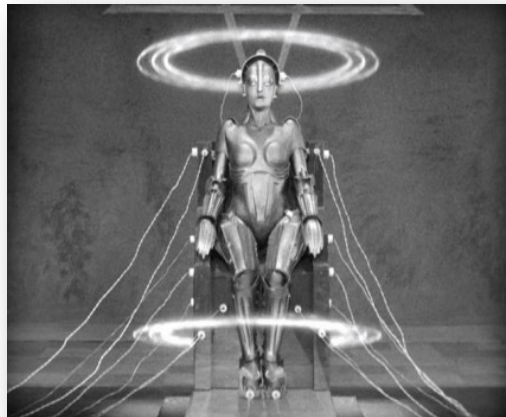




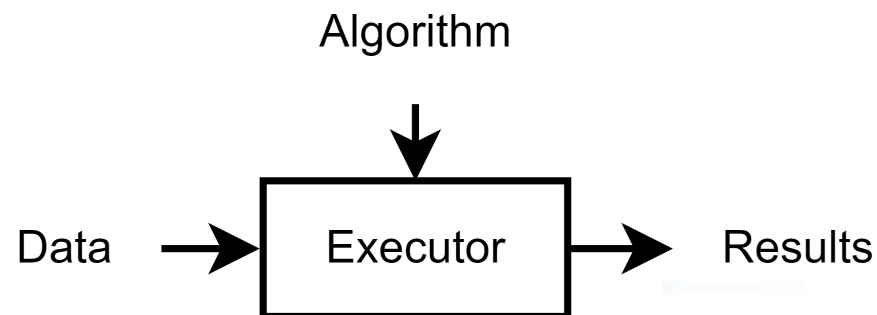
UNIVERSITÀ
DI PARMA

automi

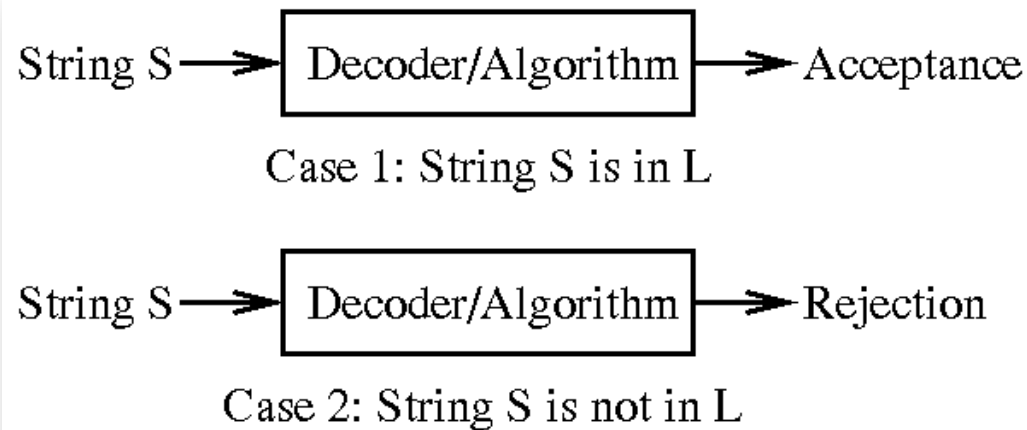


alberto ferrari - fondamenti di informatica

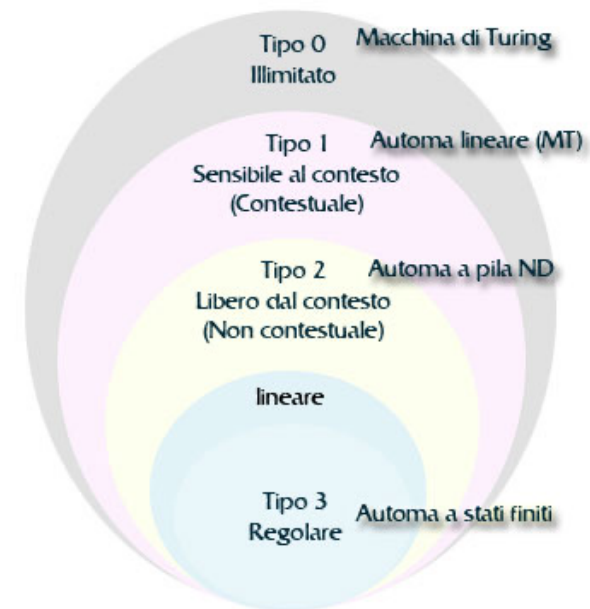
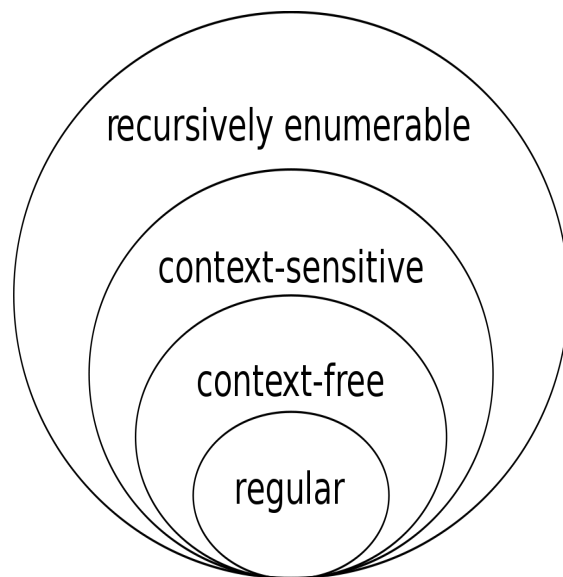
- ***automa***: macchina astratta
- realizza un certo algoritmo, secondo un modello di calcolo
- algoritmo definito nel “*linguaggio macchina*” dell'automa
- riceve ed elabora dei dati di ingresso



- ***riconoscimento*** di linguaggi
 - problema dell'appartenenza (*membership*)
 - data una stringa x , stabilire se essa appartiene ad L



linguaggi e automi



- linguaggi di **tipo 3 (regular)**
 - riconosciuti da **automi a stati finiti** (Finite State Machine)
 - es.: $\{a^n b : n \geq 0\}$ generato da $S \rightarrow aS \mid b$
- linguaggi di **tipo 2 (context-free)**
 - ric. da **automi a pila non deterministici** (Nondeterministic PushDown Automata)
 - es.: $\{a^n b^n : n \geq 1\}$ generato da $S \rightarrow aSb \mid ab$
- linguaggi di **tipo 1 (context-sensitive)**
 - riconosciuti da **automi limitati linearmente** (Linear Bounded Automata)
 - Es.: $\{a^n b^n c^n : n \geq 1\}$
- linguaggi di **tipo 0 (recursively enumerable)**
 - riconosciuti da **macchine di Turing** (Turing Machine)
 - $x \notin L$, semidecidibile: il processo può non terminare!

automi e linguaggi

MACCHINE (AUTOMI) A STATI FINITI

esempi di automi a stati finiti

- ***semafori***

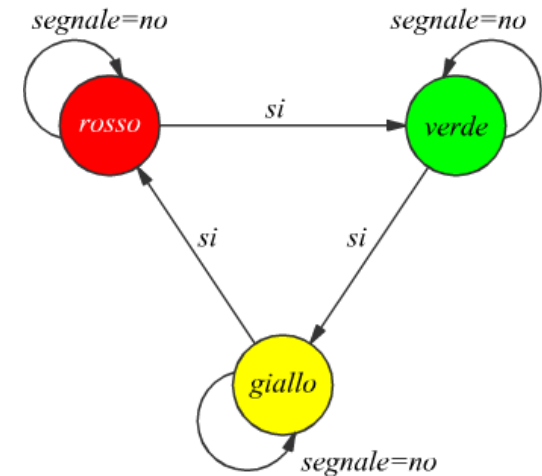
- insieme di stati (verde, giallo, rosso)
- il passaggio da uno stato a un altro avviene in base a un timer o a un sensore

- ***distributori automatici***

- passano tra stati in base agli input degli utenti
 - inserimento di monete, selezione e erogazione del prodotto
 - ogni input produce un cambiamento di stato

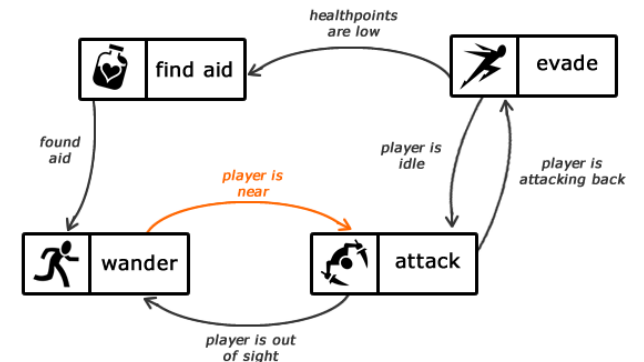
- ***ascensori***

- vari stati (piani)
- gli input dei pulsanti e le condizioni delle porte stabiliscono il passaggio da uno stato a un altro



personaggi non giocanti

- i personaggi non giocanti (PNG) sono figure in un videogioco o in un gioco di ruolo che non sono controllate direttamente da un giocatore, ma dal game master o dall'intelligenza artificiale.
- il "cervello" di un nemico in un videogame può essere implementato utilizzando un automa a stati finiti
 - ogni stato rappresenta un'azione
 - attaccare, eludere, trovare aiuto, muovere
 - nel grafo i nodi sono gli stati
 - i collegamenti sono le transizioni definiti dalle situazioni di gioco
 - player is near (il giocatore è vicino)
 - healthpoints are low (i punti salute sono bassi) ...



macchina a stati finiti (FSM)

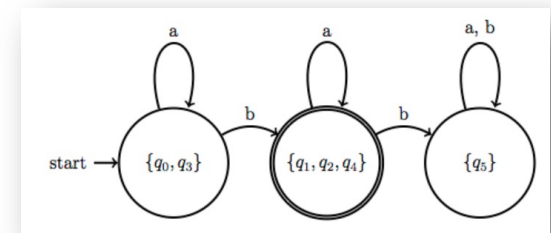
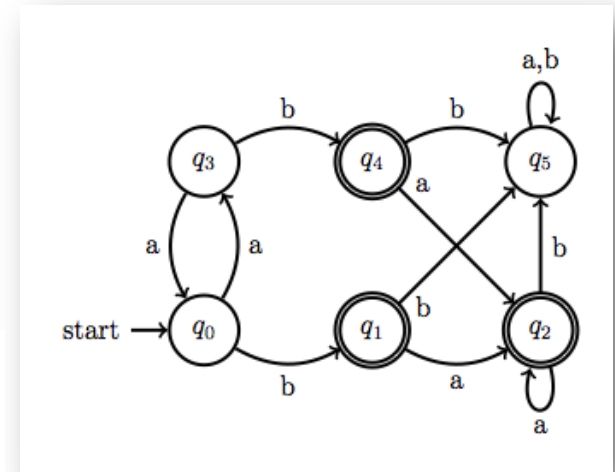
- automa a stati finiti (Finite State Machine) definito come M
 - $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ dove:
 - $\Sigma = \{\sigma_1, \dots, \sigma_n\}$: **alfabeto** di input
 - $Q = \{q_0, \dots, q_n\}$: insieme finito non vuoto di **stati**
 - $F \subseteq Q$: insieme di stati **finali**
 - $q_0 \in Q$: stato **iniziale**
 - $\delta: Q \times \Sigma \rightarrow Q$: **funzione di transizione**
 - in base allo stato e al simbolo di input attuali ...
 - determina lo stato successivo

FSM riconoscono tutti e soli i linguaggi regolari

esempio

- $M = \langle \Sigma, Q, \delta, q_0, F \rangle$
- $\Sigma = \{a, b\}$: alfabeto di input
- $Q = \{q_1, q_2, q_3, q_4, q_5\}$: insieme degli stati
 - $\{q_1, q_2, q_4\} \subseteq Q$: insieme di stati finali
 - $q_0 \in Q$: stato iniziale
- $\delta: Q \times \Sigma \rightarrow Q$: funzione di transizione
 - in base allo stato e al simbolo di input attuali
 - determina lo stato successivo
 - rappresentata da:
 - tabella di transizione
 - diagramma degli stati

δ	a	b
$\rightarrow q_0$	q_3	q_1
$*q_1$	q_2	q_5
$*q_2$	q_2	q_5
q_3	q_0	q_4
$*q_4$	q_2	q_5
q_5	q_5	q_5



esempio

$M = \langle \{a, b\}, \{qS, qA, qB, qC\}, \delta, qS, \{qS\} \rangle$

δ	a	b
qS	qA	qB
qA	qS	qC
qB	qC	qS
qC	qB	qA

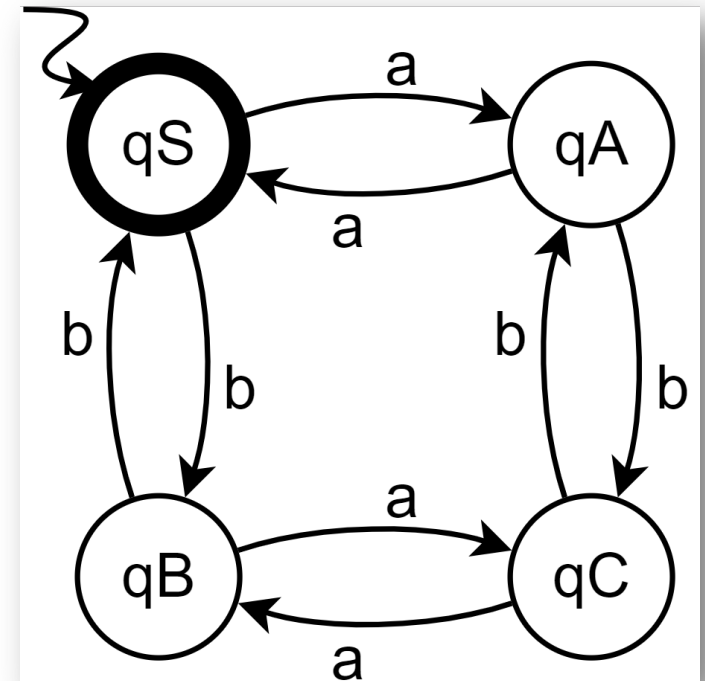
grammatica equivalente:

$S \rightarrow aA \mid bB \mid \varepsilon$

$A \rightarrow aS \mid bC$

$B \rightarrow aC \mid bS$

$C \rightarrow aB \mid bA$



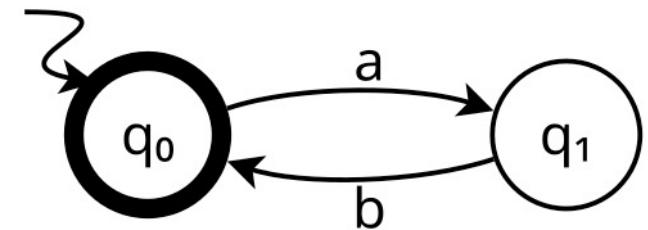
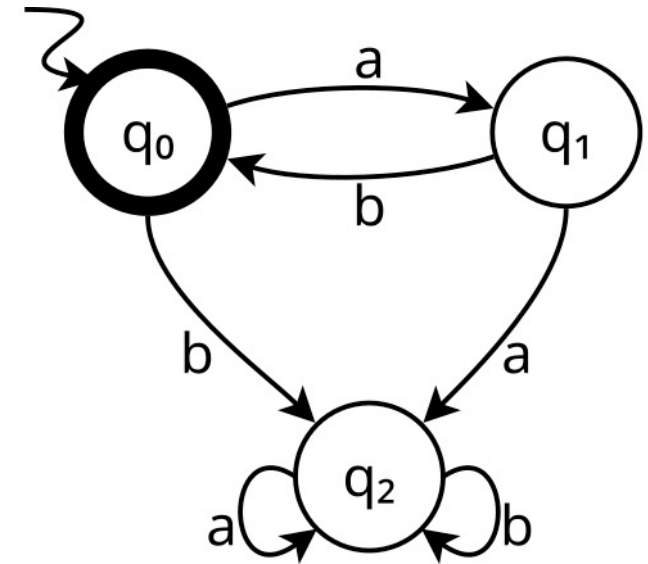
quali fra seguenti stringhe sono accettate? **aaabbb**, **abab**, **aabb**, **babba**

https://fondinfo.github.io/play/?c19_fsm.py

Stringhe con a in numero pari e b in numero pari

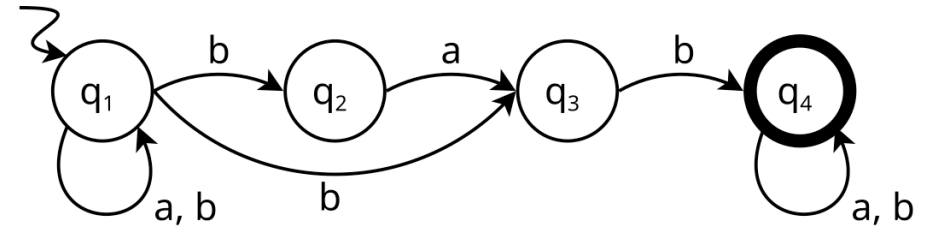
stati trappola

- se nella tabella di transizione una cella è vuota la computazione si blocca in quel punto
- la stringa viene rifiutata, perché l'automa non riesce a consumare tutto l'input
 - si può introdurre uno stato trappola che raccoglie tutte le transizioni mancanti: ogni cella vuota viene sostituita con una transizione verso lo stato trappola
- nell'automa di esempio in figura
- ***q2*** è uno ***stato trappola*** (*dead state*)
 - non esiste via d'uscita
 - qualsiasi sequenza successiva non è riconosciuta
- rappresenta spesso condizioni di errore
 - es. simbolo di input non previsto

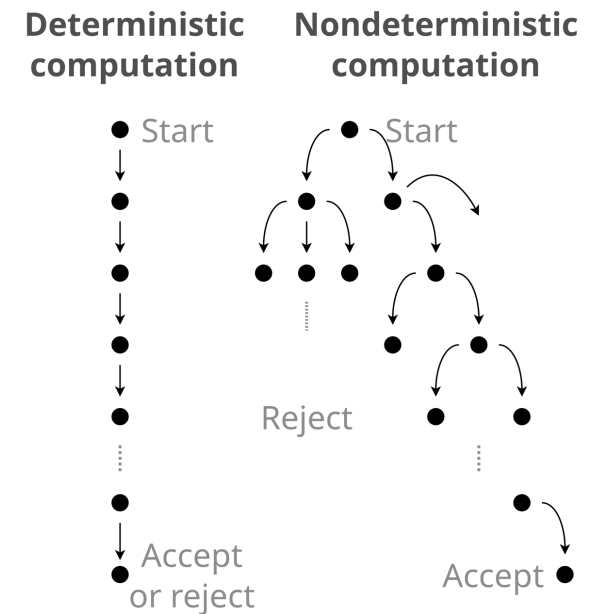
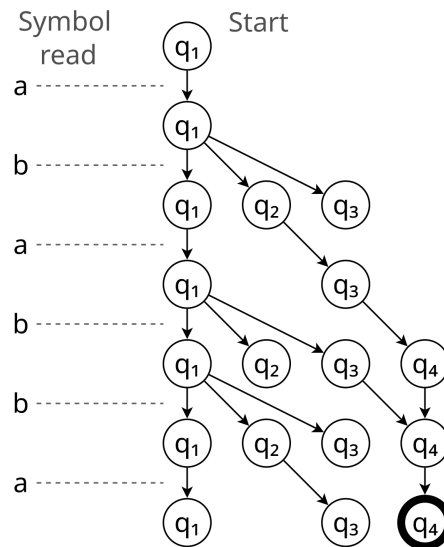


macchina a stati finiti non deterministica

- Nondeterministic Finite Automaton
- $M = \langle \Sigma, Q, \delta_N, q_0, F \rangle$
- $\Sigma = \{\sigma_1, \dots, \sigma_n\}$: alfabeto di input
- $Q = \{q_0, \dots, q_m\}$: insieme finito non vuoto di stati
- $F \subseteq Q$: insieme di stati finali
- $q_0 \in Q$: stato iniziale
- $\delta_N: Q \times \Sigma \rightarrow P(Q)$: funzione di transizione
 - determina ***insieme di stati successivi***
 - $P(Q)$ è l'insieme delle parti di Q , ossia l'insieme di tutti i possibili sottoinsiemi di Q



-



FSM - NFA

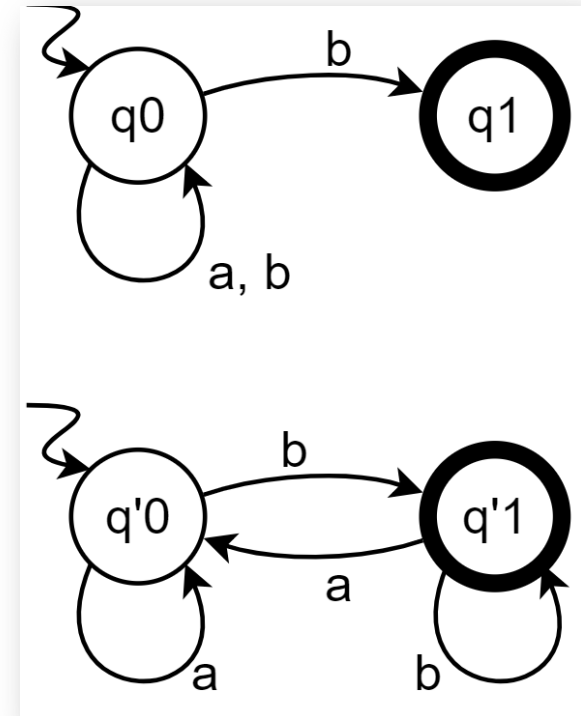
$M = \langle \{a, b\}, \{q_0, q_1\}, \delta, q_0, \{q_1\} \rangle$

δ	a	b
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{\}$	$\{\}$

$M' = \langle \{a, b\}, \{q'_0, q'_1\}, \delta', q'_0, \{q'_1\} \rangle$

δ'	a	b
q'_0	q'_0	q'_1
q'_1	q'_0	q'_1

Per ogni automa a stati finiti non deterministico è possibile costruire un automa a stati finiti deterministico in grado di riconoscere lo stesso linguaggio



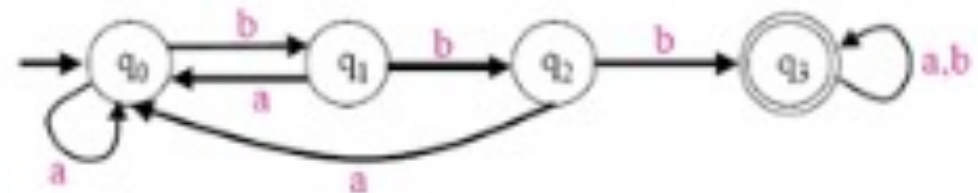
accetta qualsiasi stringa terminante con b

automi a stati finiti
esercizi



FSA - tabella di transizione

- costruire la tabella di transizione dell'automa rappresentato mediante diagramma degli stati
 - lo stato finale è q_3
- per ognuna delle seguenti stringhe dire se viene riconosciuta o meno dall'automa
 - abbab
 - bbabbaaa
 - ababbbab
 - aaabbb



FSA - riconoscimento

- l'automa con
 - alfabeto: {a, b}; stati: {q1, q2, q3, q4}; stato iniziale: q1; stati finali accettanti: {q4}
 - tabella di transizione:

	a	b
→q1	q2	q4
q2	q3	q4
q3	q4	q2
q4	q2	q1

- quali stringhe riconosce fra le seguenti?
 - baab, abbab, ba, bbaaa

NFSA - riconoscimento

- l'automa con
 - alfabeto: {a, b}; stati: {q1, q2, q3, q4}; stato iniziale: q1; stati finali accettanti: {q1, q2}
 - tabella di transizione:

	a	b
→q1	{q2,q3}	
q2	{q2,q3}	
q3		{q2,q4}
q4	{q2,q3}	

- quali stringhe riconosce fra le seguenti?
 - abba, ε, aabb, bbb, aa