



UNIVERSITÀ
DEGLI STUDI
DI TORINO

Automati a stati finiti deterministici

a.a. 2018-2019

Automi, Linguaggi e Calcolabilità, J. E. Hopcroft, R. Motwani, J. D. Ullman

Automi a stati finiti [cap. 2]

2.1 Una descrizione informale degli automi a stati finiti. (leggere).

2.2 Automi a stati finiti deterministici.

2.3 Automi a stati finiti non deterministici.

(Tranne Esempio 2.9)

2.4 Un'applicazione: ricerche testuali. (leggere).

2.5 Automi a stati finiti con epsilon-transizioni.

(Teorema 2.22 solo enunciato, senza dimostrazione.)

Come descrivere i lessemi e come riconoscerli?

- a) Ogni categoria sintattica è formata da lessemi, che soddisfano un pattern su un alfabeto



Insiemi di stringhe



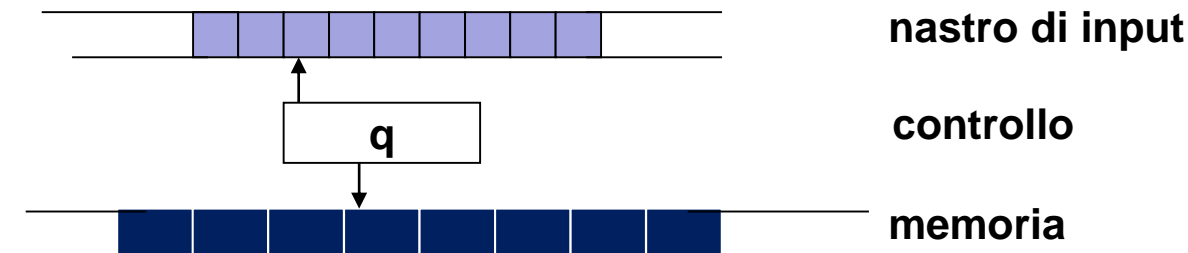
Linguaggio

- b) La descrizione dei lessemi (il pattern) è fornita da espressioni regolari o altre forme di descrizione come le grammatiche
- c) Il riconoscimento è fatto per mezzo di una macchina (un algoritmo che la implementa)

Automati con un numero finito di stati

Tutte le volte che si effettua una query, si usa un text editor, si controlla la correttezza di una frase e la si traduce, serve un algoritmo di riconoscimento: riconosce se una stringa è corretta rispetto a una definizione formale di riferimento.

Il tipo più generale e più potente di automa è la Macchina di Turing, in grado di riconoscere i linguaggi ricorsivamente enumerabili.



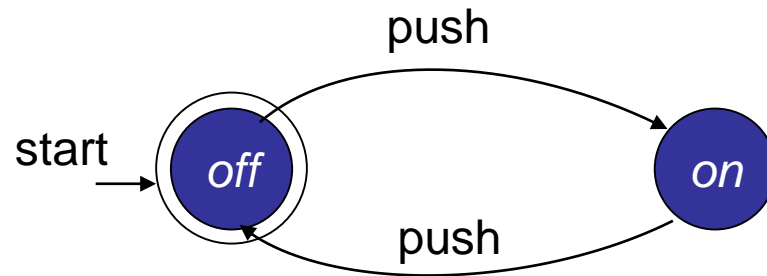
Il controllo in ogni istante è in uno stato che fa parte di un insieme finito di stati.

Vedremo automi meno potenti della macchina di Turing che differiscono da questa e tra di loro per il modo in cui è organizzata la memoria e per le funzioni (lettura, scrittura) e gli spostamenti che la testina può effettuare sul nastro di input.

Esempio:

Un semplice interruttore ha due stati, *on* e *off*.

Possiamo rappresentarlo con un diagramma di questo tipo:



- C'è uno stato iniziale “*off*”
- Un'azione sull'interruttore (*push*) fa cambiare lo stato (*transizione*).
- C'è uno stato “finale”, ancora “*off*”, quando le operazioni compiute sono finite.

Quando l'automa deve riconoscere le parole di un linguaggio, le azioni corrispondono alla lettura dei simboli di una stringa sul nastro di input; l'automa inizia a leggere con il controllo nello stato iniziale e, se il controllo si trova in uno stato finale quando la lettura termina, la risposta è “*si*”, in caso contrario la risposta è “*no*”.

Caratteristiche

- La testina sul nastro di input può solo leggere e avanzare di una posizione per volta verso destra.
- La memoria è assente.
- La macchina non è in grado di ricordare se non per mezzo degli stati.

Ad esempio, cosa gli stati possono ricordare se l'automa deve rispondere “*si*” quando esamina:

- una stringa sull'alfabeto $\{0,1\}$ *che presenta almeno un'occorrenza della sottostringa 011*, oppure
- una stringa sull'alfabeto $\{0,1,.\}$ del seguente *linguaggio di numeri binari*:

$$L = \{0\} \cup \{1\} \{0,1\}^* \{.\} \{0,1\}^+$$

Verso gli “automi a stati finiti”

Linguaggio delle stringhe su $\{0,1\}$ che hanno almeno un'occorrenza della sottostringa 011.

$$\Sigma = \{0,1\} \quad L = \{0,1\}^* \{011\} \{0,1\}^*$$

Gli stati servono a ricordare e distinguere i diversi casi che si possono presentare esaminando una stringa sull'alfabeto $\{0,1\}$:

1. La stringa esaminata non contiene 0
2. L'ultimo carattere è 0 e la sottostringa 011 non è ancora stata letta
3. Gli ultimi due caratteri sono 01
4. È stata trovata la sottostringa 011

Diagramma di transizione

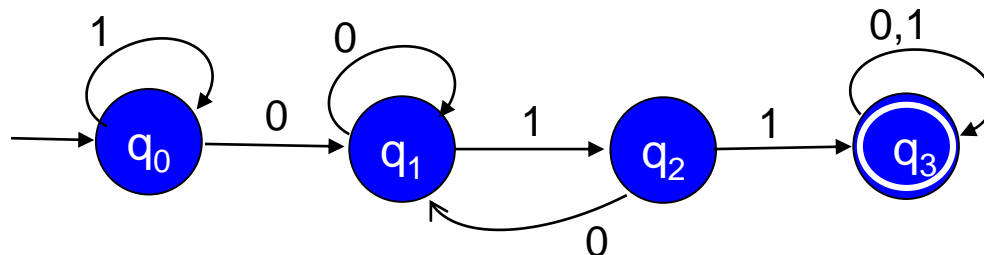
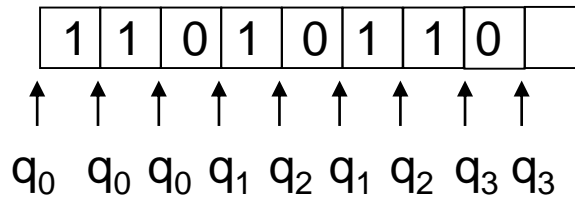
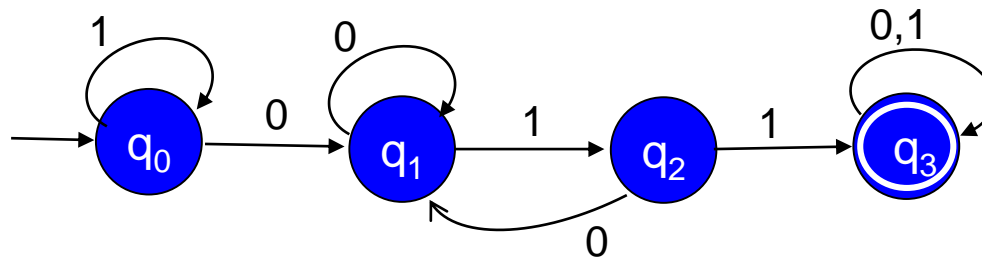


Diagramma di transizione



Nastro di input

Stati del controllo

Linguaggio dei numeri binari

$$\Sigma = \{0,1,.\}$$
$$L = (\{0\} \cup \{1\} \{0,1\}^*) \{.\} \{0,1\}^+$$

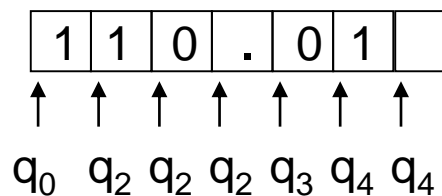
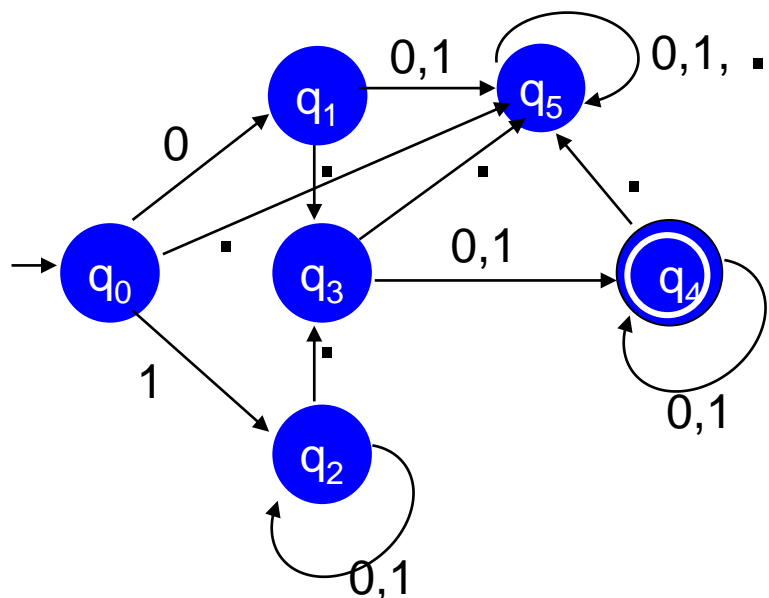
Quali casi si possono presentare esaminando una stringa sull'alfabeto $\{0,1,.\}$? Ad esempio 101.00 e 0.1 appartengono a L, mentre 101. e 0 non vi appartengono.

1. nessun simbolo è ancora stato esaminato
2. il primo carattere è 0
3. il primo carattere è 1
4. è stato incontrato il punto
5. è stato incontrato almeno un carattere dopo il punto
6. la stringa in esame non può appartenere al linguaggio

Verso gli “automi a stati finiti”

1. nessun simbolo è ancora stato esaminato (q_0 stato iniziale)
2. il primo carattere è 0 (q_1)
3. il primo carattere è 1 (q_2)
4. è stato incontrato il punto (q_3)
5. è stato incontrato almeno un carattere dopo il punto (q_4)
6. la stringa in esame non può appartenere al linguaggio (q_5)

Diagramma di transizione



Nastro di input

Stati del controllo

- Un DFA è una quintupla

$$A = (Q, \Sigma, \delta, q_0, F)$$

- Q è un insieme finito di stati
 - Σ è un alfabeto (simboli in input)
 - δ è una funzione di transizione $Q \times \Sigma \rightarrow Q$
 - $q_0 \in Q$ è lo stato iniziale
 - $F \subseteq Q$ è l'insieme degli stati finali
- Funzione di transizione estesa alle stringhe: $\hat{\delta}: Q \times \Sigma^* \rightarrow Q$

Definizione

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

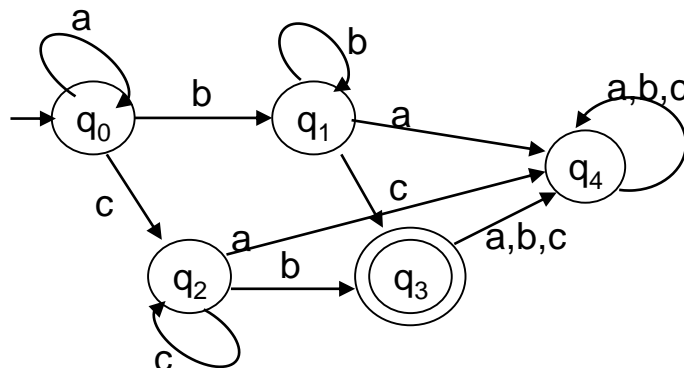
Automa finito deterministico: DFA

Definizione

$$\hat{\delta}(q, \varepsilon) = q$$

$$\hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$$

Esempio



$$\begin{aligned} \hat{\delta}(q_0, aabbc) &= \delta(\hat{\delta}(q_0, aabb), c) = \delta(\delta(\hat{\delta}(q_0, aab), b), c) = \\ &= \delta(\delta(\delta(\hat{\delta}(q_0, aa), b), b), c) = \delta(\delta(\delta(\delta(\hat{\delta}(q_0, a), a), b), b), c) = \\ &= \delta(\delta(\delta(\delta(\delta(\hat{\delta}(q_0, \varepsilon), a), a), b), b), c) = \delta(\delta(\delta(\delta(\delta(q_0, a), a), b), b), c) = \\ &= \delta(\delta(\delta(\delta(q_0, a), b), b), c) = \delta(\delta(\delta(q_0, b), b), c) = \delta(\delta(q_1, b), c) = \\ &= \delta(q_1, c) = q_3 \end{aligned}$$

Useremo spesso δ al posto di $\hat{\delta}$ in quanto si comportano nello stesso modo sulle stringhe di lunghezza 1 (caratteri dell'alfabeto).

- Formalmente, il linguaggio riconosciuto (o accettato) da A è

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$$

- I linguaggi riconosciuti (accettati) da automi a stati finiti sono chiamati linguaggi regolari.
- Due automi sono equivalenti se accettano lo stesso linguaggio.

Automa finito deterministico: DFA

Esempio

Se rappresentiamo l'azione di push con il simbolo “ p ” possiamo rappresentare l'interruttore con il seguente automa a stati finiti:

$$I = (\{on, off\}, \{p\}, \delta, off, \{off\})$$

$$\delta: \begin{cases} \delta(on, p) = off \\ \delta(off, p) = on \end{cases}$$

Diagramma di transizione

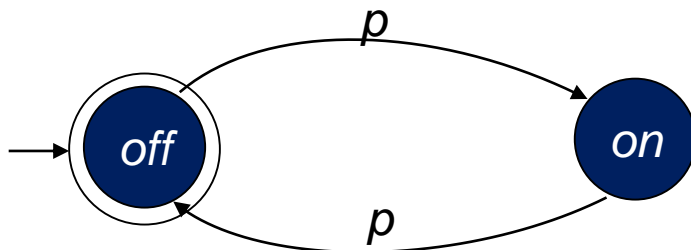


Tabella di transizione

	p
\rightarrow off	on
on	off

Il linguaggio riconosciuto sarà allora: $\{pp\}^*$.

Automa finito deterministico: DFA

Esempio

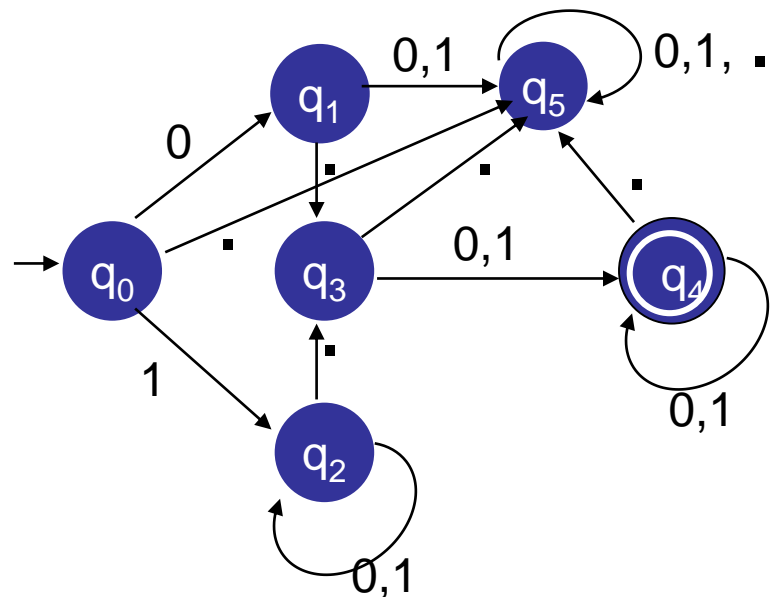
$M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1, \cdot\}, \delta, q_0, \{q_4\})$

$\delta(q_0, 0) = q_1$	$\delta(q_2, 0) = q_2$	$\delta(q_4, 0) = q_4$
$\delta(q_0, 1) = q_2$	$\delta(q_2, 1) = q_2$	$\delta(q_4, 1) = q_4$
$\delta(q_0, \cdot) = q_5$	$\delta(q_2, \cdot) = q_3$	$\delta(q_4, \cdot) = q_5$
$\delta(q_1, 0) = q_5$	$\delta(q_3, 0) = q_4$	$\delta(q_5, 0) = q_5$
$\delta(q_1, 1) = q_5$	$\delta(q_3, 1) = q_4$	$\delta(q_5, 1) = q_5$
$\delta(q_1, \cdot) = q_3$	$\delta(q_3, \cdot) = q_5$	$\delta(q_5, \cdot) = q_5$

Tabella di transizione

	0	1	\cdot
$\rightarrow q_0$	q_1	q_2	q_5
q_1	q_5	q_5	q_3
q_2	q_2	q_2	q_3
q_3	q_4	q_4	q_5
* q_4	q_4	q_4	q_5
q_5	q_5	q_5	q_5

Diagramma di transizione

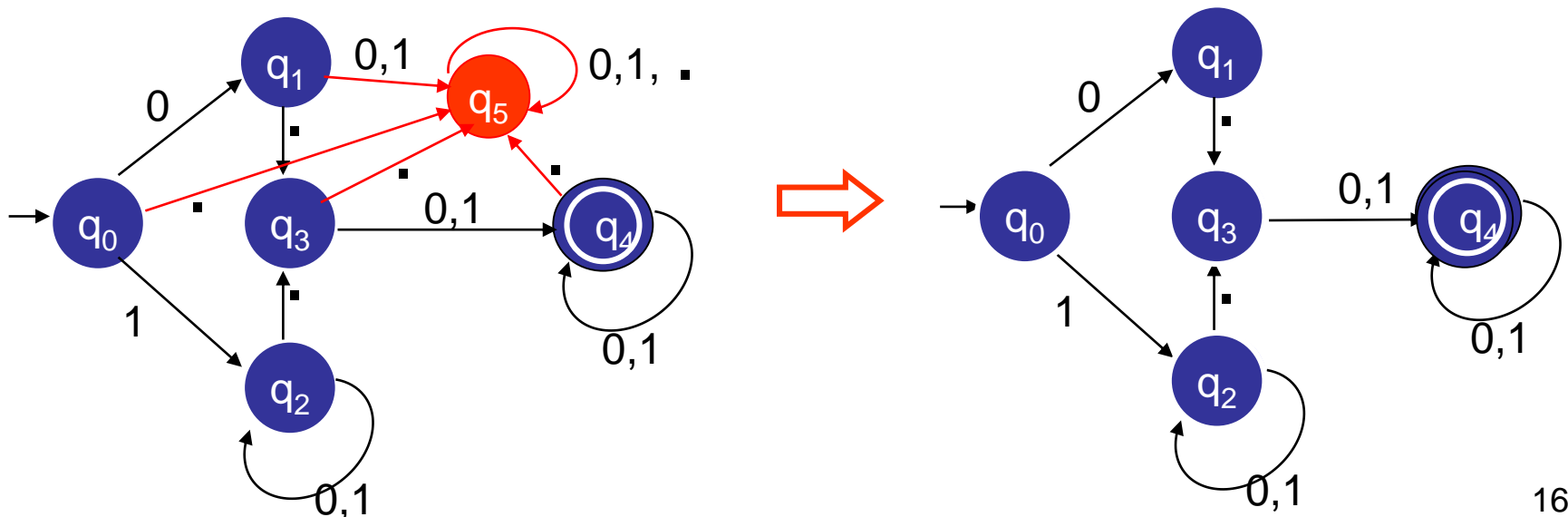


Automa finito deterministico: stato di errore

Una stringa viene accettata se, quando l'analisi termina, il controllo si trova in uno stato finale.

In alcuni casi, come nell'esempio precedente, l'automa prima di fermarsi può raggiungere uno stato in cui si può continuare la lettura della stringa, ma dal quale il controllo non può più uscire: stato di errore (o 'dead state', stato di morte, o 'stato trappola').

Lo stato di errore di solito viene *sottinteso* per non appesantire il diagramma di transizione o la tabella.



Automa finito deterministico: stato di errore

Esempio

$\langle \{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \delta, q_0, \{q_3\} \rangle$

$$\delta(q_0, a) = q_0$$

$$\delta(q_0, b) = q_1$$

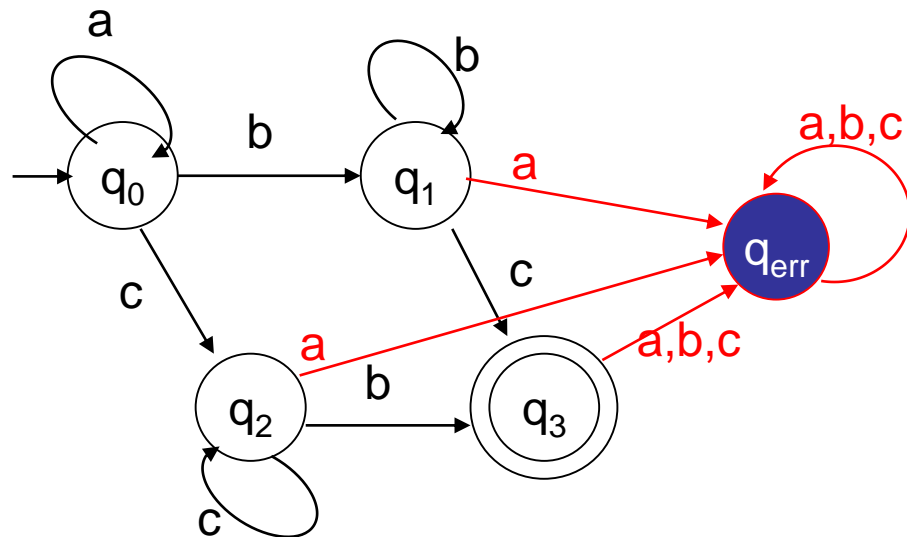
$$\delta(q_0, c) = q_2$$

$$\delta(q_1, b) = q_1$$

$$\delta(q_1, c) = q_3$$

$$\delta(q_2, c) = q_2$$

$$\delta(q_2, b) = q_3$$



Notare che $\delta(q_1, a)$, $\delta(q_2, a)$, $\delta(q_3, a)$, $\delta(q_3, b)$, $\delta(q_3, c)$, non sono esplicitamente definite, la loro definizione è sottintesa, ma la funzione di transizione è sempre una funzione totale.

Si considerino i DFA con le seguenti tabelle di transizione:

δ	0	1
$\rightarrow A$	A	B
*B	B	A

δ	0	1
$\rightarrow *A$	B	A
*B	C	A
C	C	C

Per ognuno dei due DFA descrivere il linguaggio accettato sia informalmente sia per mezzo di un descrittore di insiemi.

Costruire un DFA per ognuno dei seguenti linguaggi sull'alfabeto $\{0,1\}$:

- Insieme di tutte le stringhe che finiscono con 00.
- Insieme di tutte le stringhe con tre zero consecutivi.
- Insieme delle stringhe con 011 come sottostringa.
- Insieme delle stringhe che cominciano o finiscono (o entrambe le cose) con 01.
- Insieme di tutte le stringhe che iniziano con almeno due 0 e terminano con almeno due 1.
- Insieme delle stringhe che contengono un numero di 1 divisibile per 5.

Costruire un DFA per ognuno dei seguenti linguaggi sull'alfabeto $\{a, b\}$:

- $\{a^n b^n \mid 0 \leq n \leq 3\}$.
- Insieme di tutte le stringhe che non presentano b consecutivi e in cui tutte le sequenze di a siano di lunghezza pari.
- Insieme delle stringhe che contengono un numero di a multiplo di tre.

Disegnare il diagramma di transizione di automi finiti deterministici che riconoscano i linguaggi sull'alfabeto $\{a, b\}$ che soddisfano una delle seguenti condizioni:

- ogni occorrenza del carattere a sia seguita immediatamente da almeno due occorrenze del carattere b .
- ogni occorrenza del carattere a sia seguita immediatamente da esattamente due occorrenze del carattere b .
- inizino con almeno due a e terminino con almeno due b .