AUTOMI A STATI FINITI

Capitolo 6 - Esercizi

Docente: Cataldo Musto

UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Esercizio 1

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | |w| dispari }

Esercizio 1

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | |w| dispari }

- **Schema di ragionamento**
  - **Generalmente ogni «caratteristica» della parola oppure ogni vincolo del linguaggio denota uno stato**
  - **Quali caratteristiche abbiamo in questo caso?**
    - **La lunghezza**
      - **Può essere pari**
      - **Può essere dispari**

Esercizio 1

- Costruire l’automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | |w| dispari }

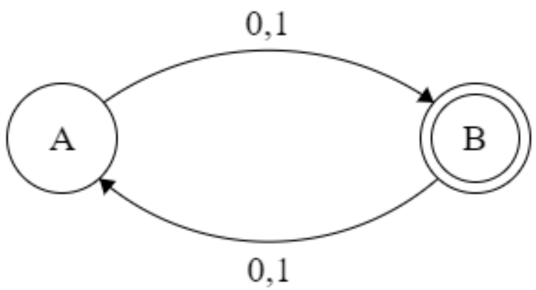
- **Schema di ragionamento**
  - **Generalmente ogni «caratteristica» della parola oppure ogni vincolo del linguaggio denota uno stato**
  - **Quali caratteristiche abbiamo in questo caso?**
    - **La lunghezza**
      - **Può essere pari**
      - **Può essere dispari**

Ci serve uno **stato dove «memorizzare» l’informazione che la lunghezza della parola è dispari** (stato finale!) e uno stato dove memorizziamo che la lunghezza è pari

Due stati in totale!

Esercizio 1

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | |w| dispari }



Si parte dallo stato A

Leggo un carattere (può essere 0/1)

Al primo carattere letto, vado nello stato B

(stato finale, perché la lunghezza è dispari)

Se ci sono altri caratteri, torno nello stato A (lunghezza pari)

E così via fino alla lettura completa della stringa. Se è dispari saremo in uno stato finale, altrimenti la parola non viene accettata.

Esercizio 2

- Costruire l’automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | w(i) = 1 per ogni i dispari}
    - **Es. 10101 = SI**
    - **Es. 111110 = SI**
    - **Es. 0111 = NO**

Esercizio 2

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | w(i) = 1 per ogni i dispari}

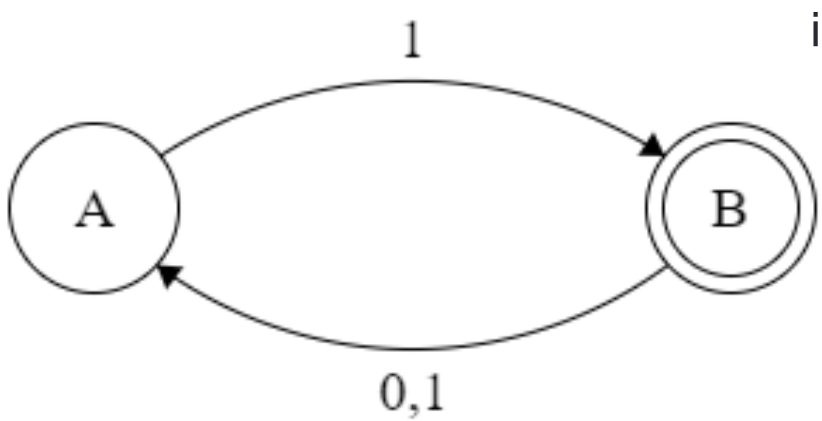
    - **Es. 10101 = SI**
    - **Es. 111110 = SI**
    - **Es. 0111 = NO**

- In questo caso serve uno stato che rappresenti le posizioni dispari e uno stato che rappresenti le posizioni pari
  - Nelle posizioni pari devo leggere «1» per passare alle posizioni dispari
  - Nelle posizioni dispari posso leggere qualsiasi cosa

Esercizio 2

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | w(i) = 1 per ogni i dispari}

    - **Es. 10101 = SI**
    - **Es. 111110 = SI**
    - **Es. 0111 = NO**



Soluzione

- A = stato che rappresenta i simboli in posizione pari (0,2,4 etc.)
- B = stato che rappresenta i simboli in posizione dispari (1,3,5…)

Esercizio 3

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | il penultimo simbolo è zero}

    - **Es. 10101 = SI**
    - **Es. 111110 = NO**
    - **Es. 00111 = NO**

Suggerimento: ci serve uno stato dove «memorizzare» che il penultimo simbolo letto è per forza uno zero

Esercizio 3

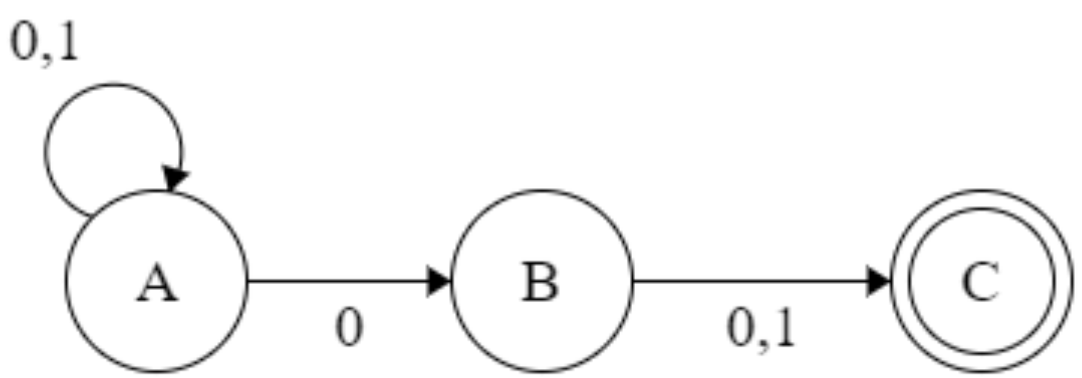
- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | il penultimo simbolo è zero}
    - **Es. 10101 = SI**
    - **Es. 111110 = NO**
    - **Es. 00111 = NO**

Posso sapere a priori, mentre leggo dei caratteri, se il simbolo è il penultimo o no?

NO, Quindi l'automa sarà non deterministico!

Esercizio 3

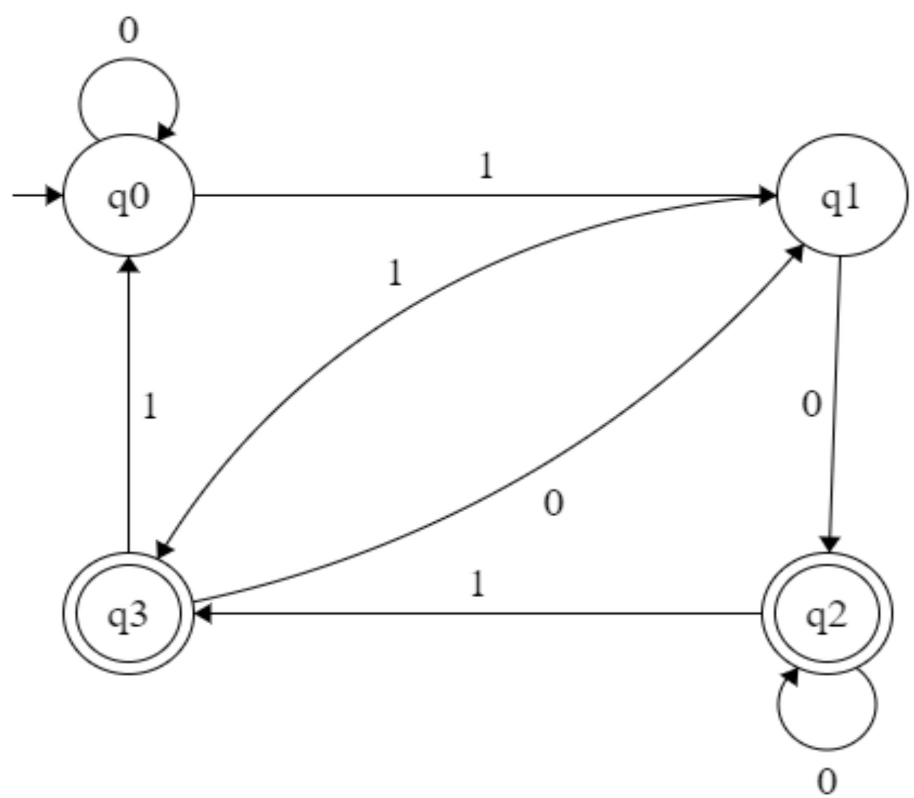
- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | il penultimo simbolo è zero}
    - **Es. 10101 = SI**
    - **Es. 111110 = NO**
    - **Es. 00111 = NO**



Convertirlo in Deterministico!

Esercizio 3

- Costruire l'automa a partire da una descrizione del linguaggio
  - L = {w in {0,1}* | il penultimo simbolo è zero}
    - **Es. 10101 = SI**
    - **Es. 111110 = NO**
    - **Es. 00111 = NO**



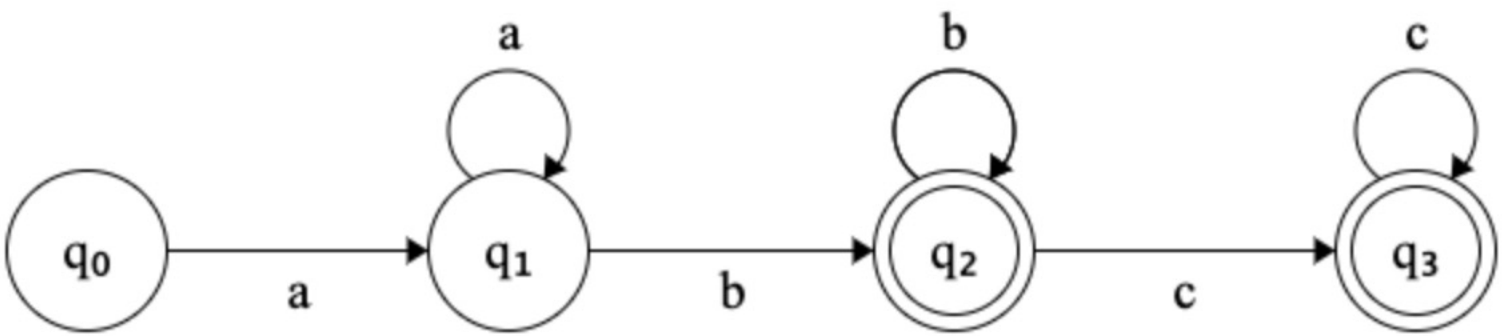
Deterministico!
Nota: gli stati sono stati mappati con nuove etichette

Esercizio 4

1 Data $\boldsymbol{r} = \mathtt{aa^*bb^*c^*}$, definire un DFA che accetti $\overline{\mathsf{L}(\boldsymbol{r})}$

Esercizio 4

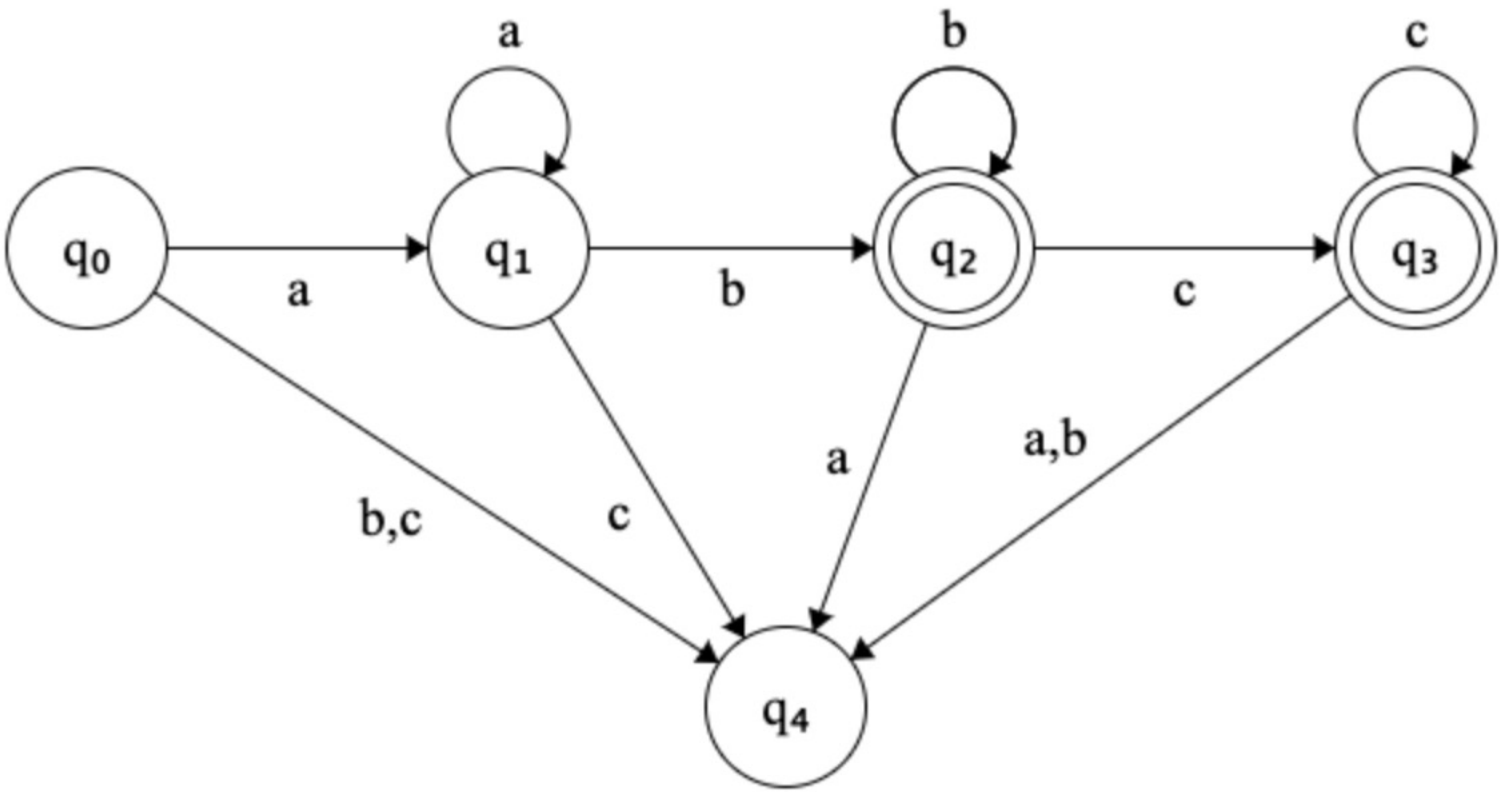
1 Data $r = \mathtt{aa^*bb^*c^*}$, definire un DFA che accetti $\overline{\mathsf{L}(r)}$



Automa Che Riconosce il Linguaggio

Esercizio 4

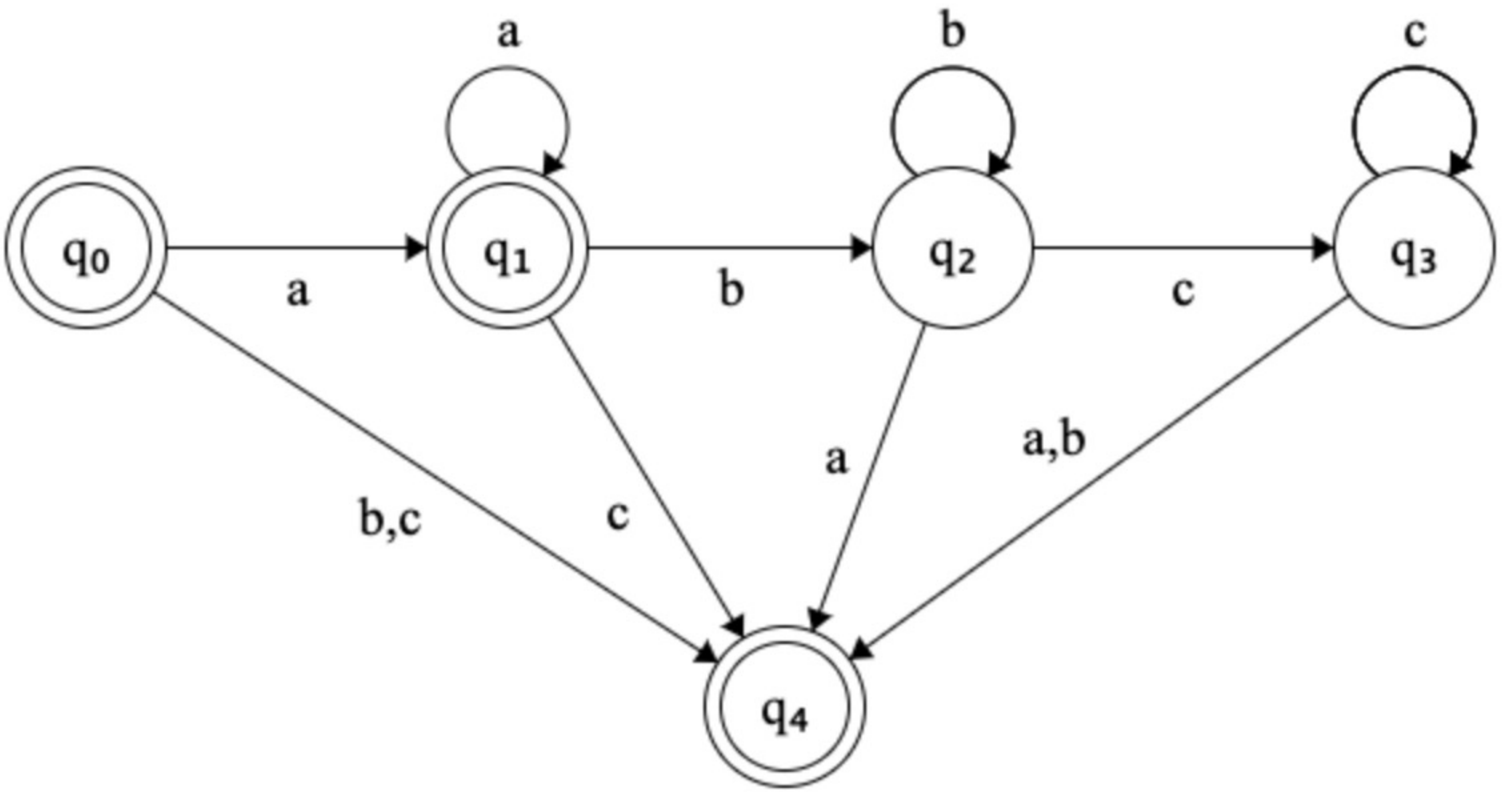
1 Data $r = aa^*bb^*c^*$, definire un DFA che accetti $\overline{L(r)}$



Automa Completo con Stato Pozza

Esercizio 4

1 Data $\boldsymbol{r} = \mathtt{aa^*bb^*c^*}$, definire un DFA che accetti $\overline{\mathsf{L}(\boldsymbol{r})}$



Automa Complemento

Esercizio 5

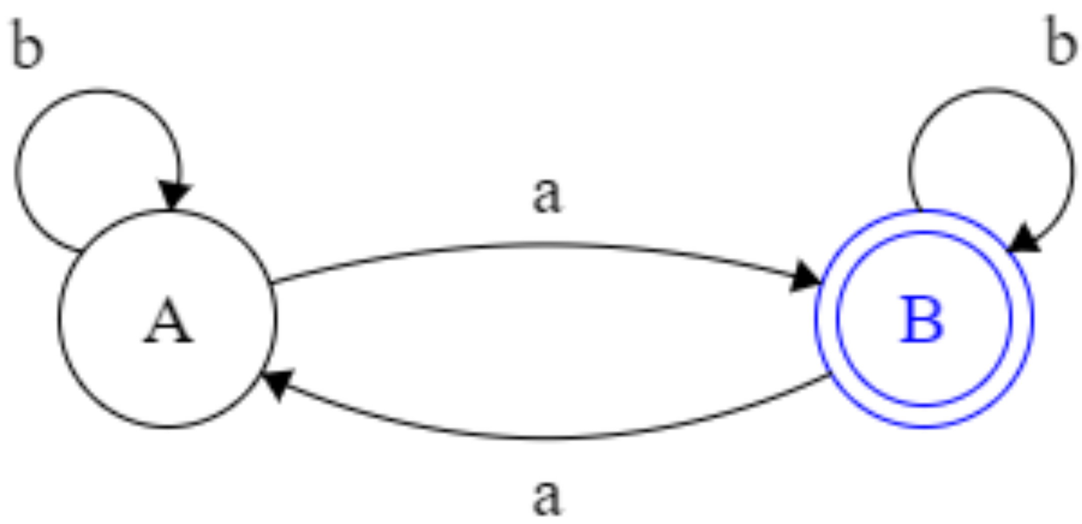
1: Definire un'espr. reg. per $\{w \in \{\mathrm{a}, \mathrm{b}\}^* \mid [n_{\mathrm{a}}(w)/2] \neq 0\}$

(numero di a dispari)

Esercizio 5

1: Definire un'espr. reg. per $\{w \in \{a, b\}^* \mid [n_a(w)/2] \neq 0\}$

(numero di a dispari)



Esercizio 5

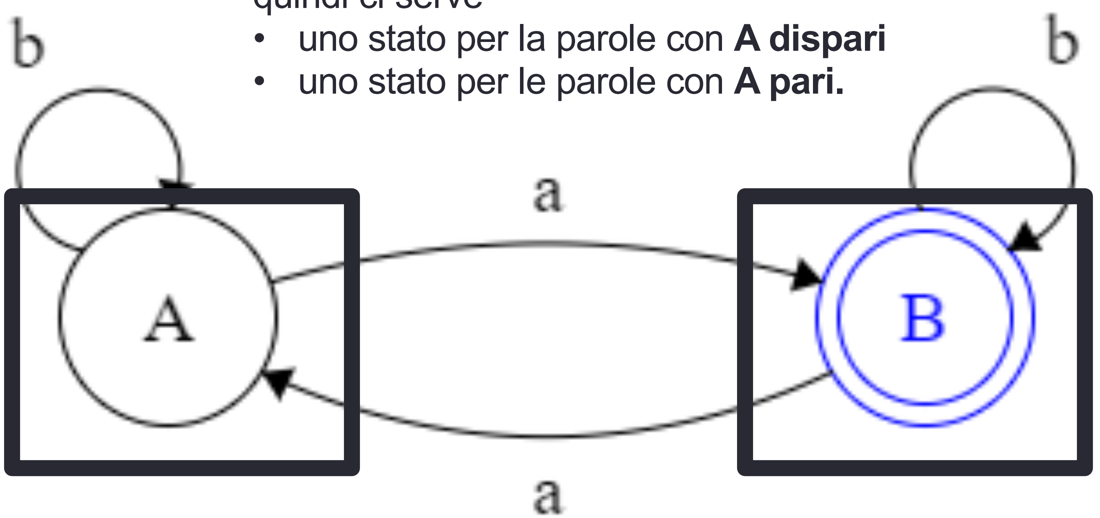
1: Definire un'espr. reg. per $\{w \in \{a, b\}^* \mid [n_a(w)/2] \neq 0\}$

(numero di a dispari)

Il numero di stati necessari dipende dalle caratteristiche delle parole del Linguaggio,

quindi ci serve

- uno stato per la parole con **A dispari**
- uno stato per le parole con **A pari.**

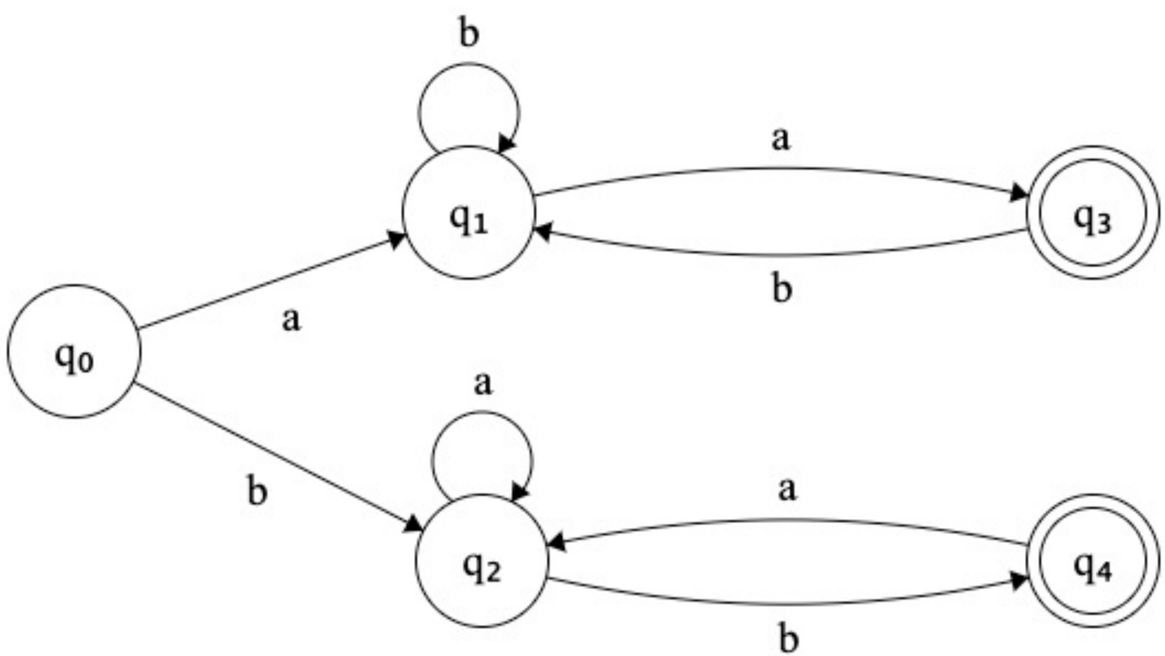


Esercizio 6

Costruire un automa che riconosca il linguaggio delle stringhe w, |w|>1 che iniziano e finiscono con lo stesso simbolo (supponiamo alfabeto X = **{a,b}**)

Esercizio 6

Costruire un automa che riconosca il linguaggio delle stringhe w, |w|>1 che iniziano e finiscono con lo stesso simbolo (supponiamo alfabeto X = **{a,b}**)

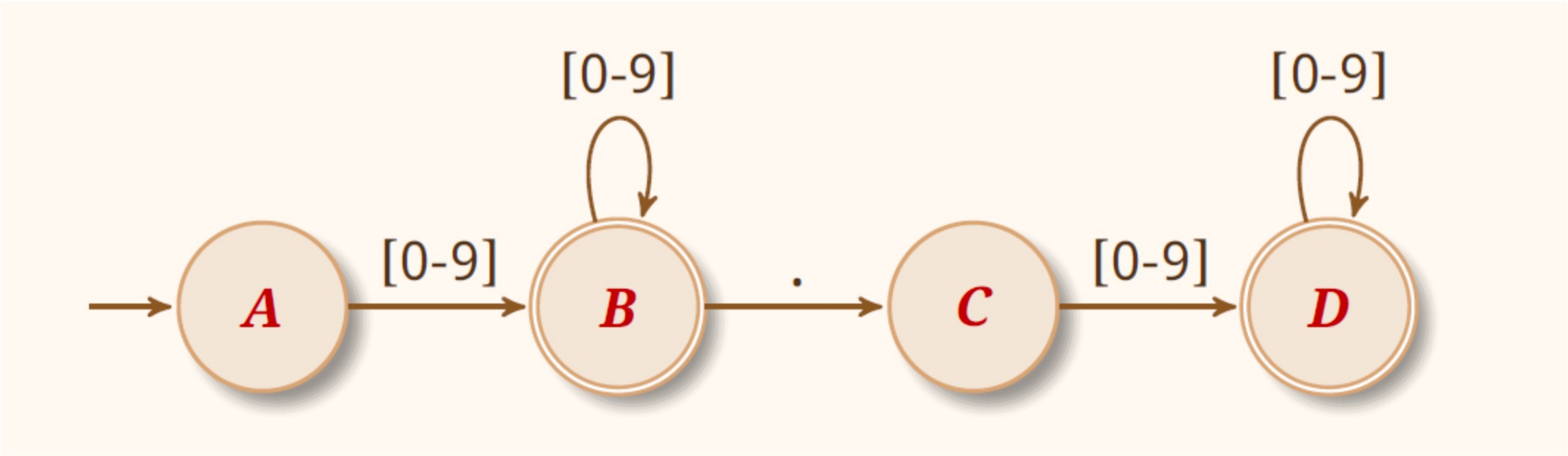


Esercizio 7

Costruire un automa che riconosce il linguaggio dei numeri con la virgola seguiti da cifre decimali **(es. 27.85 oppure 27.9992)**

Esercizio 7

Costruire un automa che riconosce il linguaggio dei numeri con la virgola seguiti da cifre decimali **(es. 27.85 oppure 27.9992)**

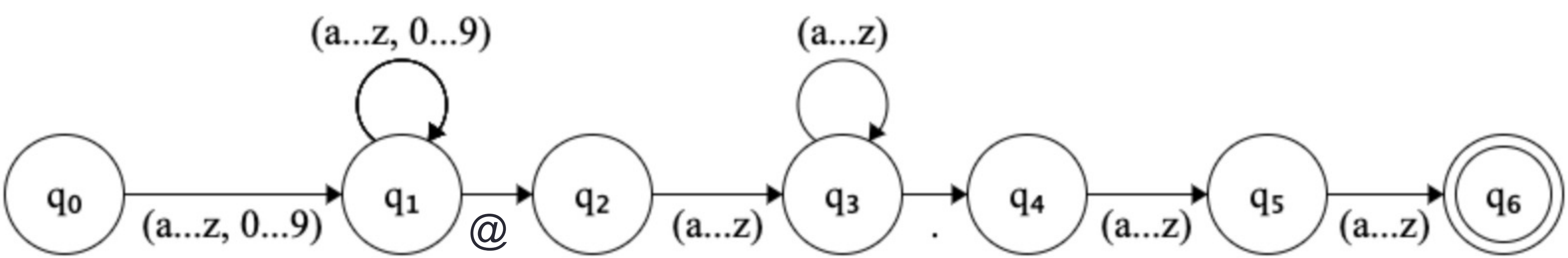


Esercizio 8

Costruire un automa che riconosca il linguaggio degli indirizzi e-mail, nella forma
[lettere e numeri]@[lettere].[dominio]
Per semplicità assumiamo il dominio senza numeri e con estensione di 2 caratteri (es «.it»)

Esercizio 8

Costruire un automa che riconosca il linguaggio degli indirizzi e-mail, nella forma
[lettere e numeri]@[lettere].[dominio]
Per semplicità assumiamo il dominio senza numeri e con estensione di 2 caratteri (es «.it»)



Esercizio 9

Costruire un automa che riconosca il linguaggio degli indirizzi e-mail, nella forma
[lettere e numeri]@[lettere].[dominio]
Per semplicità assumiamo il dominio senza numeri e con estensione di 2 caratteri (es «.it»)

Estendere l'esercizio con i seguenti vincoli
- L'indirizzo e-mail deve contenere almeno un numero
- Il dominio può essere di due o tre cifre

(Senza soluzione) <3

Esercizio 10

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

X mod 4 = 0
X in formato binario

→ Deve riconoscere 0, 4, 8, 12, 16, 20, etc.

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Analizziamo la struttura dei numeri binari

0	**0**	**SI**	**8**	**1000**	**SI**
1	1	NO	9	1001	NO
2	10	NO	10	1010	NO
3	11	NO	11	1011	NO
4	**100**	**SI**	**12**	**1100**	**SI**
5	101	NO	13	1101	NO
6	110	NO	14	1110	NO
7	111	NO	15	1111	NO
			16	**10000**	**SI … etc**

Quale caratteristica notiamo?

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Analizziamo la struttura dei numeri binari

0	**0**	**SI**	**8**	**1000**	**SI**
1	1	NO	9	1001	NO
2	10	NO	10	1010	NO
3	11	NO	11	1011	NO
4	**100**	**SI**	**12**	**1100**	**SI**
5	101	NO	13	1101	NO
6	110	NO	14	1110	NO
7	111	NO	15	1111	NO
			16	**10000**	**SI**
			20	**10100**	**SI**
			24	**11000**	**SI (etc.)**

Quale caratteristica notiamo?

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Analizziamo la struttura dei numeri binari

0	**0**	**SI**	**8**	**1000**	**SI**
1	1	NO	9	1001	NO
2	10	NO	10	1010	NO
3	11	NO	11	1011	NO
4	**100**	**SI**	**12**	**1100**	**SI**
5	101	NO	13	1101	NO
6	110	NO	14	1110	NO
7	111	NO	15	1111	NO
			16	**10000**	**SI**
			20	**10100**	**SI**
			24	**11000**	**SI (etc.)**

Quale caratteristica notiamo? Le stringhe valide terminano con un doppio 0.

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

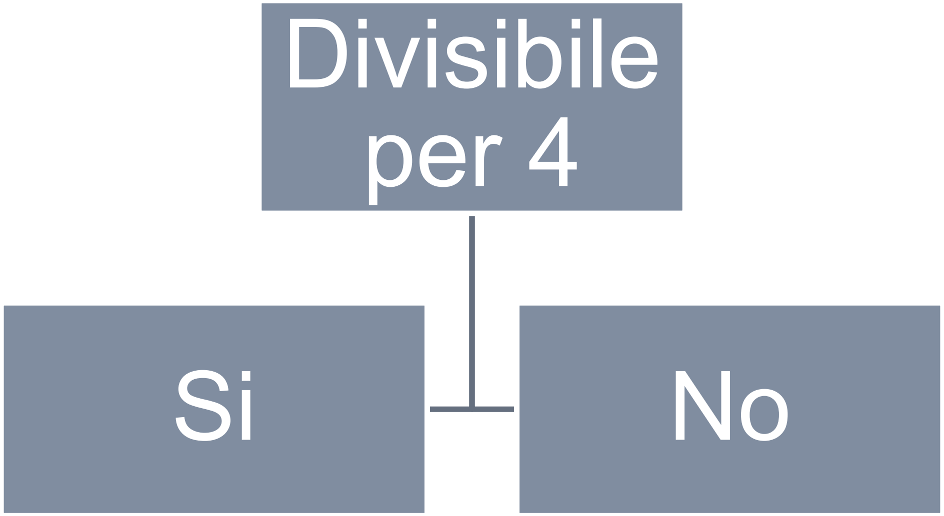
Suggerimento per la progettazione degli stati
Dividete in numeri in classi di equivalenza

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Suggerimento per la progettazione degli stati

Dividete in numeri in classi di equivalenza



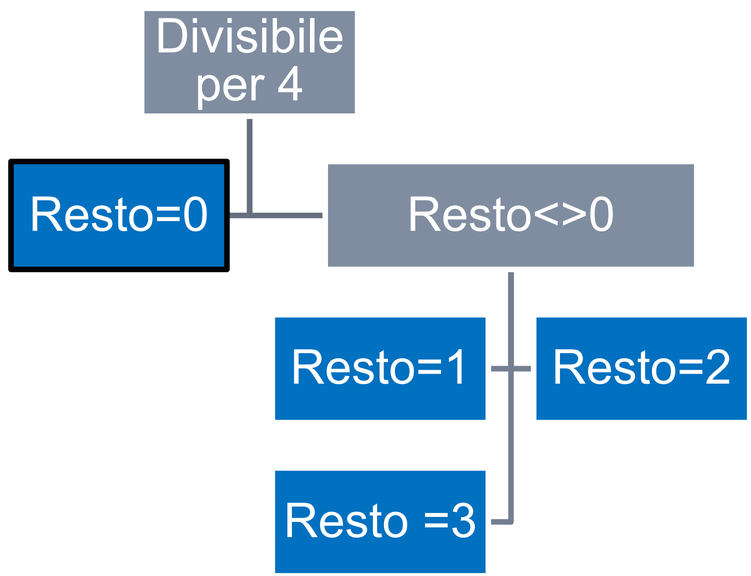
Partizioniamo ulteriormente la classe dei «no»

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Suggerimento per la progettazione degli stati

Dividete in numeri in classi di equivalenza



Questi sono gli stati dell'automa.

Ogni stato è mappato con una classe di equivalenza

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.

Suggerimento per la progettazione degli stati
Dividete in numeri in classi di equivalenza

Q_0 = stato per i numeri con resto 0 (stato finale)
Q_1 = stato per i numeri con resto 1
Q_2 = stato per i numeri con resto 2
Q_3 = stato per i numeri con resto 3

Esercizio 9

Costruire un automa che riconosca il linguaggio dei numeri binari divisibili per 4.