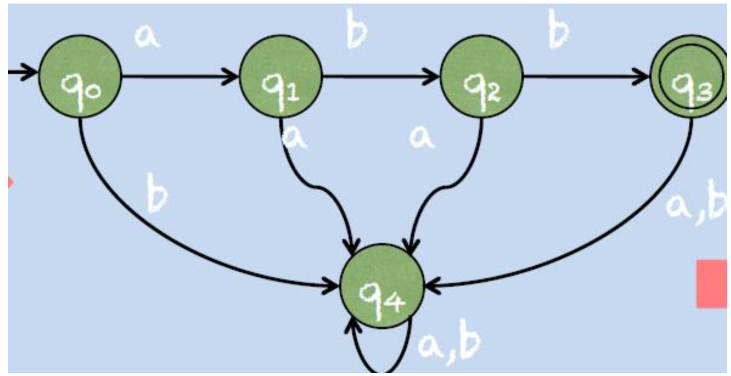
# La Macchina di Turing: computazione

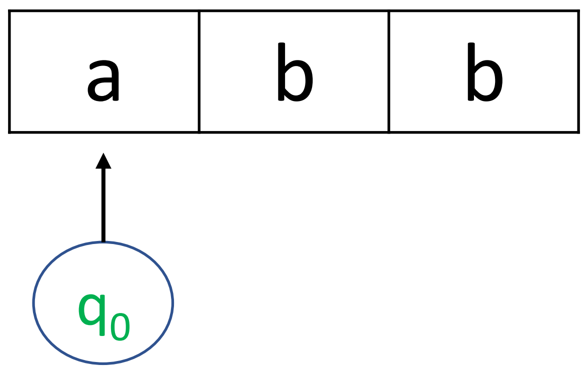
# Oggi

- Formalizzare il concetto di computazione di una MdT, di parola accettata e di linguaggio riconosciuto
- Obiettivo: definire il concetto di «procedura effettiva», «algoritmo»
- Partendo dal modello semplice di DFA

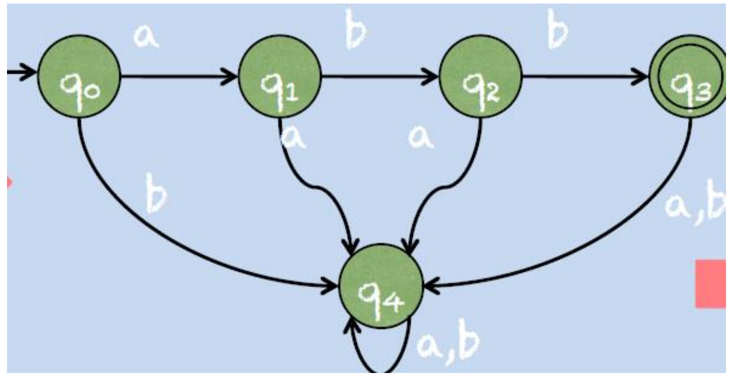
# Un DFA all’opera



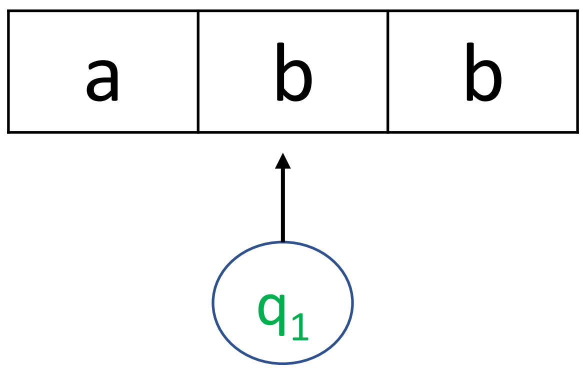

Input string: abb



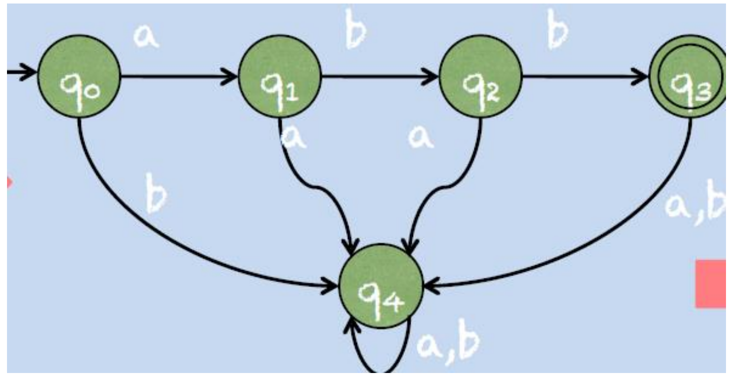
# Un DFA all’opera



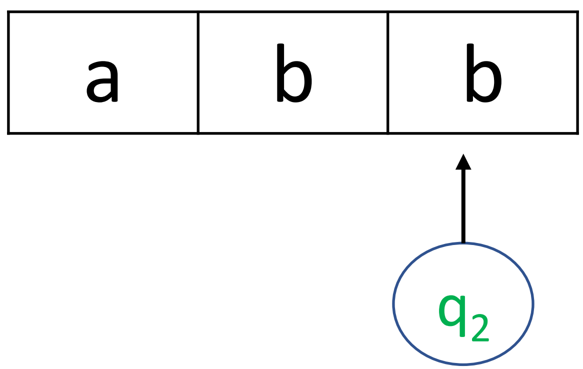

Input string: abb



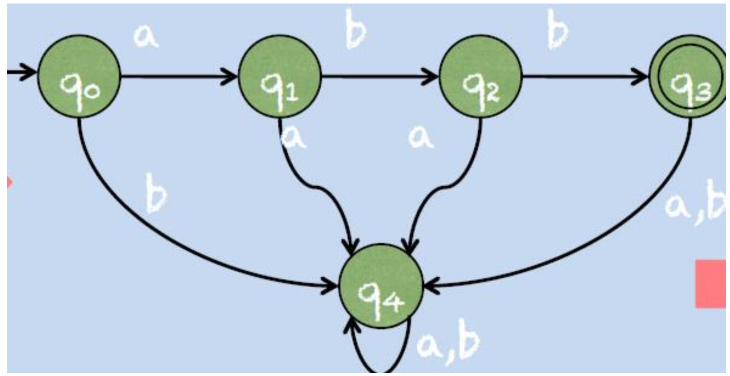
# Un DFA all’opera




Input string: abb

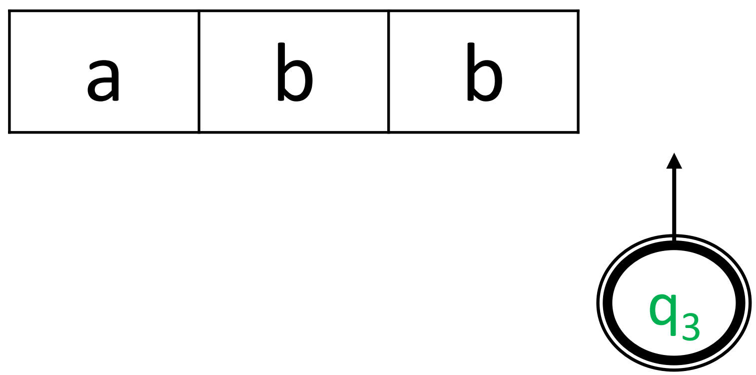


# Un DFA all’opera




*ACCEPT*

Input string: abb

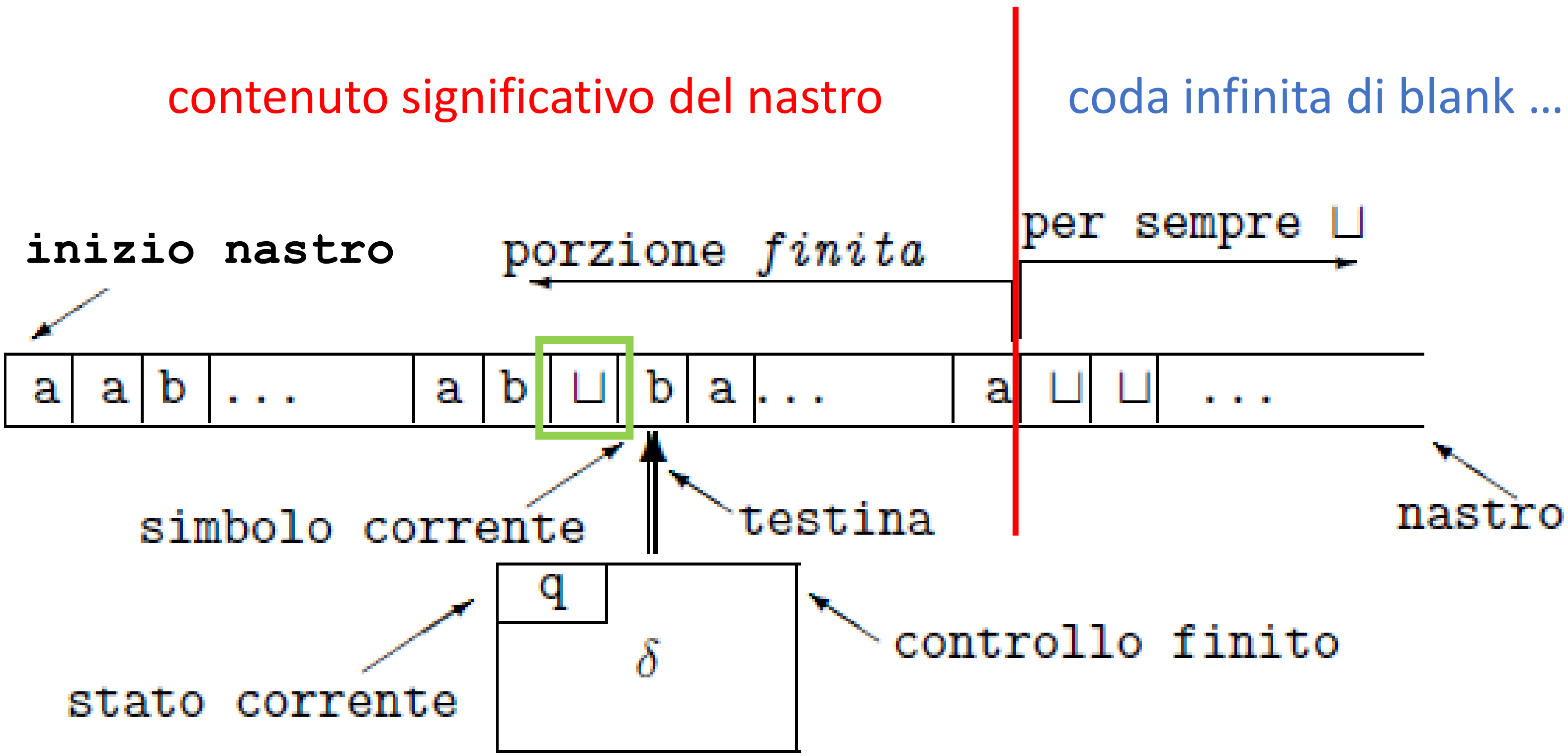


# Una MdT

Una Turing Machine è

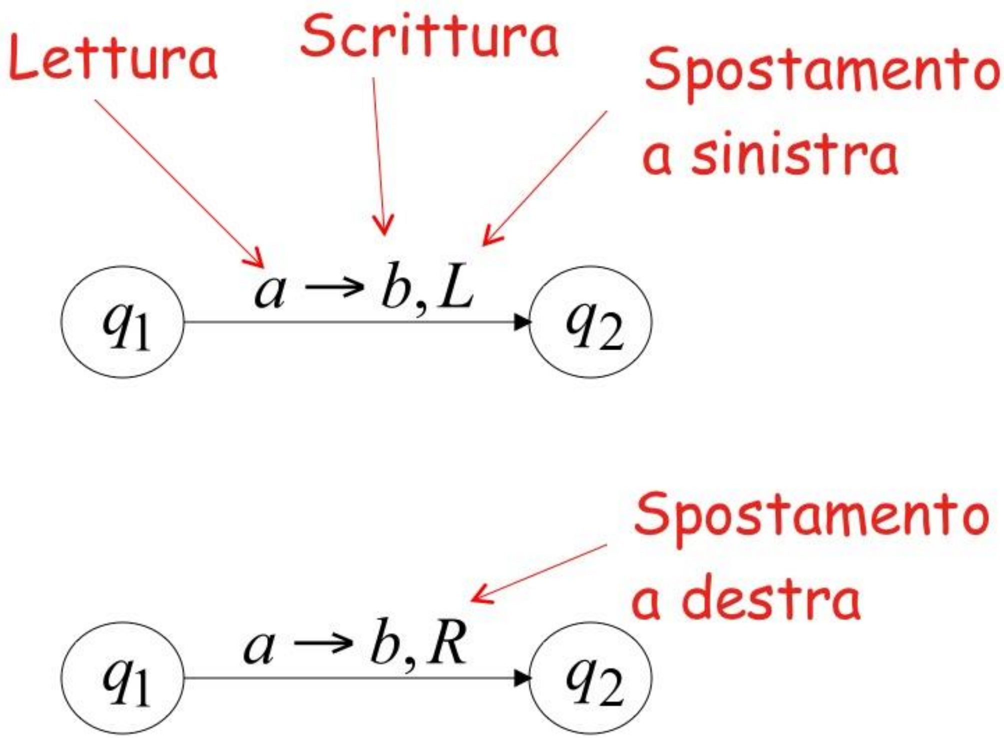
- una macchina a stati finiti con un nastro semi–infinito
- La testina può muoversi in entrambe le direzioni.
- Può leggere, scrivere in ogni cella del nastro
- Quando la MdT raggiunge uno stato accept/reject allora accetta/rifiuta immediatamente.

# Una MdT




Il contenuto significativo del nastro è una stringa $w \in \Gamma^*$, con la convenzione che il suo ultimo carattere (se $w \neq \varepsilon$ ) non sia blank.
La stringa $w$ PUO' contenere blank al suo interno.
A volte nel progetto di una MdT si definiscono delle transizioni per scrivere un carattere speciale nella prima cella del nastro, per meglio individuarla.

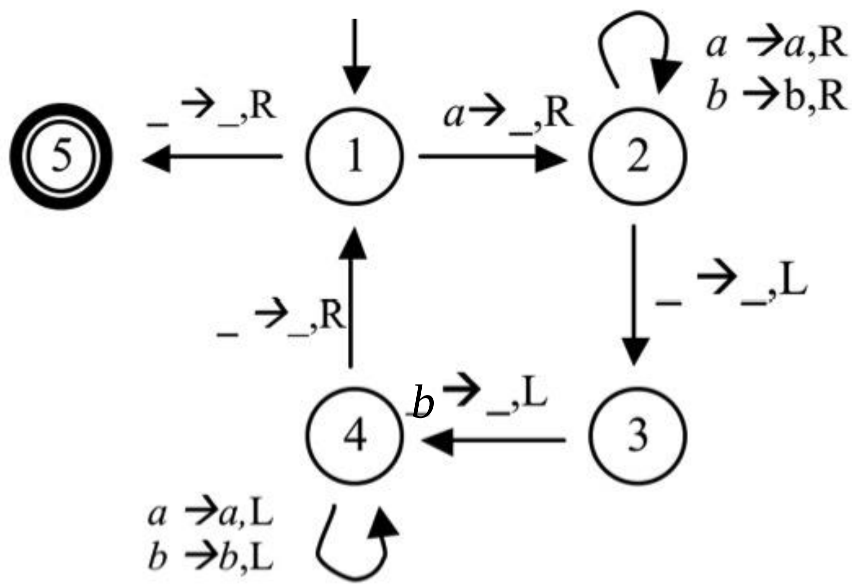
# Stati e Transizioni



# Diagramma di stato

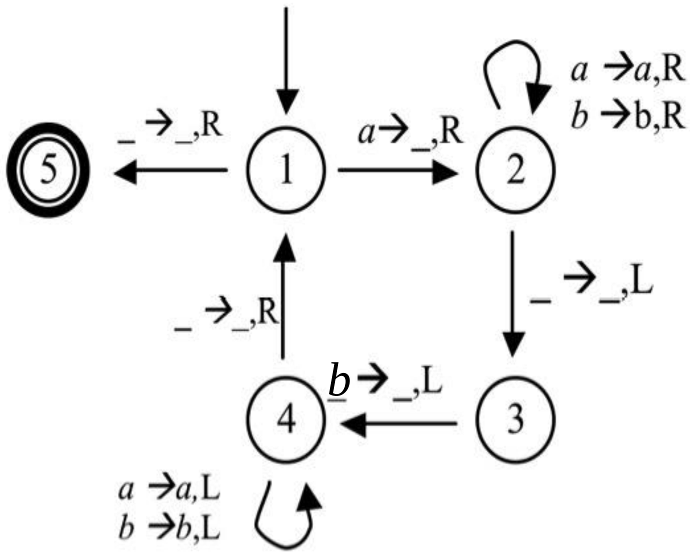
Cancella ripetutamente: prima occorrenza di (a) e ultima di (b)
se la stringa era del tipo $a^n b^n$, non rimangono simboli
Cinque passi

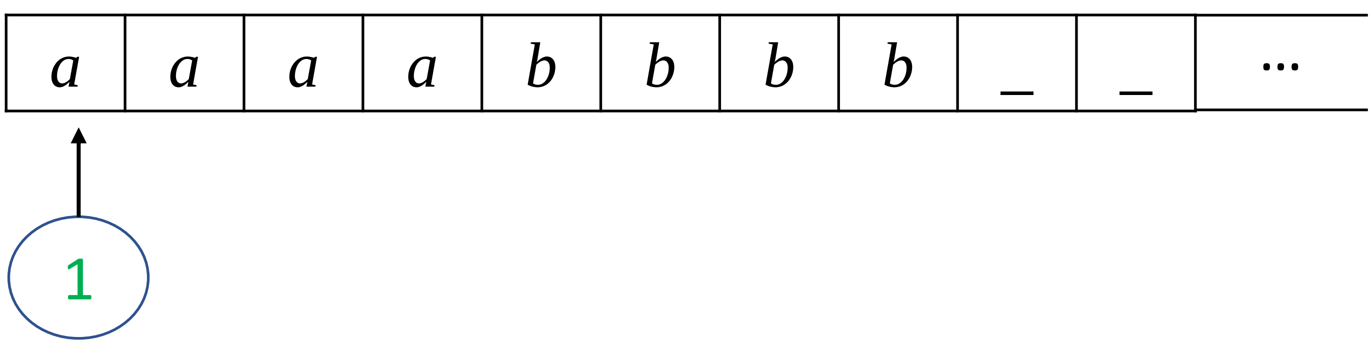
1. Se leggi _, vai a 5. Se leggi a, scrivi _ e vai a 2.
2. Spostati a destra (R) di tutti a e b. Al primo _, muovi a sinistra (L) e vai a 3
3. se leggi b, scrivi _ e vai a 4.
4. Spostati a sinistra (L) di tutti a e b. Leggendo _, muovi R e vai a 1.
5. Accept.



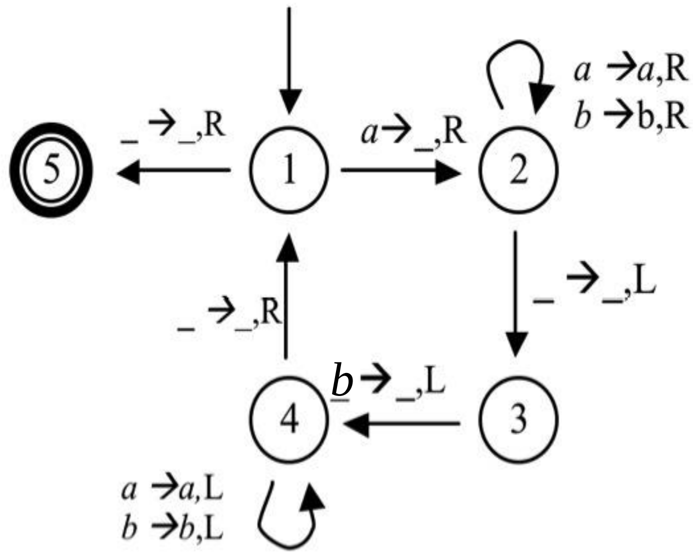

Le transizioni non indicate portano in uno stato di Reject (per esempio se in 1. leggi b).

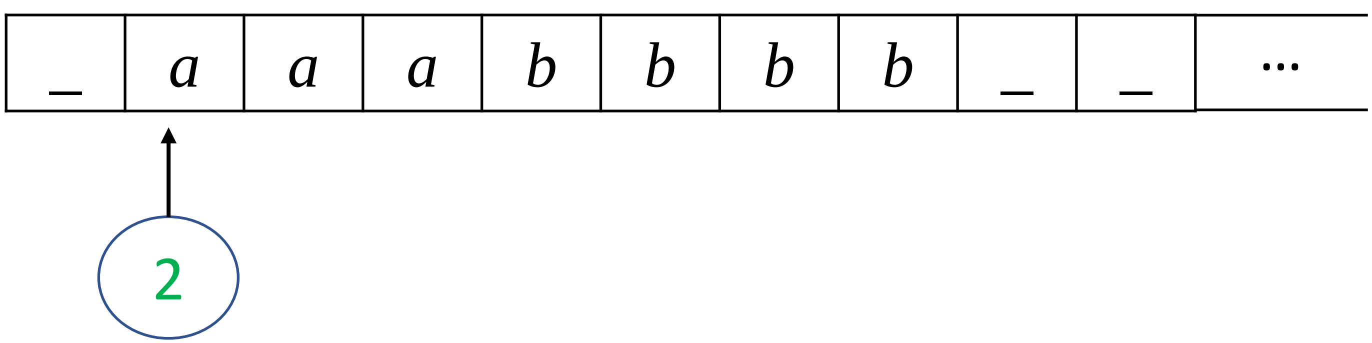
# MdT all’opera su *aaaabbbb*



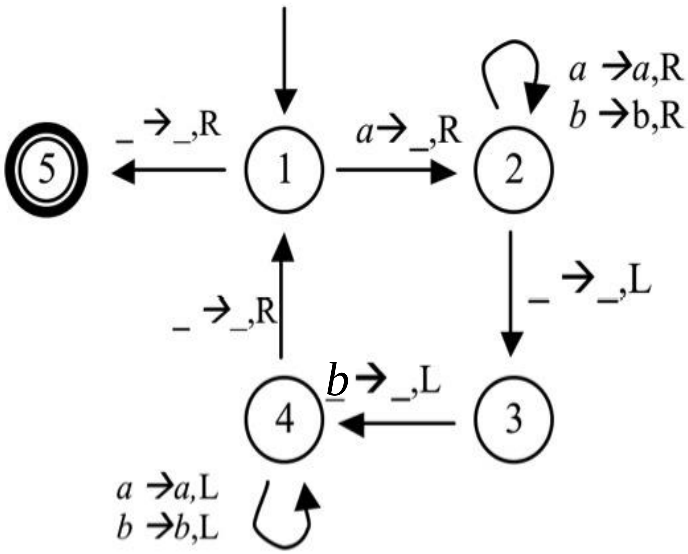



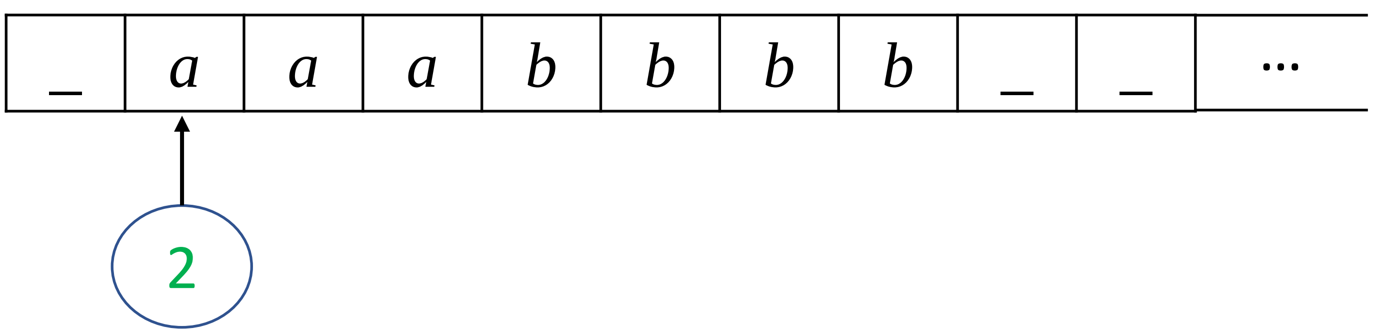
# MdT all’opera su *aaaabbbb*



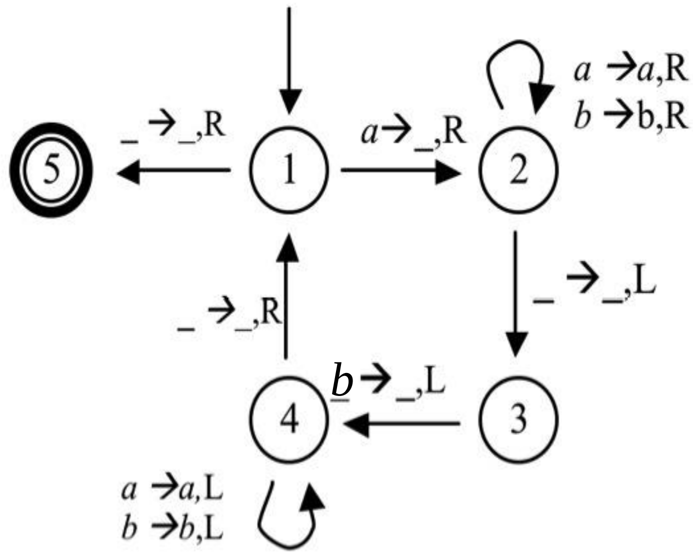



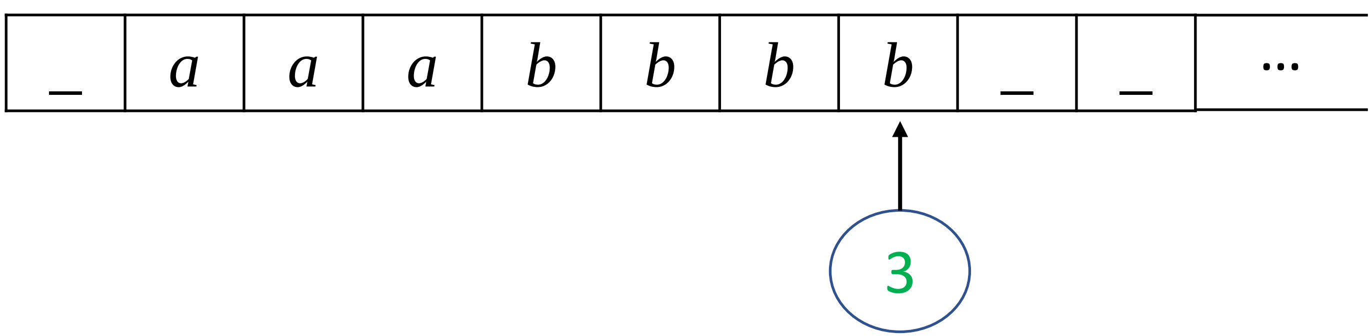
# MdT all’opera su *aaaabbbb*



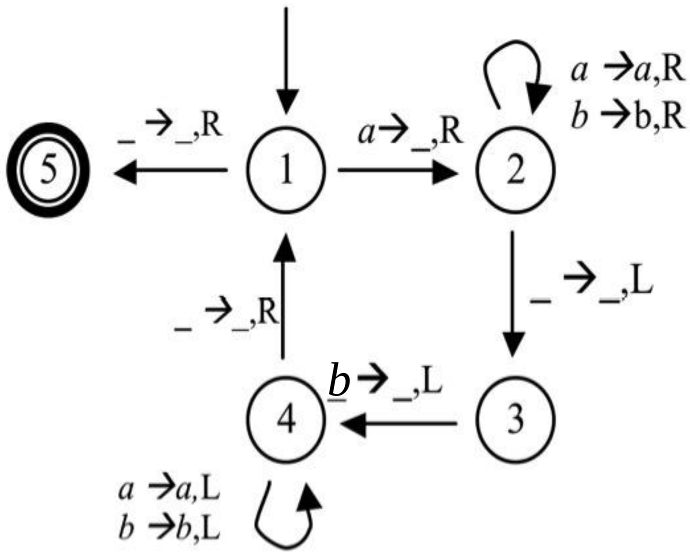



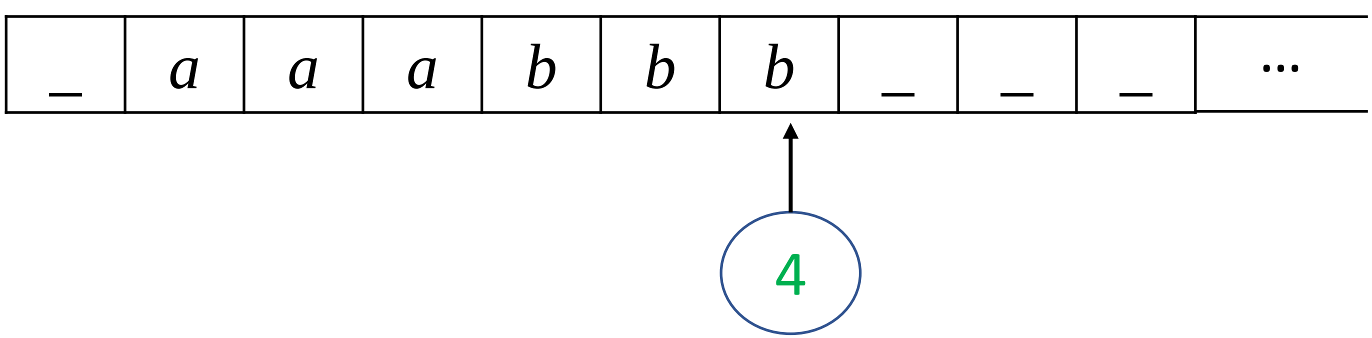
# MdT all’opera su *aaaabbbb*



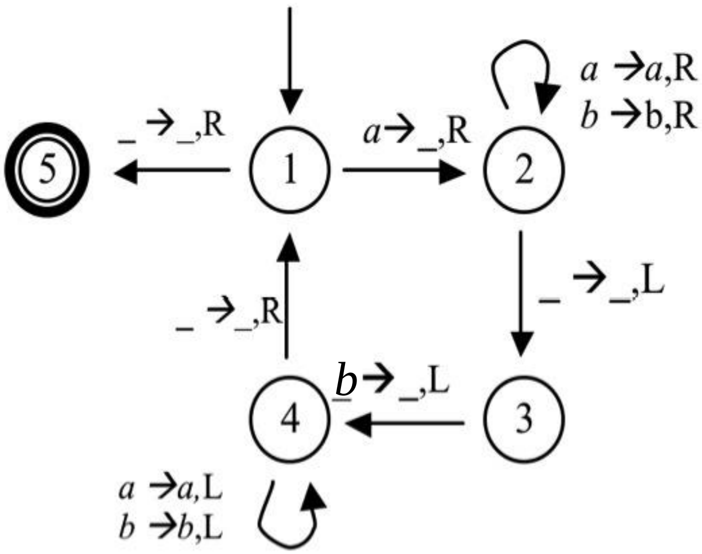
# MdT all’opera su *aaaabbbb*

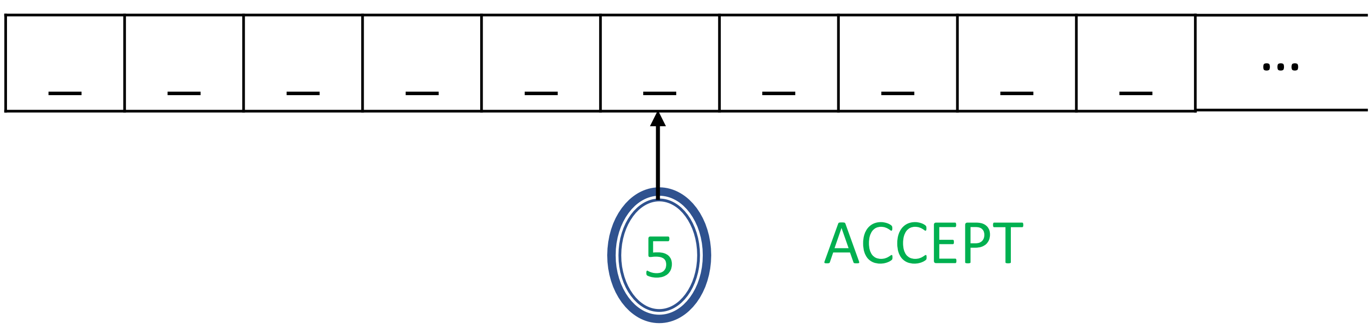


# Puntini, puntini …

# MdT all’opera su *aaaabbbb*




ACCEPT

# Descrizione formale MdT

Una Macchina di Turing è una settupla

$$(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$$

- **Insieme Stati** $Q$
- **Alfabeto di lavoro** $\Sigma$ ($\_ \notin \Sigma$)
- $\Gamma$: **Alfabeto del nastro** ($\_ \in \Gamma,\ \Sigma \subset \Gamma$)
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$: funzione transizione
- $q_0$: stato **iniziale**
- $q_{accept}$: stato **accept**
- $q_{reject}$: stato **reject**

Una Macchina di Turing è una settupla

$$(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$$

- **Insieme Stati** $Q$
- **Alfabeto di lavoro** $\Sigma$ ($\_ \notin \Sigma$)
- $\Gamma$: **Alfabeto del nastro** ($\_ \in \Gamma$, $\Sigma \subset \Gamma$)
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$: funzione transizione
- $q_0$: stato **iniziale**
- $q_{accept}$: stato **accept**
- $q_{reject}$: stato **reject**

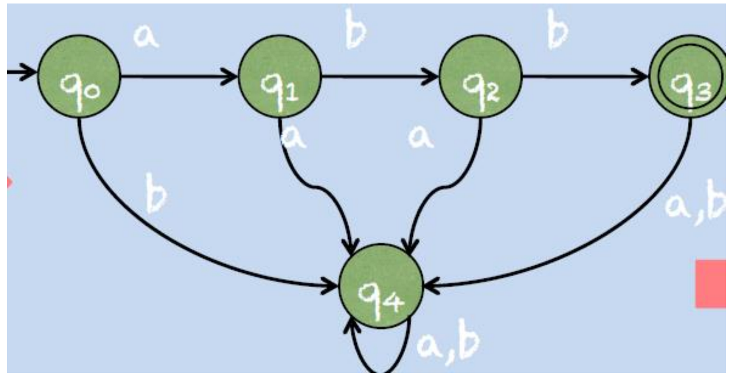
Un Automa Finito (DFA) è una quintupla

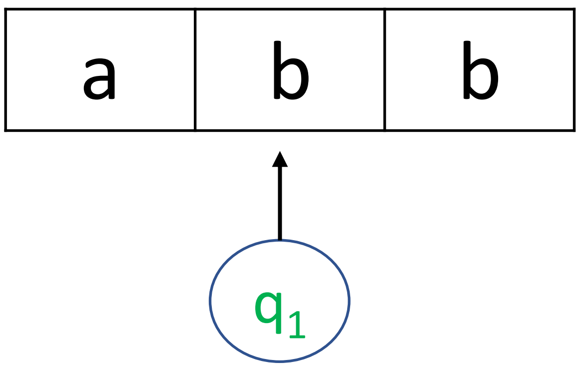
$$(Q, \Sigma, \delta, q_0, F)$$

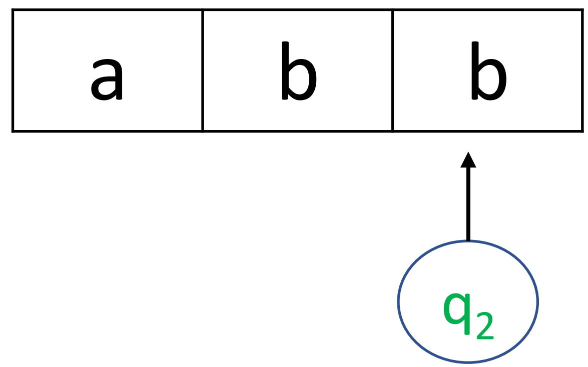
- **Insieme Stati** Q
- **Alfabeto** $\Sigma$
- $\delta : Q \times \Sigma \to Q$ : funzione di transizione
- $q_0 \in Q$ : stato **iniziale**
- $F \subseteq Q$ : stati **finali**

Caccia alle differenze!

# Un passo della computazione del DFA



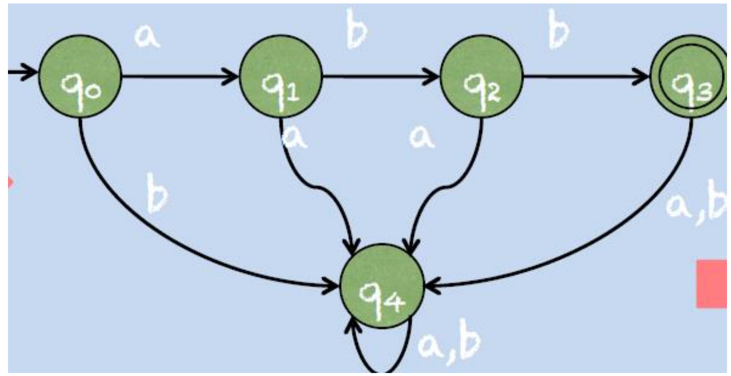



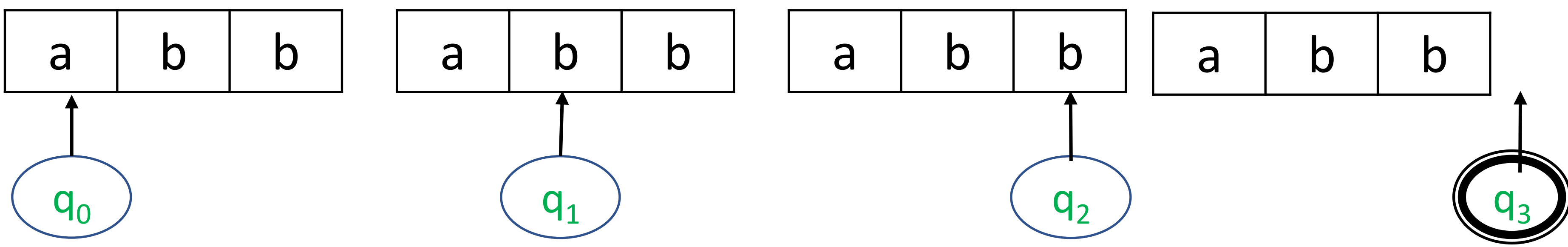




$$\delta\,(q_1, b) = q_2$$

# Ua computazione del DFA

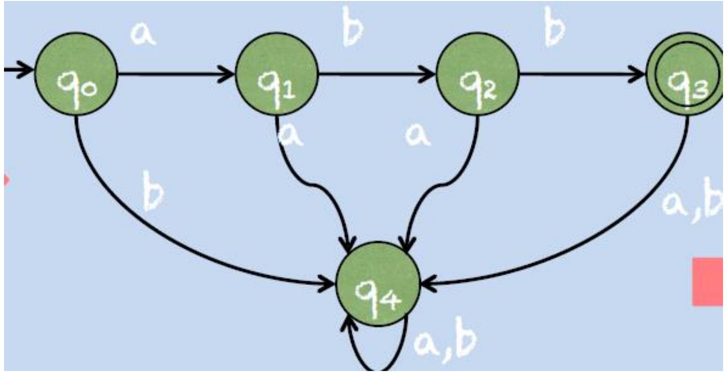
Su input abb




E’ sufficiente elencare gli stati: $q_0$ , $q_1$ , $q_2$ , $q_3$ .
E possiamo ricostruire tutta la computazione: il resto lo sappiamo.

# Ua computazione del DFA

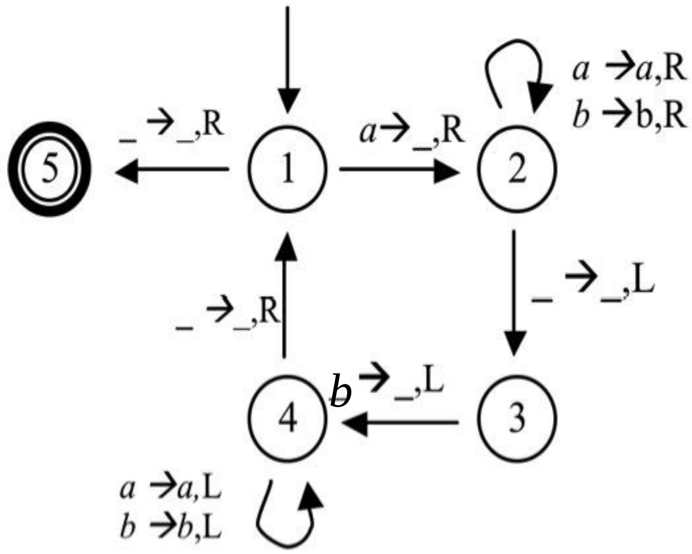



Dati un DFA, una stringa e una sequenza di stati possiamo capire se è una computazione valida del DFA sulla stringa.

Esempio: la sequenza $q_0$ , $q_4$ , $q_2$ , $q_3$ è una computazione valida del DFA su abb?

$\delta(q_0, a) = q_4$ ? dove a è la prima lettera di input
$\delta(q_4, b) = q_2$? dove b è la seconda lettera di input
$\delta(q_2, b) = q_3$? dove b è la successiva lettera di input

# MdT all’opera su *aabb*

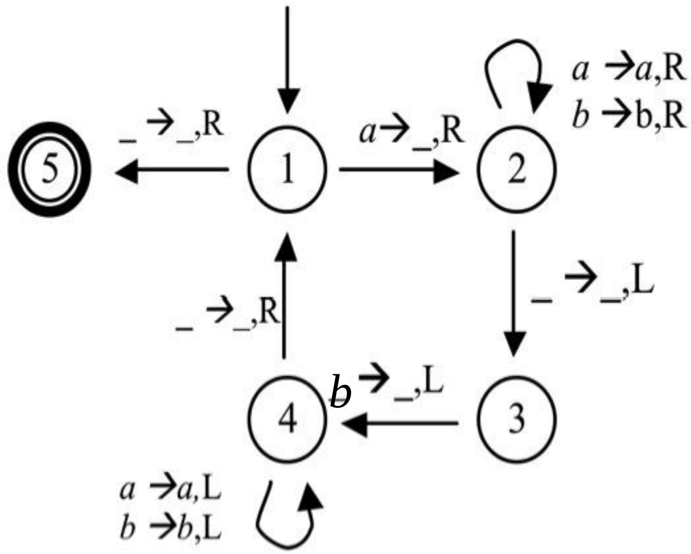



| | | | | |
|---|---|---|---|---|
| *a* | *a* | *b* | *b* | … |

Elencare gli stati: 1 , 2 , 2 , 2 , 2 , 3 , 4 , 4 , 4 , 1 , 2 , 2 , 3 , 4 , 1 , 5

è sufficiente a definire la computazione?

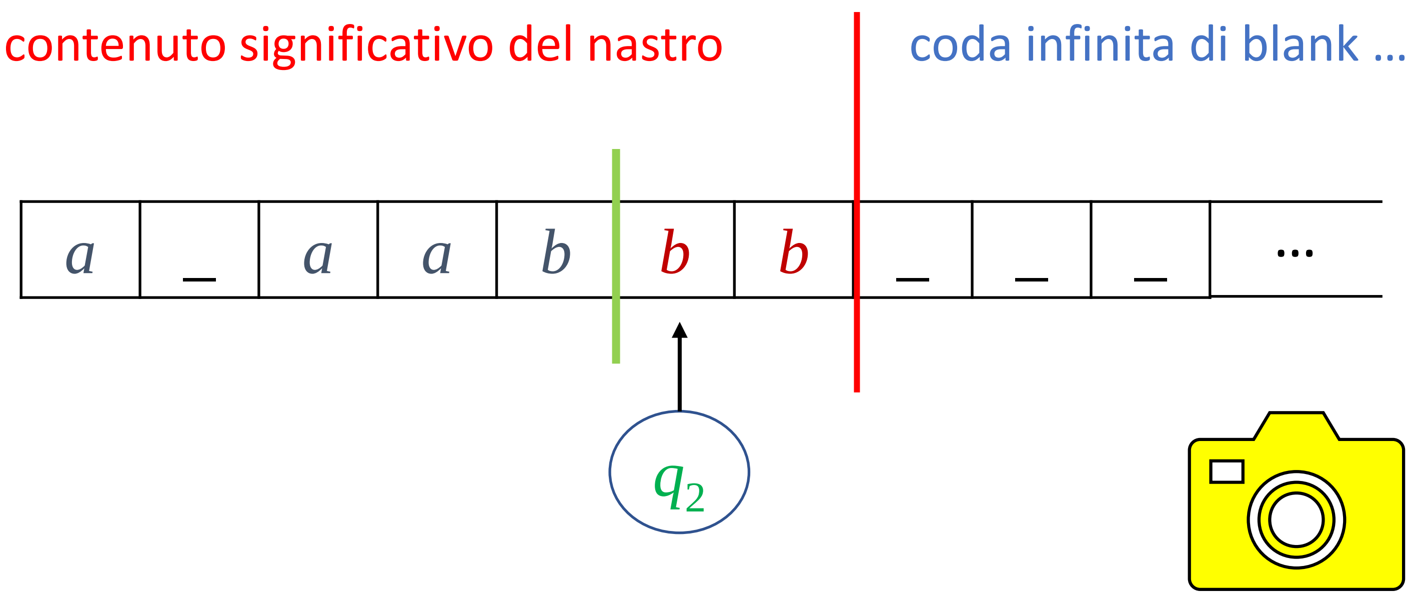
# MdT all'opera su *aabb*




Dati una MdT, una stringa e una sequenza di stati posso capire se è una computazione valida della MdT sulla stringa?
Esempio: la sequenza 1 , 2 , 2 , 2 , 3 , 4 , 1 , 5 è una computazione valida del MdT su *ab*?
Devo tenere traccia di altre informazioni per poter verificare i passi.

# Configurazione di una MdT

Occorre fare un'istantanea di stato, posizione e contenuto significativo del nastro correnti




$$a\_\ a\ a\ b\ q_2\ b\ b$$

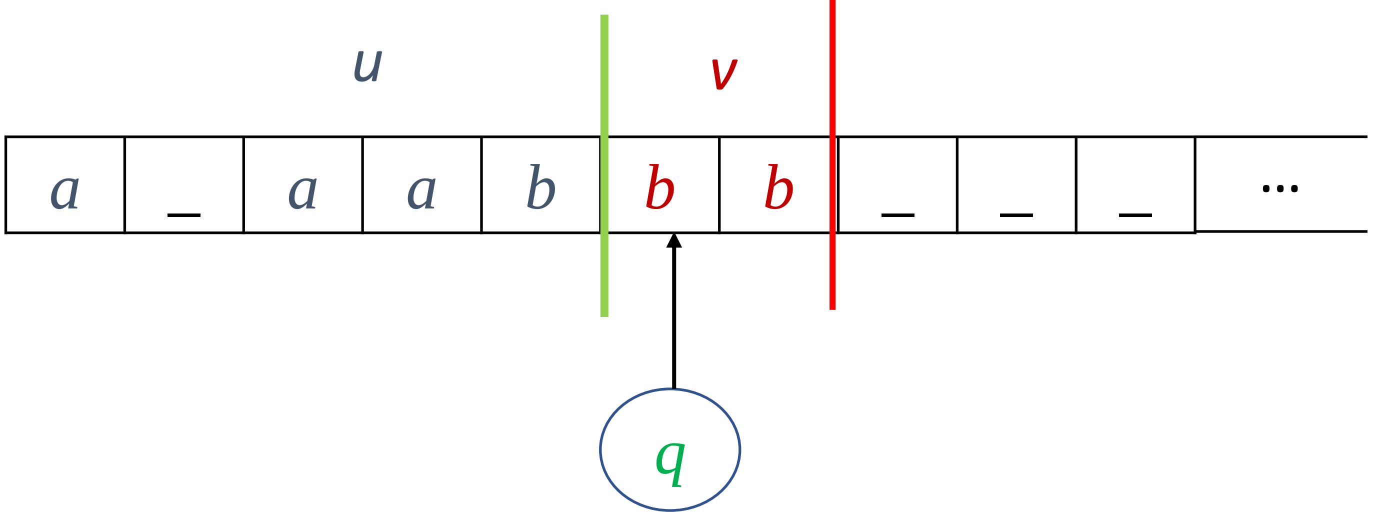
# Configurazione di una MdT

La configurazione $C = u\ q\ v$ corrisponde a

contenuto significativo del nastro =

$u\ v$

coda infinita di blank …



# Configurazione di una MdT

Descrizione concisa della situazione del calcolo di una MdT ad un certo istante, anche detta descrizione istantanea.

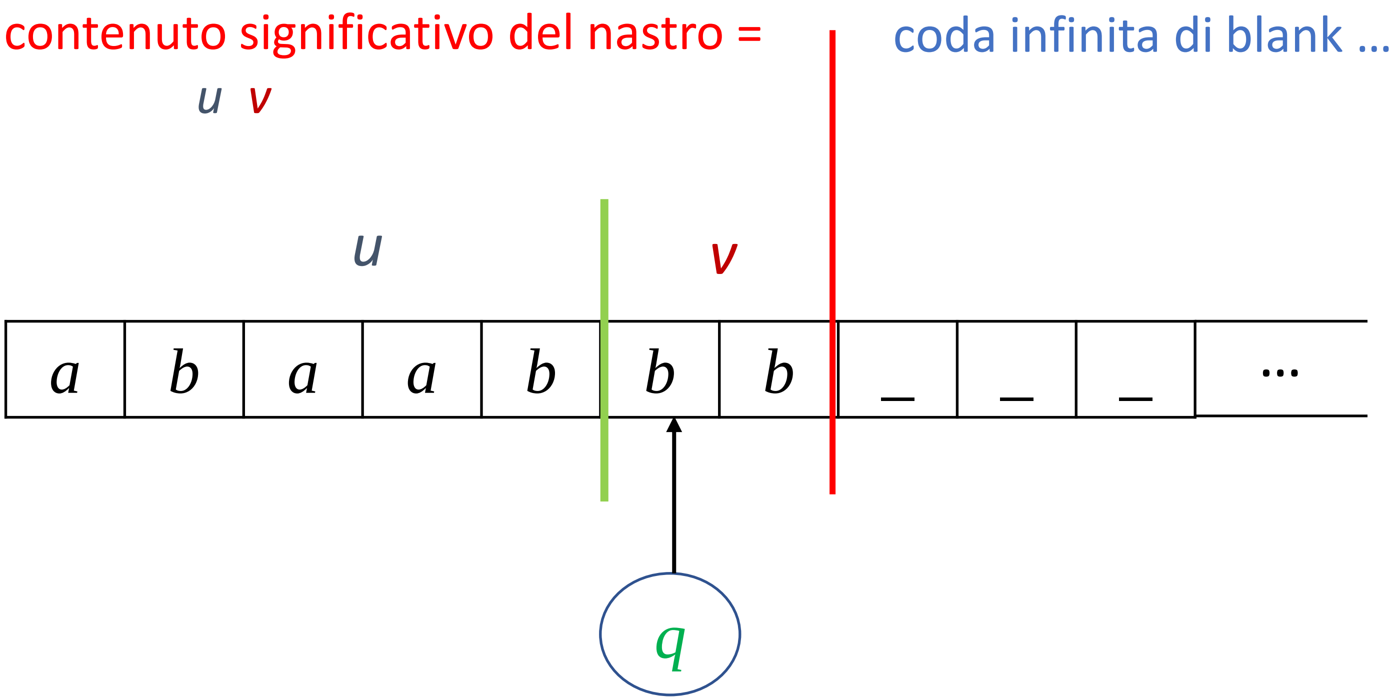
Configurazione di una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

$$C = u\ q\ v$$

- $q \in$ Q è lo stato corrente
- $u\ v \in \Gamma^*$ è il contenuto significativo del nastro (senza _ finali, se $u\ v \neq \varepsilon$)
- La testina è posizionata sul primo simbolo di $v$, se $v \neq \varepsilon$, su _ altrimenti

# Configurazione di una MdT: esempi

Qual è la configurazione corrispondente?



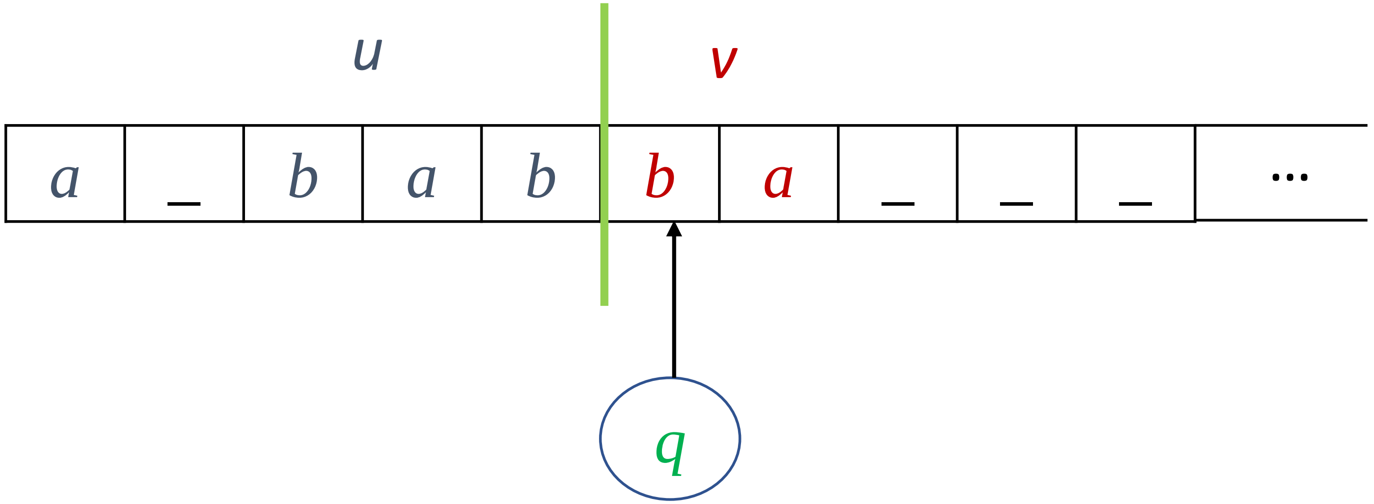

La configurazione corrispondente è: $u\ q\ v = abaab\ q\ bb$

# Configurazione di una MdT: esempi

Quale situazione rappresenta la configurazione

$$u\ q\ v = a\_bab\ q\ ba \quad ?$$

Contenuto significativo del nastro è $u\ v = a\_bab\ ba$



# Configurazioni particolari

In una configurazione *C* = *u q v*, sia *u* che *v* possono essere $\varepsilon$

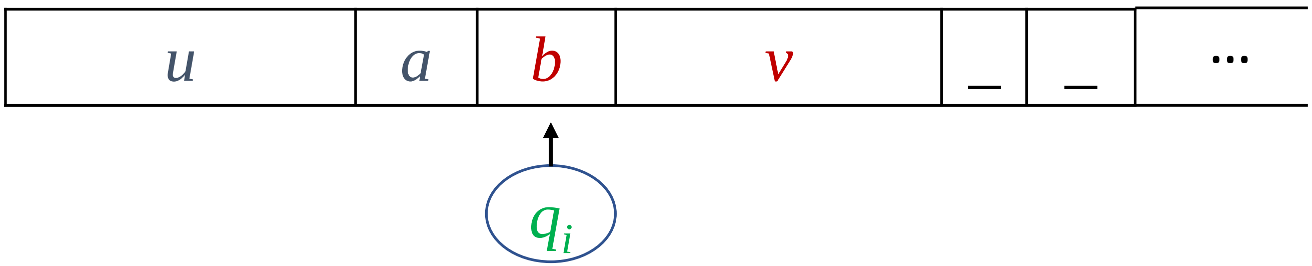
- Se *u* = $\varepsilon$, *C* = *q v*, allora la testina è posizionata sulla prima lettera di *v* nella prima cella del nastro (contenuto significativo nastro è $\varepsilon$ v = v)

- Se *v* = $\varepsilon$, *C* = *u q*, allora la testina è posizionata sulla prima cella della porzione di nastro contenente solo _ (ricorda che uv=u è la porzione significativa del nastro, senza la coda infinita di _ )

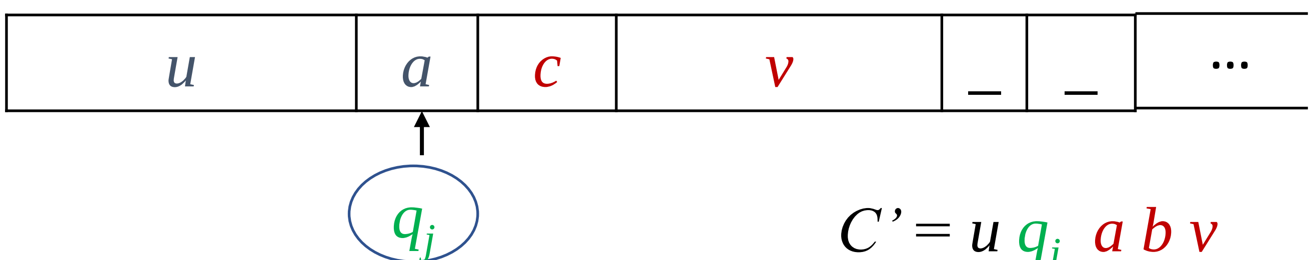
- *u q* è equivalente a *u q* _ ; la parte vuota del nastro è riempita con tutti _

# Computazione di una MdT: passo verso sinistra

Supponiamo che $C = u\ a\ q_i\ b\ v$




Se $\delta\,(q_i, b) = (\ q_j\ , c, L)$
quale sarà la successiva configurazione $C'$ ?

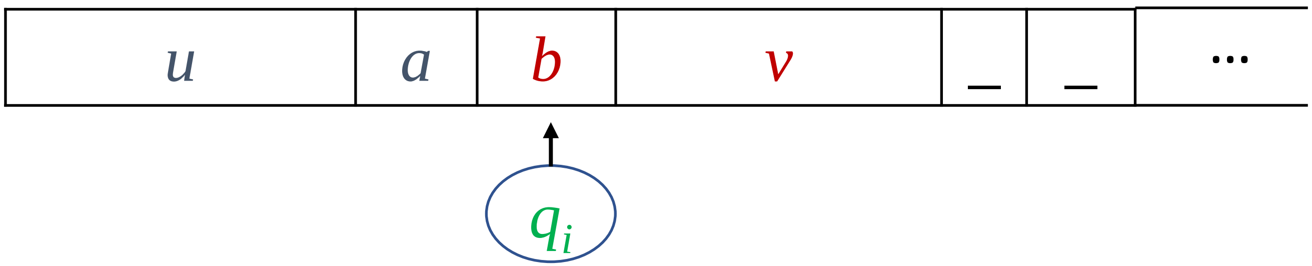



$C' = u\ q_j\ \ a\ b\ v$

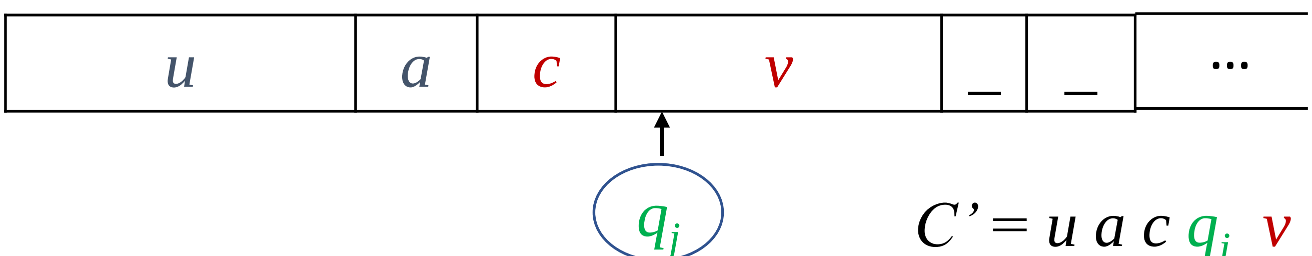
Diremo che $C$ produce $C'$, in simboli $C \rightarrow C'$

# Computazione di una MdT: passo verso destra

Supponiamo che $C = u\ a\ q_i\ b\ v$




Se $\delta\,(q_i, b) = (\ q_j\ , c, R)$
quale sarà la successiva configurazione $C'$ ?




$C' = u\ a\ c\ q_j\ \ v$

Diremo che $C$ produce $C'$, in simboli $C \rightarrow C'$

# Casi particolari

La definizione generale è più complessa perché bisogna considerare anche i casi particolari ($C = qv$, $C = uq$ con $u, v$ eventualmente uguali a $\epsilon$).

Ad esempio $q_i bv$ produce $q_j cv$
se $\delta(q_i, b) = (q_j, c, L)$.

$q_i bv$ produce $cq_j v$ se $\delta(q_i, b) = (q_j, c, R)$.

# Passo di computazione

Siano $C_1, C_2$ due configurazioni di una MdT $M$.

Se $C_1$ produce $C_2$, scriveremo

$$C_1 \to C_2$$

La trasformazione $\to$ di $C_1$ in $C_2$ prende il nome di **passo di computazione**.

Corrisponde a un'applicazione della funzione di transizione di $M$.

# Esempio

$$
\begin{aligned}
\delta(q_0, 0) &= (q_0, 0, R), \quad \delta(q_0, 1) = (q_0, 1, R), \\
\delta(q_0, \sqcup) &= (q_1, \sqcup, L), \\
\delta(q_1, 1) &= (q_2, 1, L), \quad \delta(q_2, 0) = (q_3, 0, L), \\
\delta(q_3, 1) &= (q_{accept}, 1, L)
\end{aligned}
$$

$q_0 11 \rightarrow 1 q_0 1 \rightarrow 11 q_0 \rightarrow 1 q_1 1 \rightarrow q_2 11 \rightarrow q_{reject} 11$

$q_0 101 \rightarrow 1 q_0 01 \rightarrow 10 q_0 1 \rightarrow 101 q_0 \rightarrow 10 q_1 1 \rightarrow 1 q_2 01 \rightarrow$
$q_3 101 \rightarrow q_{accept} 101$

# Computazione di una MdT

*Siano* $C, C'$ *configurazioni.*
$C \rightarrow^* C'$ *se esistono configurazioni* $C_1, \ldots, C_k$, $k \geq 1$ *tali che*

1. $C_1 = C$,
2. $C_i \rightarrow C_{i+1}$, *per* $i \in \{1, \ldots, k-1\}$,
   *(ogni* $C_i$ *produce* $C_{i+1}$*)*
3. $C_k = C'$.

*Diremo che* $C \rightarrow^* C'$ *è una* **computazione** *(di lunghezza* $k-1$*).*

Quando k = 1?

# Configurazioni

Una configurazione C si dice:

- iniziale su input w se $C = q_0 w$, con w $\in \Sigma^*$
- di accettazione se $C = u\ q_{accept}\ v$
- di rifiuto se $C = u\ q_{reject}\ v$

Poiché non esistono transizioni da $q_{accept}$ e da $q_{reject}$, allora le configurazioni di accettazione e di rifiuto sono dette configurazioni di arresto.

# Risultati di una computazione

Tre possibili Risultati computazione:

1. $M$ accetta – se si ferma in $q_{accept}$
2. $M$ rifiuta – se si ferma in $q_{reject}$
3. $M$ cicla/loop – se non si ferma mai

Mentre $M$ funziona non si può dire se è in loop; si potrebbe fermare in seguito oppure no.

# Parola accettata

**Definizione**

*Una MdT $M$ accetta una parola $w \in \Sigma^*$ se esiste una computazione $C \rightarrow^* C'$, dove $C = q_0 w$ è la configurazione iniziale di $M$ con input $w$ e $C' = u q_{accept} v$ è una configurazione di accettazione.*

Quindi $M$ accetta $w \in \Sigma^*$ se e solo se esistono configurazioni $C_1, C_2, \ldots, C_k$ di $M$, tali che

1. $C_1 = q_0 w$ è la configurazione iniziale di $M$ con input $w$
2. $C_i \rightarrow C_{i+1}$ per ogni $i \in \{1, \ldots, k-1\}$
3. $C_k$ è una configurazione di accettazione.

# Esempio 1

$$\begin{aligned}
\delta(q_0, 0) &= (q_0, 0, R), \quad \delta(q_0, 1) = (q_0, 1, R), \\
\delta(q_0, \sqcup) &= (q_1, \sqcup, L), \\
\delta(q_1, 1) &= (q_2, 1, L), \quad \delta(q_2, 0) = (q_3, 0, L), \\
\delta(q_3, 1) &= (q_{accept}, 1, L)
\end{aligned}$$

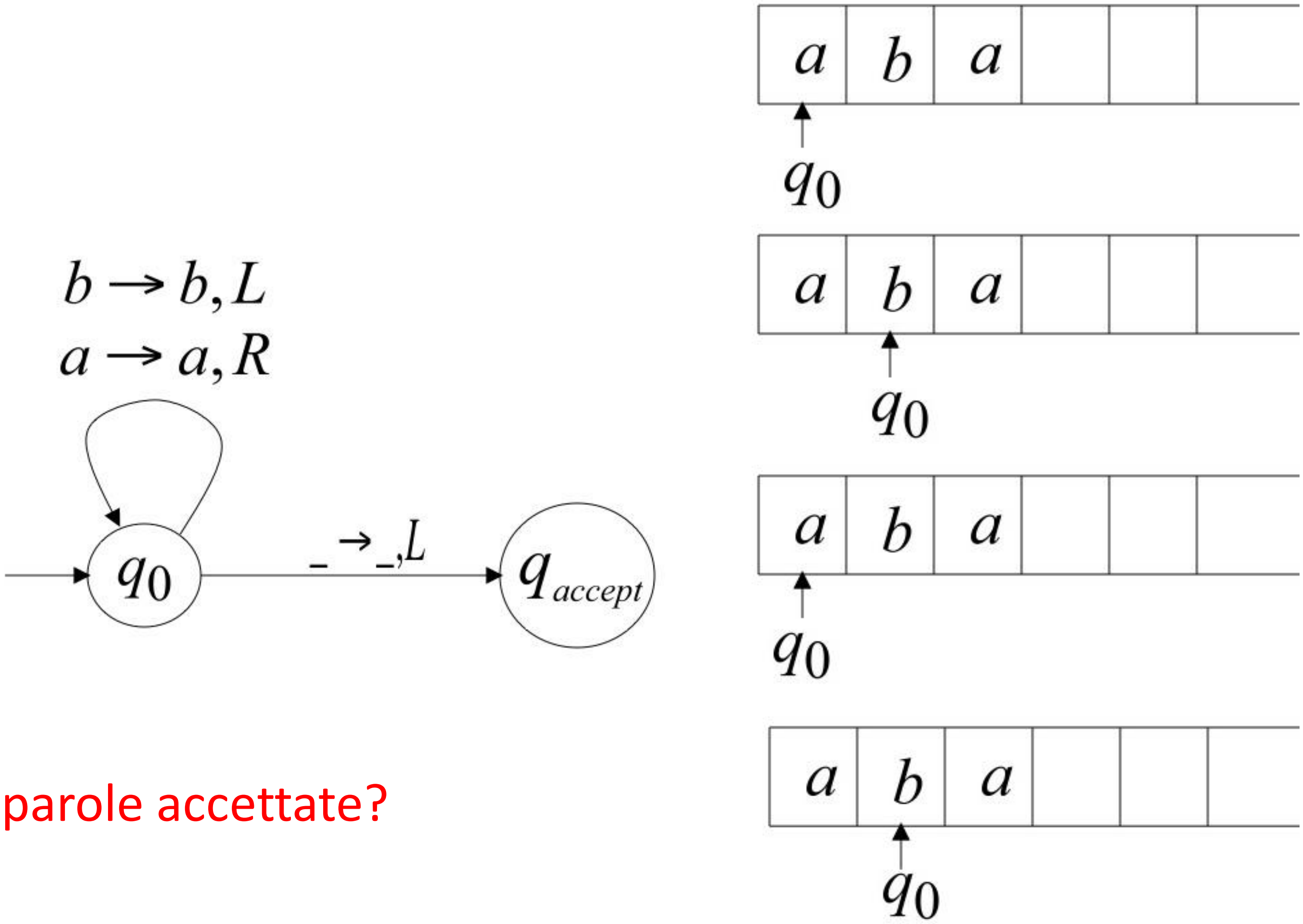
$$q_0 11 \rightarrow 1 q_0 1 \rightarrow 11 q_0 \rightarrow 1 q_1 1 \rightarrow q_2 11 \rightarrow q_{reject} 11$$

$q_0 11 \rightarrow^* q_{reject} 11$: 11 è rifiutata.

$$q_0 101 \rightarrow 1 q_0 01 \rightarrow 10 q_0 1 \rightarrow 101 q_0 \rightarrow 10 q_1 1 \rightarrow 1 q_2 01 \rightarrow q_3 101 \rightarrow q_{accept} 101$$

$q_0 101 \rightarrow^* q_{accept} 101$: 101 è accettata.

# Esempio di non terminazione




Quali parole accettate?

# Esempio di non terminazione

Esempio: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, con
$Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \sqcup\}$,
$\delta(q_0, a) = (q_0, a, R)$, $\delta(q_0, b) = (q_0, b, L)$,
$\delta(q_0, \sqcup) = (q_{accept}, \sqcup, L)$.

$$q_0 aa \to a q_0 a \to aa q_0 \to a q_{accept} a$$

$\boxed{q_0 aa \to^* a q_{accept} a\text{: } aa \text{ è accettata.}}$

# Esempio di non terminazione

Esempio: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, con
$Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \sqcup\}$,
$\delta(q_0, a) = (q_0, a, R)$, $\delta(q_0, b) = (q_0, b, L)$,
$\delta(q_0, \sqcup) = (q_{accept}, \sqcup, L)$.

$$q_0 aba \to aq_0 ba \to q_0 aba \to aq_0 ba \to \ldots$$

$$q_0 aba \to^* aq_0 ba$$ cicla e non si ferma mai

$\boxed{aba \text{ non è accettata.}}$

È errato dire che *aba* è rifiutata.

# Linguaggio riconosciuto da una MdT

**Definizione**

*Sia $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ una MdT. Il linguaggio $L(M)$ riconosciuto da $M$, è l'insieme delle stringhe che $M$ accetta:*

$$L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}.$$

Quindi

$$L(M) = \{w \in \Sigma^* \mid M \text{ accetta } w\}.$$

# Problema della fermata

A differenza del caso degli automi finiti, per i quali una stringa input è accettata o rifiutata, nel caso delle macchine di Turing c'è una terza possibilità: che per un dato input la computazione non termini.

Data una configurazione di una MdT che non è di arresto, non possiamo sapere se, a partire da tale configurazione, la computazione terminerà in qualche istante futuro o meno.

# Problema della fermata

A differenza del caso degli automi finiti, per i quali una stringa input è accettata o rifiutata, nel caso delle macchine di Turing c'è una terza possibilità: che per un dato input la computazione non termini.

Data una configurazione di una MdT che non è di arresto, non possiamo sapere se, a partire da tale configurazione, la computazione terminerà in qualche istante futuro o meno.

Sarebbe utile avere un algoritmo che, dato in input una MdT $M$ e una stringa $w$, determini in un tempo finito se la computazione eseguita da $M$ su $w$ termini o meno (**Problema della fermata**).

Vedremo che un tale algoritmo non esiste.

# Domanda

Sia $M$ una MdT. Sia $w = a_1 \dots a_n$, con $a_j \in \Sigma$. Supponiamo che $w$ è accettata da $M$.

Quindi esistono $u, v \in \Gamma^*$ tali che

$$q_0 w \to^* u q_{accept} v.$$

Possiamo concludere che in questa computazione $M$ **deve** aver letto tutti i caratteri $a_j$ di $w$?
E quindi che la computazione deve avere lunghezza almeno $n = |w|$?

Esercizio: Si consideri il linguaggio $L = \{aw \mid w \in \{a, b\}^*\}$.

- Definire una macchina di Turing che accetta tutte e sole le stringhe di $L$ ma che non si arresta su ogni input.
- Definire una macchina di Turing che accetta tutte e sole le stringhe di $L$, che si arresta su ogni input, con tre stati e senza cicli nel diagramma di stato. (svolto)

Troverete questi e altri esercizi sulla piattaforma/team