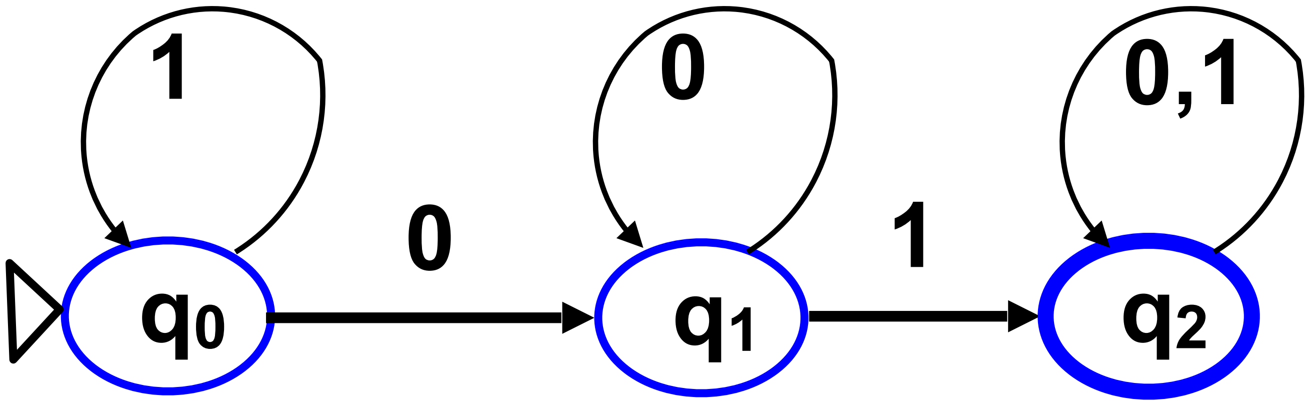
Il DFA per il complemento:



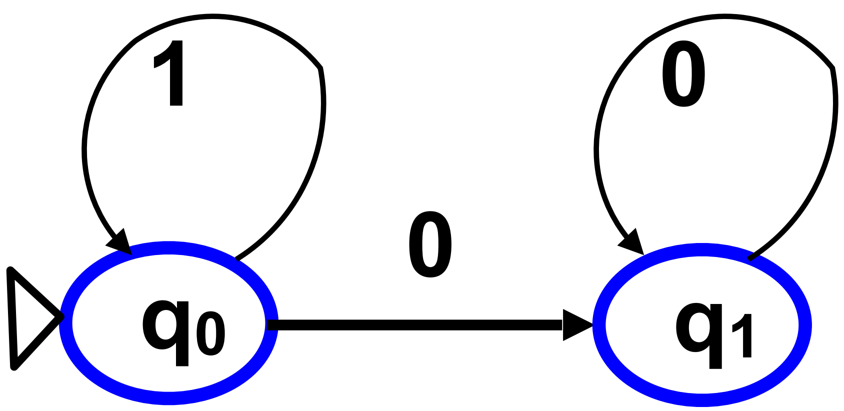
Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



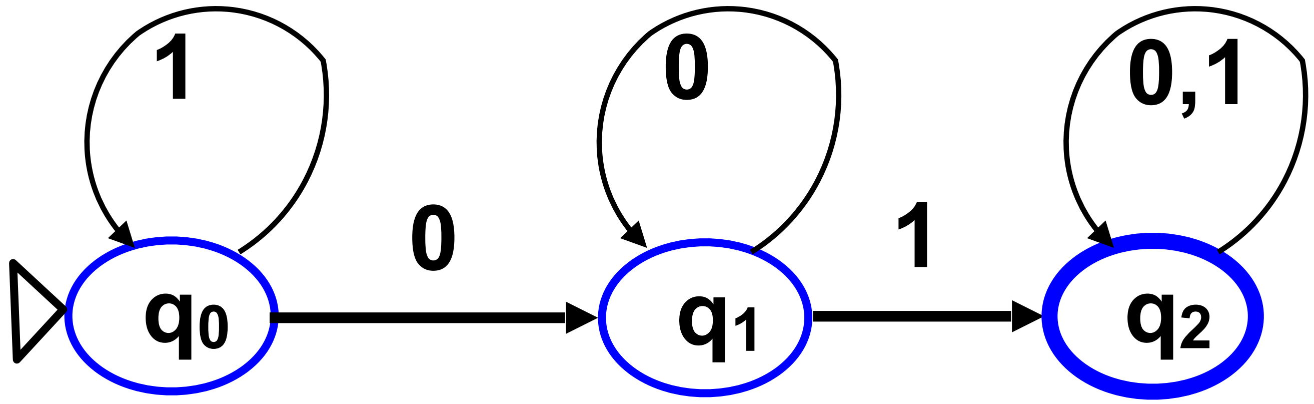
Il DFA per il complemento:



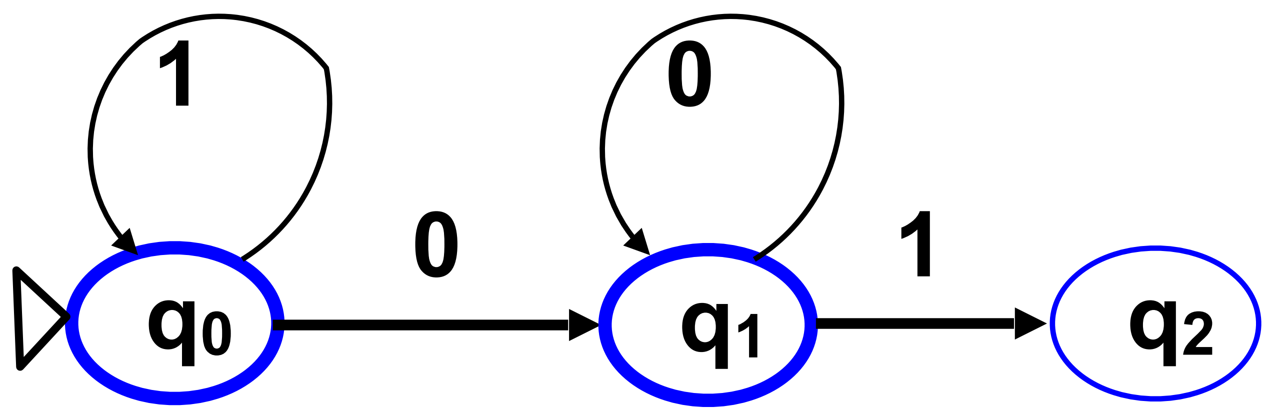
Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



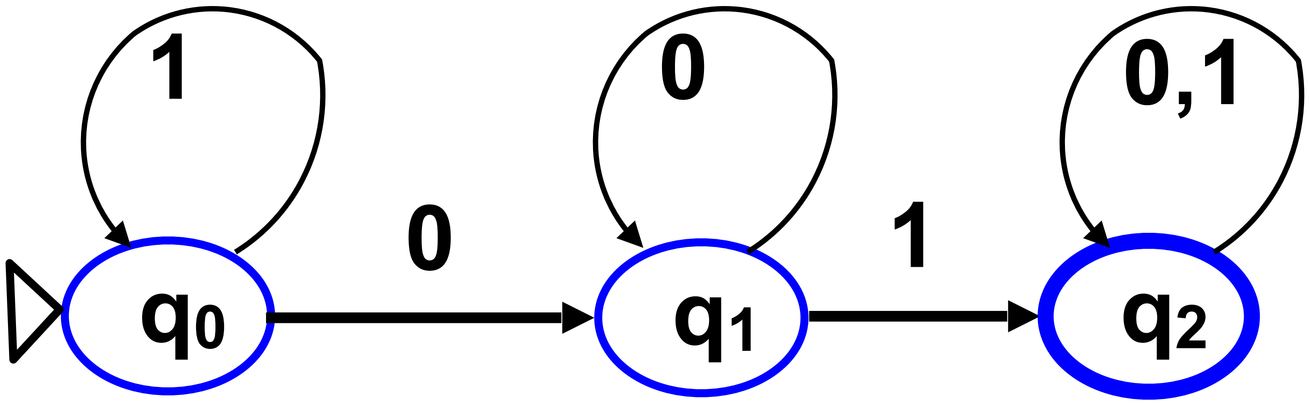
Il DFA per il complemento:



Esercizi

Si costruisca un DFA che accetta tutte le stringhe in {0,1}* che <u>non</u> contengono 01 come sottostringa.

un DFA per il linguaggio delle parole che contengono 01 come sottostringa:



Il DFA per il complemento: