

DFA/NFA pregi:

Estremamente semplici.

Particolarmente adatti a descrivere alcuni compiti importanti come - pattern matching, lavastoviglie, telecomandi, semafori, circuiti sequenziali, ...

DFA/NFA difetti:

non sono sufficientemente "potenti" per risolvere numerosi problemi importanti.

Obiettivo: la macchina più semplice che sia "potente" quanto i computer convenzionali.

Macchina di Turing



Alan Mathison Turing (1912 – 1954)

Nel 1936 Alan Turing schematizzò i limiti delle macchine calcolatrici, ponendo le definizioni di quella che sarebbe diventata famosa come **macchina di Turing**.

Simile ad un automa finito ma con una memoria illimitata e senza restrizioni. Una macchina di Turing è un modello molto più preciso di computer: può fare tutto ciò che un reale computer può fare!

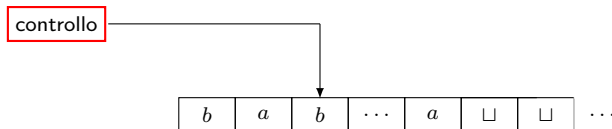
Ha permesso una formalizzazione del concetto di algoritmo e computazione.

Macchina di Turing

Una macchina di Turing è una macchina a stati finiti con un nastro infinito (memoria illimitata).

Il nastro è diviso in celle (caselle) ognuna delle quali può contenere simboli.

Ha una testina che consente di leggere e scrivere simboli ed è in grado di muoversi liberamente sul nastro.

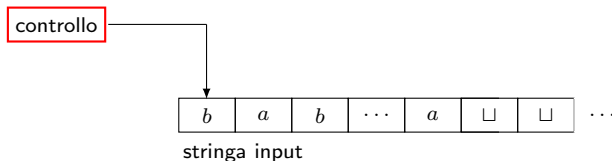


Esistono solo due stati che fanno terminare la computazione: *accept* e *reject*. Quando la Macchina di Turing raggiunge uno di essi, la computazione si ferma immediatamente.

Se non raggiunge uno stato di accettazione o rifiuto, la MdT andrà avanti per sempre, senza mai fermarsi.

Macchina di Turing

All'inizio il nastro contiene solo la stringa input. Tutto il resto è vuoto, cioè riempito con il simbolo *blank* \square .



Inizialmente la testina si trova sulla prima cella del nastro, cioè all'inizio della stringa input. Il simbolo \square segna la fine della stringa input.

Ad ogni passo, la testina legge il simbolo puntato, lo lascia inalterato o lo sovrascrive con un altro simbolo, e poi si sposta a destra o a sinistra.

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$.

- **Insieme Stati Q**

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$.

- ▶ **Insieme Stati** Q
- ▶ **Alfabeto di lavoro** Σ ($\sqcup \notin \Sigma$)

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$.

- ▶ **Insieme Stati** Q
- ▶ **Alfabeto di lavoro** Σ ($\sqcup \notin \Sigma$)
- ▶ Γ : **Alfabeto del nastro** ($\sqcup \in \Gamma$, $\Sigma \subset \Gamma$)

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$.

- ▶ **Insieme Stati** Q
- ▶ **Alfabeto di lavoro** Σ ($\sqcup \notin \Sigma$)
- ▶ Γ : **Alfabeto del nastro** ($\sqcup \in \Gamma$, $\Sigma \subset \Gamma$)
- ▶ $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$: funzione di transizione

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

- ▶ **Insieme Stati** Q
- ▶ **Alfabeto di lavoro** Σ ($\sqcup \notin \Sigma$)
- ▶ Γ : **Alfabeto del nastro** ($\sqcup \in \Gamma$, $\Sigma \subset \Gamma$)
- ▶ $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$: funzione di transizione
- ▶ q_0 : stato **iniziale**

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

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

Definizione formale di una MdT

Una Macchina di Turing è una settupla: $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

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

Funzione di transizione di una MdT

Sia M una Macchina di Turing definita da $(Q, \Sigma, \Gamma, \delta, q_o, q_{accept}, q_{reject})$.

- Ad ogni istante M occupa uno degli stati in Q .

Funzione di transizione di una MdT

Sia M una Macchina di Turing definita da $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

- ▶ Ad ogni istante M occupa uno degli stati in Q .
- ▶ La testina si trova in una cella del nastro contenente qualche simbolo $\gamma \in \Gamma$.

Funzione di transizione di una MdT

Sia M una Macchina di Turing definita da $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

- ▶ Ad ogni istante M occupa uno degli stati in Q .
- ▶ La testina si trova in una cella del nastro contenente qualche simbolo $\gamma \in \Gamma$.
- ▶ La funzione di transizione $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ dipende dallo stato q e dal simbolo di nastro γ .

Funzione di transizione di una MdT

Sia M una Macchina di Turing definita da $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.

- ▶ Ad ogni istante M occupa uno degli stati in Q .
- ▶ La testina si trova in una cella del nastro contenente qualche simbolo $\gamma \in \Gamma$.
- ▶ La funzione di transizione $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ dipende dallo stato q e dal simbolo di nastro γ .

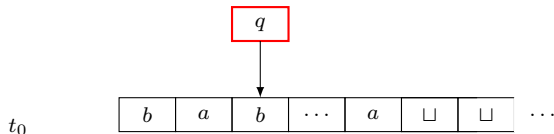
Se la macchina è in uno stato $q \in Q$ e la testina punta alla casella del nastro contenente il simbolo a , allora la macchina si comporterà come descritto dalla funzione di transizione δ sulla coppia stato-simbolo (q, a) .

Se $\delta(q, a) = (q', b, d)$, allora la macchina:

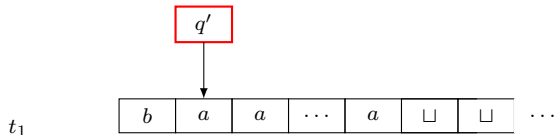
1. scrive b al posto di a ;
2. sposta la testina a sinistra o a destra a seconda che d sia rispettivamente L o R (**attenzione:** se $d = L$ e la testina si trova nella casella più a sinistra, allora la testina non si sposta);
3. passa nello stato q' ;

Funzione di transizione di una MdT

Supponiamo che la MdT al tempo t_0 si trovi nello stato q , con la testina posizionata su una casella del nastro contenente il simbolo b . Allora il comportamento della macchina al tempo t_0 è descritto da $\delta(q, b)$.

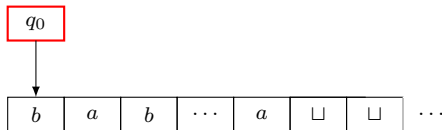


Se $\delta(q, b) = (q', a, L)$, allora la macchina scriverà a al posto di b , sposterà la testina di una casella a sinistra e transirà nello stato q' . Cioè la configurazione della MdT al tempo $t_1 = t_0 + 1$ sarà:



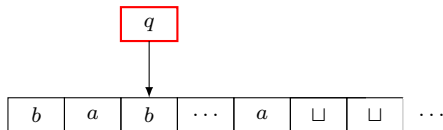
Computazione di una MdT

- ▶ La computazione parte sempre da uno stato iniziale q_0 .
- ▶ L'input è posizionato sulla parte più a sinistra del nastro: le prime n celle a sinistra, se n è lunghezza input.
- ▶ La testina si trova nella prima cella a sinistra del nastro (cella 0).

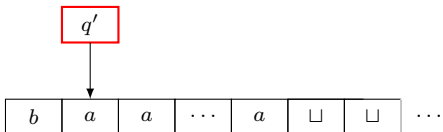


Funzione di transizione di una MdT

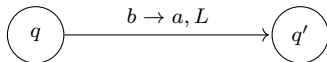
Tempo t_0 :



Tempo $t_1 = t_0 + 1$:



Rappresentazione della transizione (diagramma di stato):



Differenze tra MdT e automi finiti

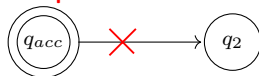
Differenze tra macchine di Turing e automi finiti.

- ▶ Una MdT ha un nastro di lunghezza infinita.
- ▶ Una MdT può sia leggere sia scrivere sul nastro.
- ▶ Ha una testina che può muoversi sia verso sinistra sia verso destra.
- ▶ Gli stati speciali q_{accept} e q_{reject} hanno **effetto immediato!**

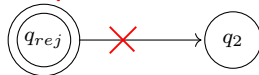
Diagramma di stato di una MdT

Gli stati di arresto non hanno archi uscenti. Quando la macchina raggiunge uno stato di arresto, la computazione termina.

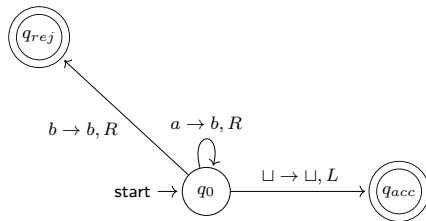
Non permesso:



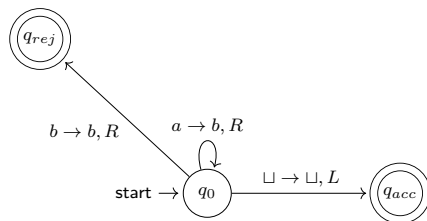
Non permesso:



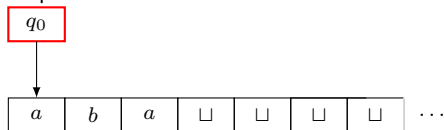
Esempio di MdT: diagramma di stato



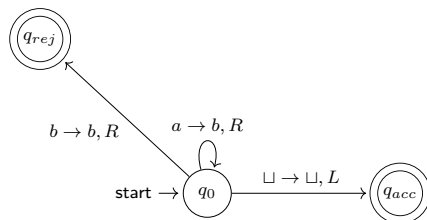
Esempio di MdT: diagramma di stato



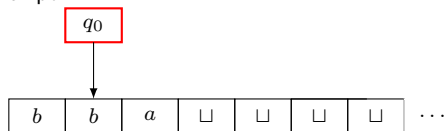
Tempo 0



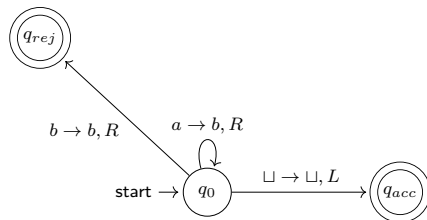
Esempio di MdT: diagramma di stato



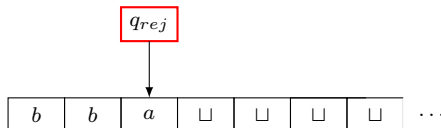
Tempo 1



Esempio di MdT: diagramma di stato

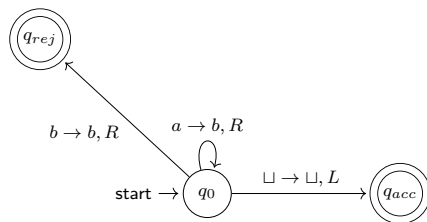


Tempo 2

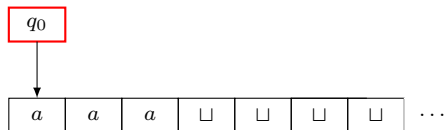


La stringa aba non è accettata.

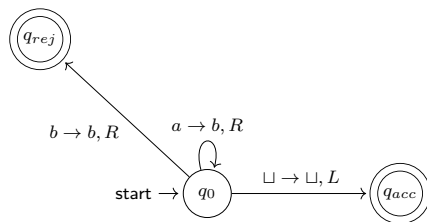
Esempio di MdT: diagramma di stato



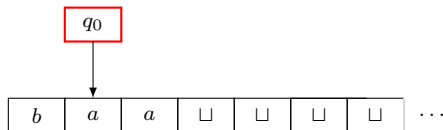
Tempo 0



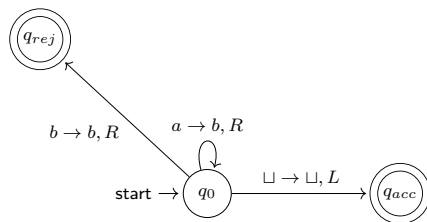
Esempio di MdT: diagramma di stato



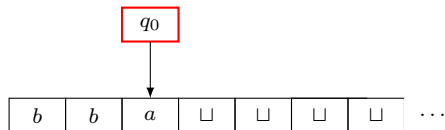
Tempo 1



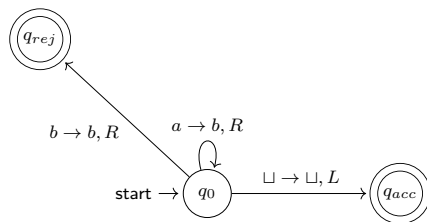
Esempio di MdT: diagramma di stato



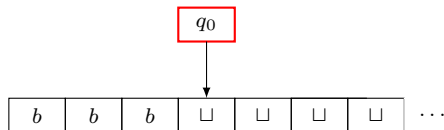
Tempo 2



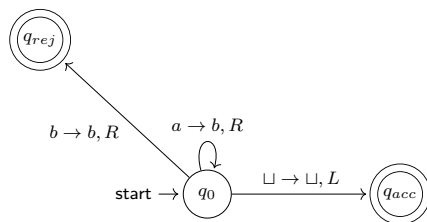
Esempio di MdT: diagramma di stato



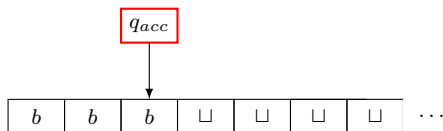
Tempo 3



Esempio di MdT: diagramma di stato

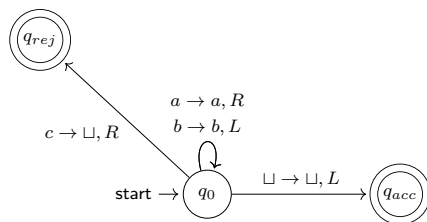


Tempo 4

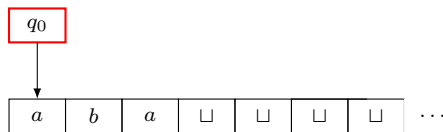
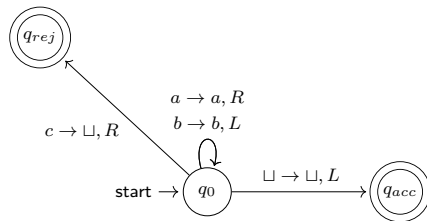


La stringa aba non è accettata.

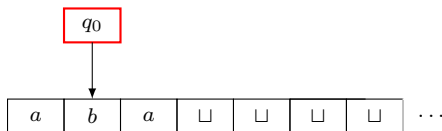
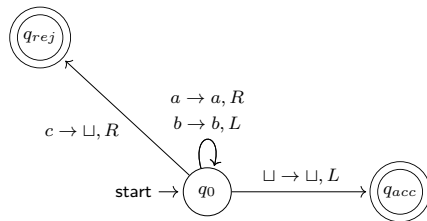
Esempio di non terminazione



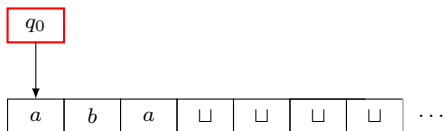
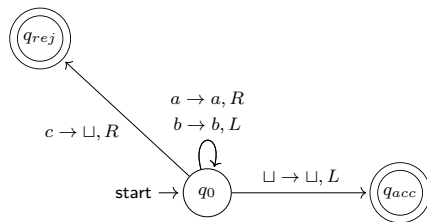
Esempio di non terminazione



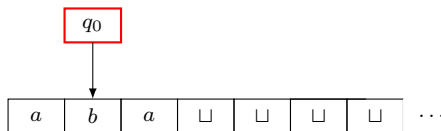
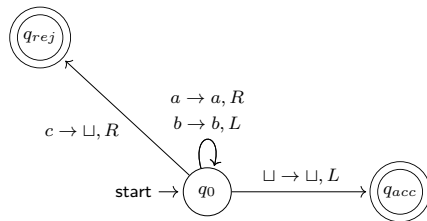
Esempio di non terminazione



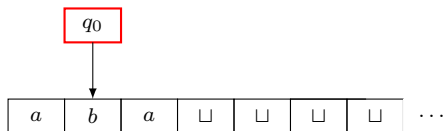
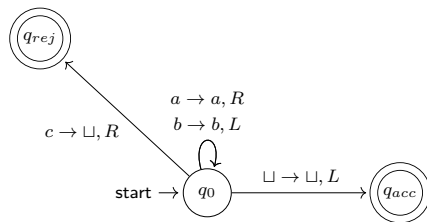
Esempio di non terminazione



Esempio di non terminazione



Esempio di non terminazione



Gli stati di terminazione non verranno mai raggiunti: la macchina cicla all'infinito.

Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

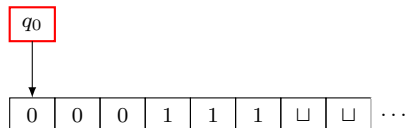
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

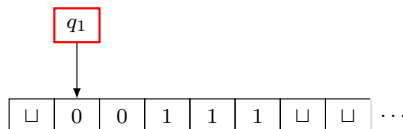
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

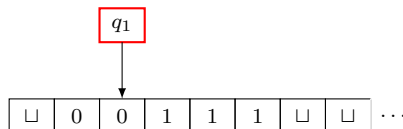
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

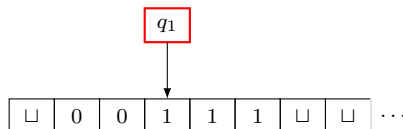
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

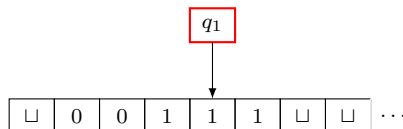
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

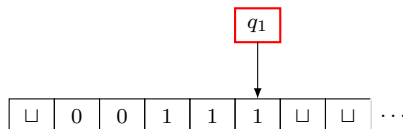
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

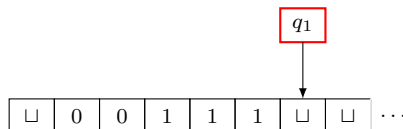
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

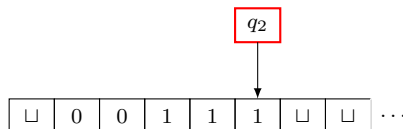
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

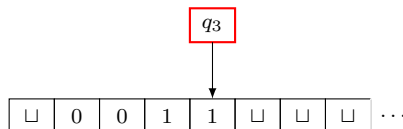
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

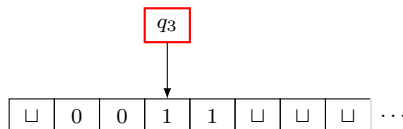
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

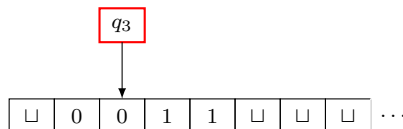
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

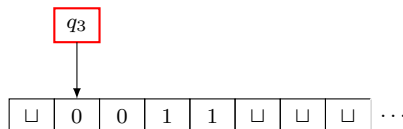
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

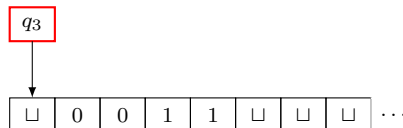
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

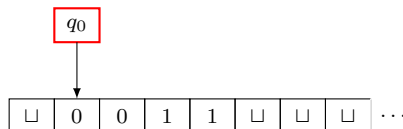
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

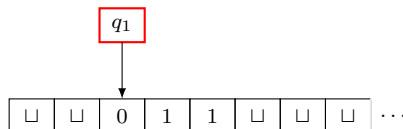
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

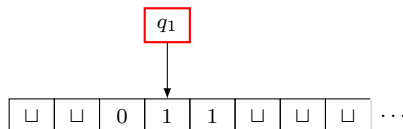
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

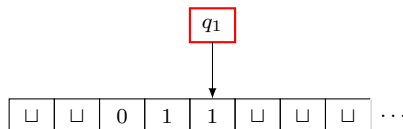
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

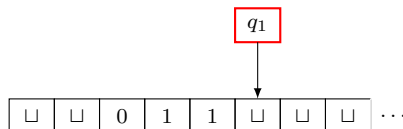
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

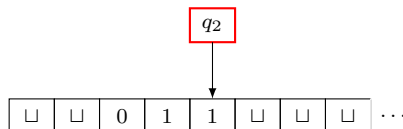
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

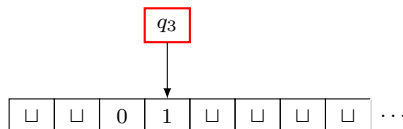
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

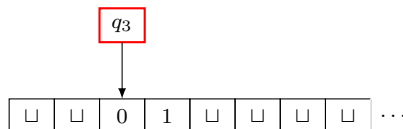
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

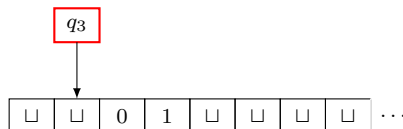
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

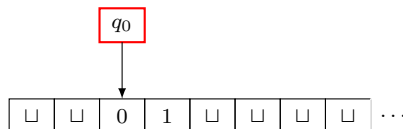
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

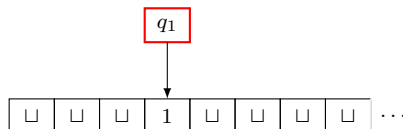
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

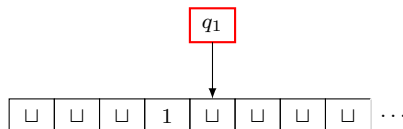
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

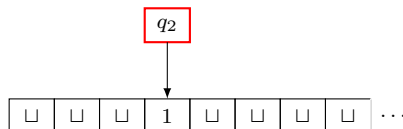
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

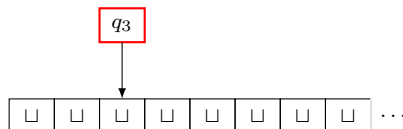
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

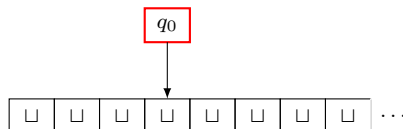
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



Esempio: linguaggi non regolari e MdT

Può essere dimostrato (con il Pumping lemma) che il seguente linguaggio non è regolare:

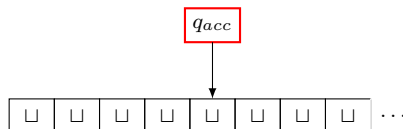
$$L = \{0^n 1^n \mid n \geq 0\}.$$

Non esiste alcun automa finito in grado di riconoscere L .

Possiamo progettare una macchina di Turing che accetta le stringhe $w \in L$ e rifiuta le stringhe $w \notin L$.

Idea

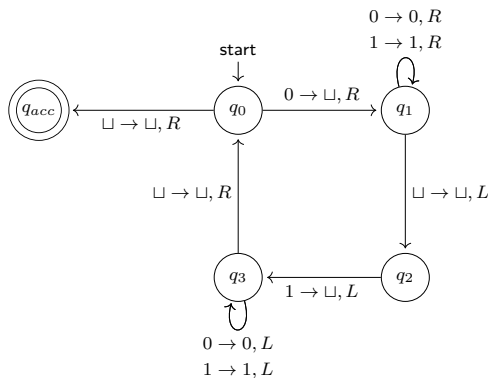
- Cancella ripetutamente la prima occorrenza di 0 e l'ultima di 1.
- La stringa è del tipo $0^n 1^n$ **se e solo se** ogni 0 all'estrema sinistra avrà un corrispondente 1 all'estrema destra.



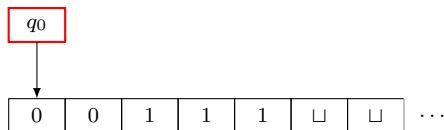
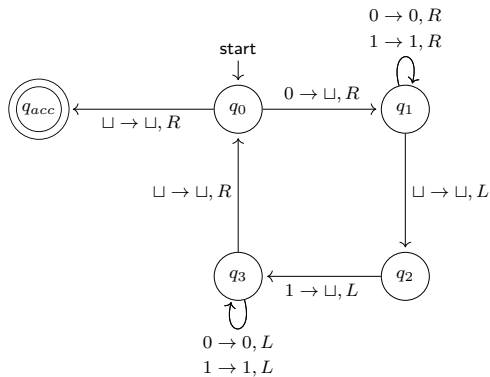
Formalizziamo: diagramma di stato

Attenzione: tutte le transizioni non indicate conducono allo stato di rifiuto:

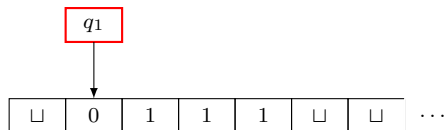
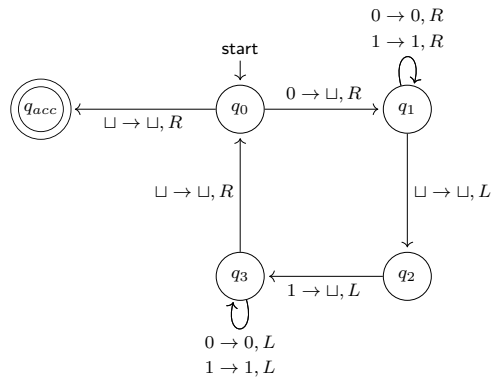
1. Se leggi \sqcup , vai a 5. Se leggi 0, scrivi \sqcup e vai a 2.
2. Spostati a destra di tutti gli 0 e 1. Al primo \sqcup , spostati a sinistra e vai a 3
3. Se leggi 1, scrivi \sqcup e vai a 4.
4. Spostati a sinistra di tutti gli 0 e 1. Se leggi \sqcup , spostati a destra e vai a 1.
5. Accetta.



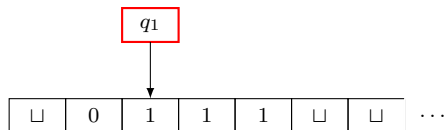
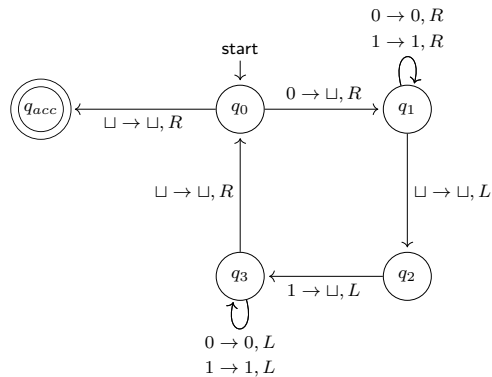
Esempio di stringa non accettata: 00111



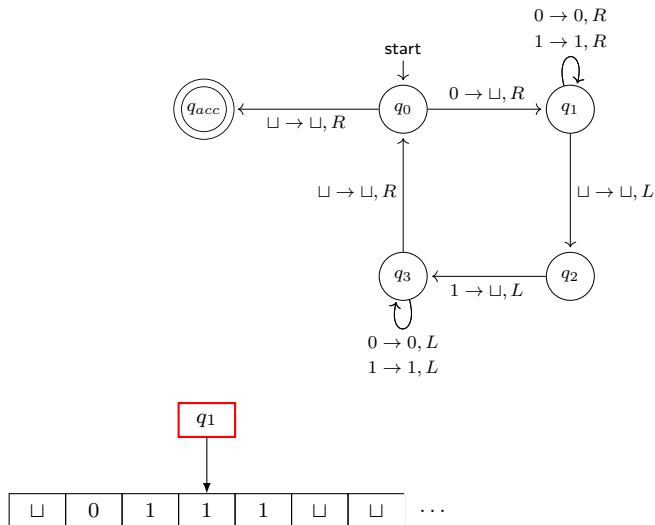
Esempio di stringa non accettata: 00111



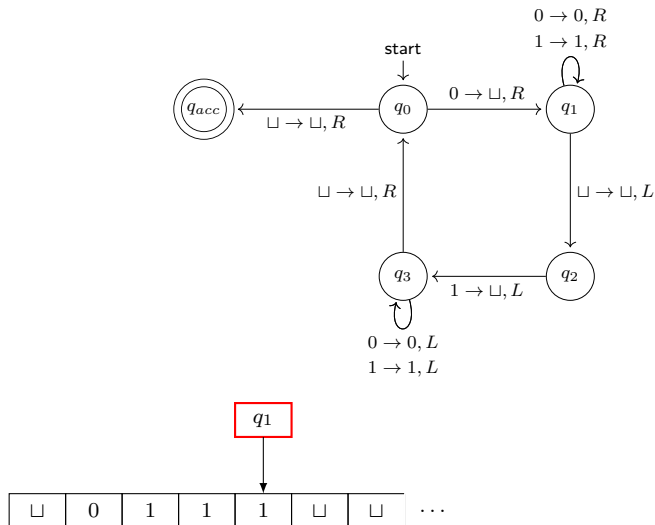
Esempio di stringa non accettata: 00111



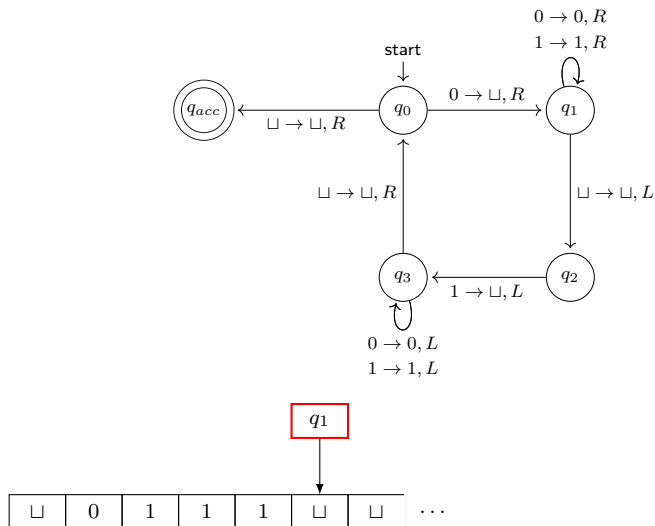
Esempio di stringa non accettata: 00111



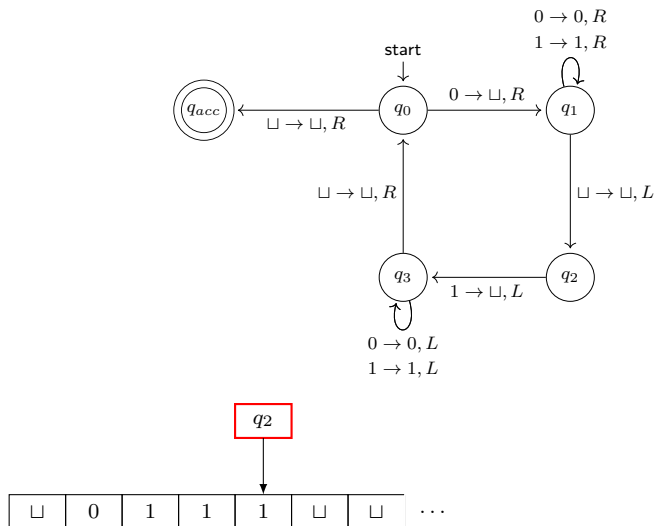
Esempio di stringa non accettata: 00111



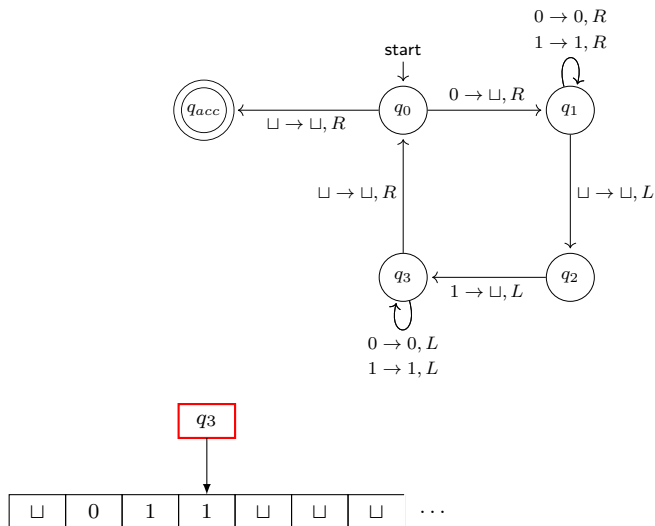
Esempio di stringa non accettata: 00111



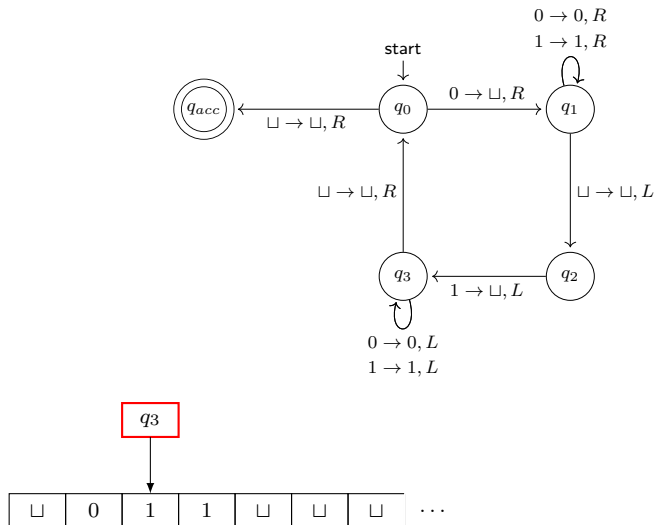
Esempio di stringa non accettata: 00111



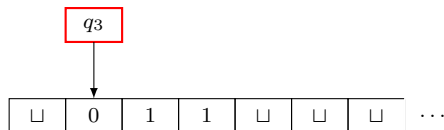
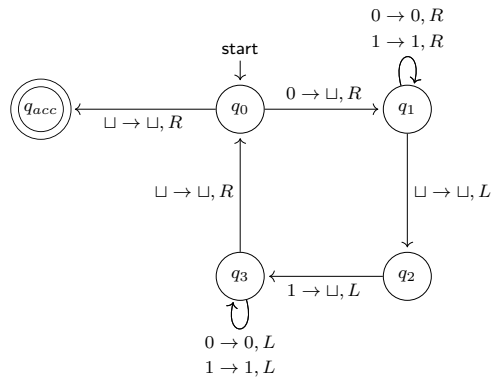
Esempio di stringa non accettata: 00111



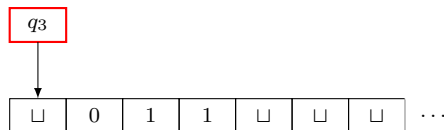
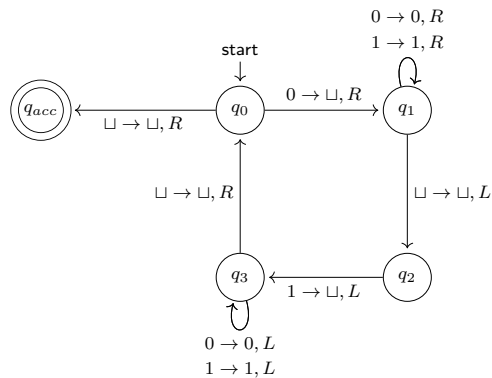
Esempio di stringa non accettata: 00111



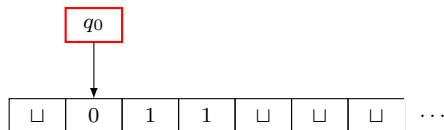
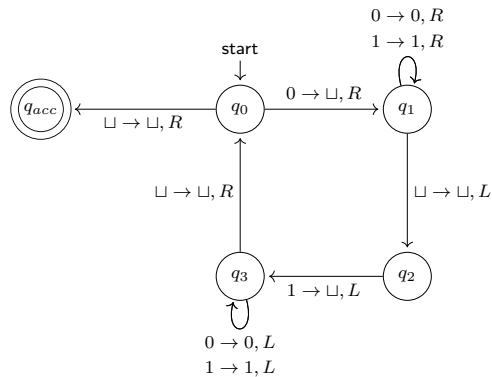
Esempio di stringa non accettata: 00111



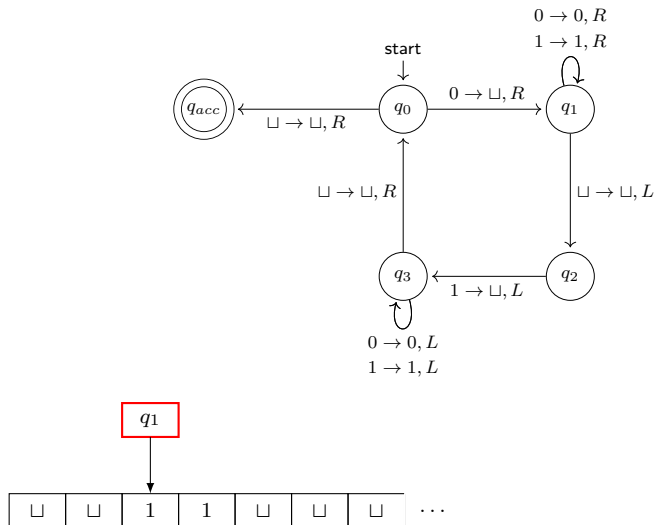
Esempio di stringa non accettata: 00111



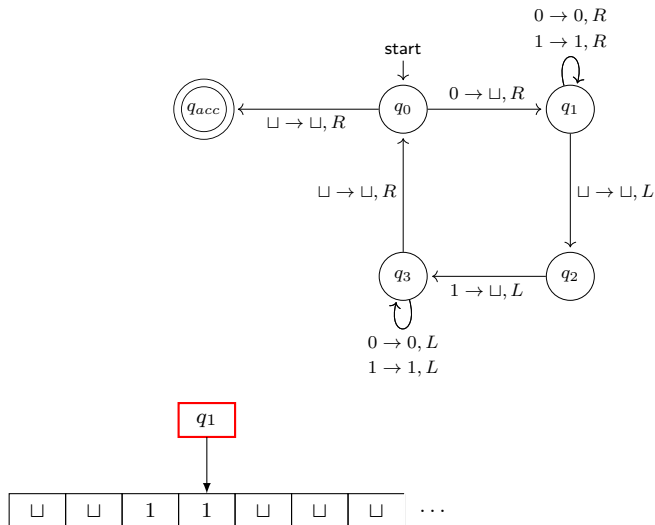
Esempio di stringa non accettata: 00111



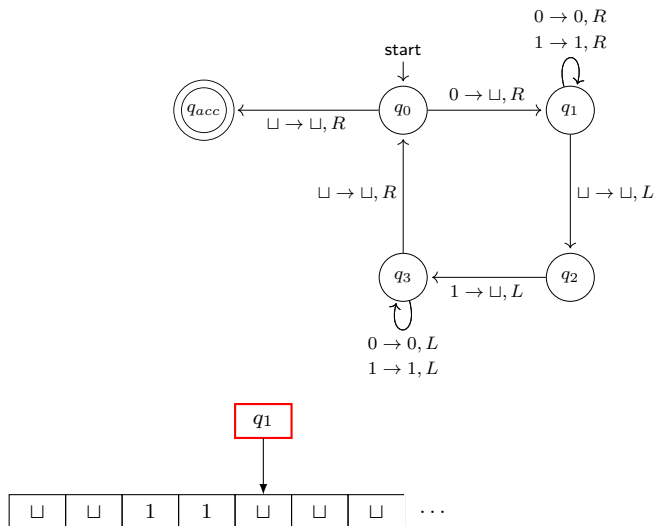
Esempio di stringa non accettata: 00111



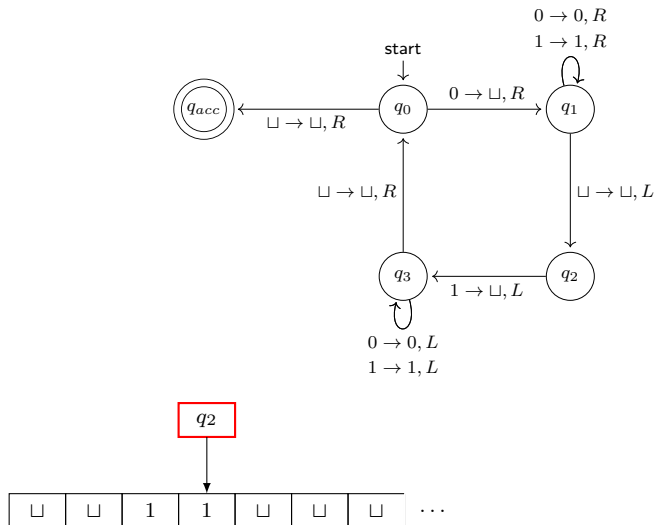
Esempio di stringa non accettata: 00111



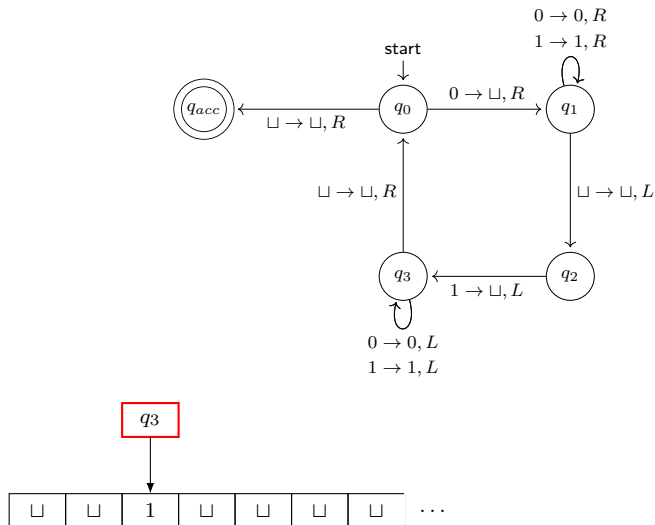
Esempio di stringa non accettata: 00111



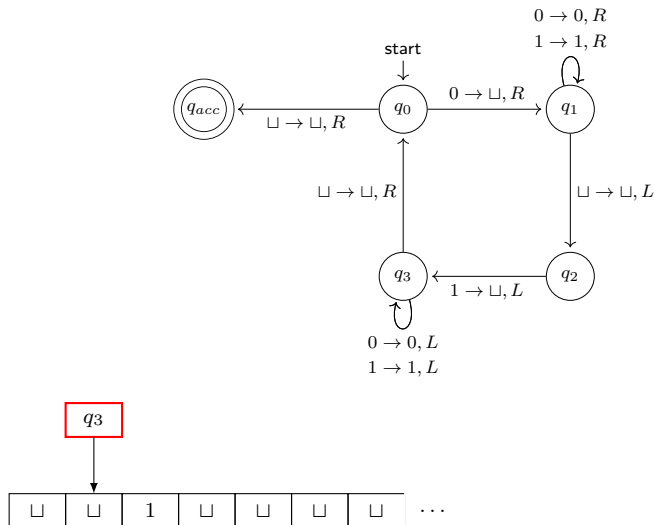
Esempio di stringa non accettata: 00111



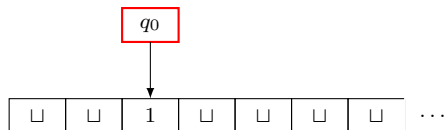
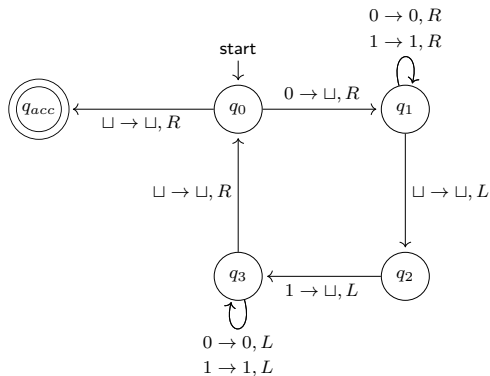
Esempio di stringa non accettata: 00111



Esempio di stringa non accettata: 00111

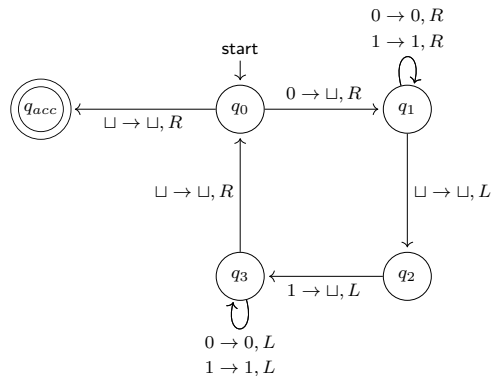


Esempio di stringa non accettata: 00111

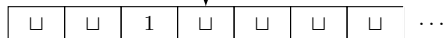


Non è indicata la transizione per 1: si va nello stato di rifiuto.

Esempio di stringa non accettata: 00111



q_{rej}



Non è indicata la transizione per 1: si va nello stato di rifiuto.

Progettiamo ora una macchina di Turing per il seguente linguaggio:

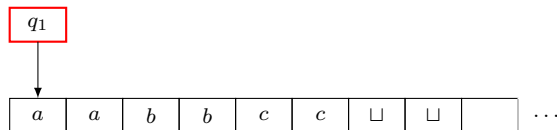
$$L = \{a^n b^n c^n \mid n \geq 0\}.$$

Idea

- ▶ Leggiamo la stringa da sinistra a destra: sovrascriviamo la prima occorrenza di a con X , la prima occorrenza di b con Y e la prima occorrenza di c con Z .
- ▶ Facciamo tornare indietro la testina e iteriamo.
- ▶ La stringa è di tipo $a^n b^n c^n$ se e solo se ad ogni lettura da sinistra a destra troviamo sempre una a , una b e una c ; tranne che nell'ultima iterazione nella quale non dovremo trovare né a , né b né c .

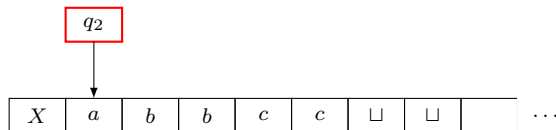
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



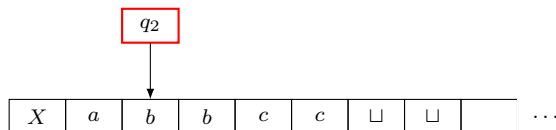
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



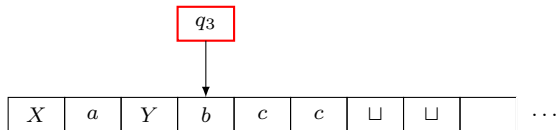
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



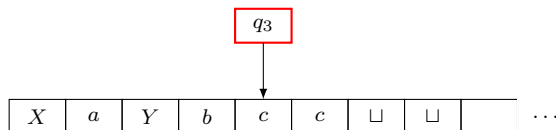
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



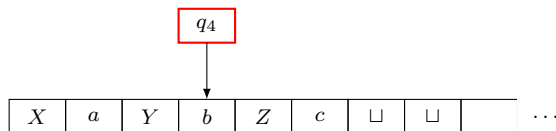
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



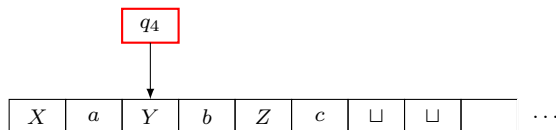
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



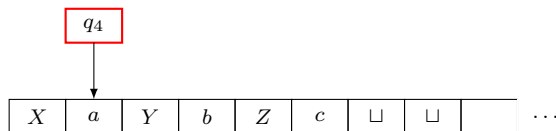
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



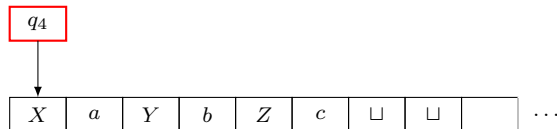
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



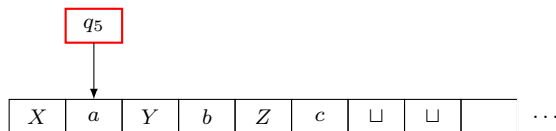
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



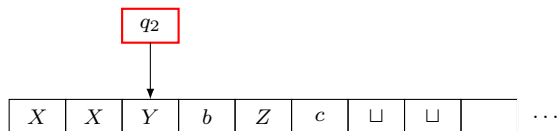
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



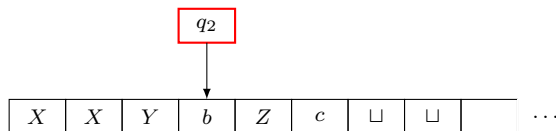
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



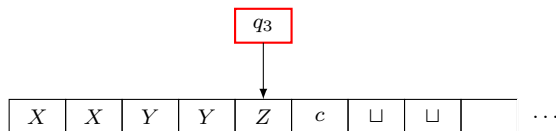
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



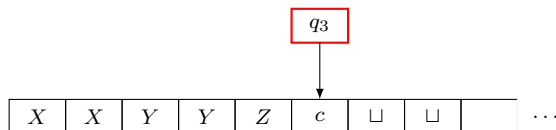
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



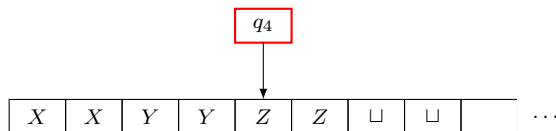
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



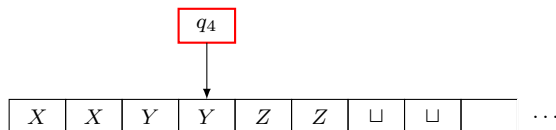
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.

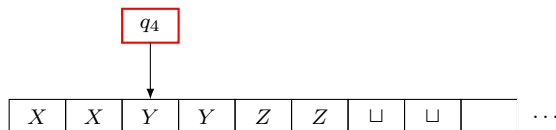


MdT per le stringhe $a^n b^n c^n$

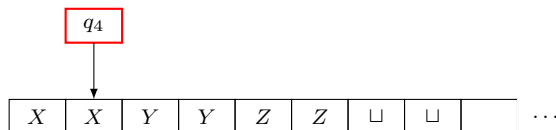
1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



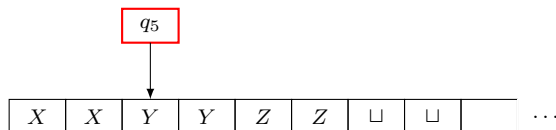
1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.

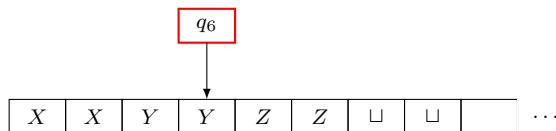


1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



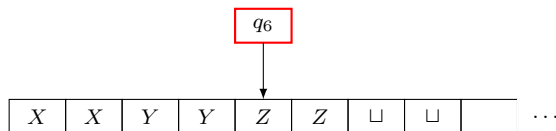
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



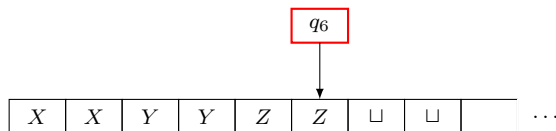
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



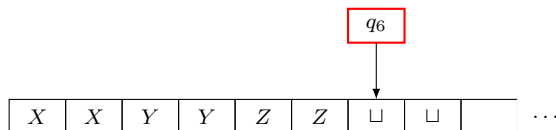
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



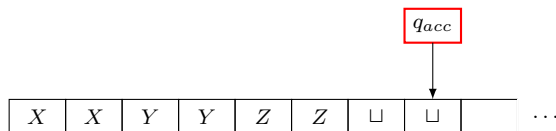
MdT per le stringhe $a^n b^n c^n$

1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.

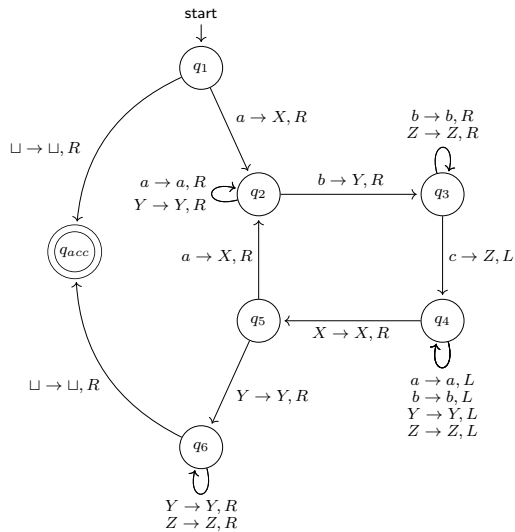


MdT per le stringhe $a^n b^n c^n$

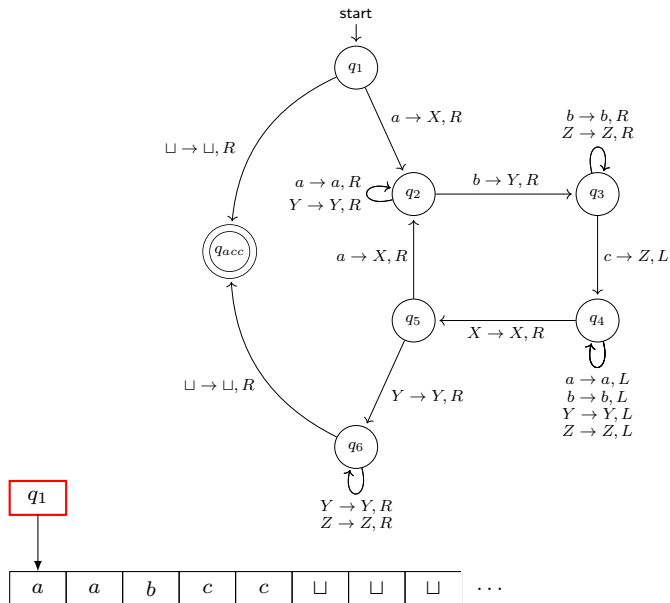
1. Se leggi \sqcup vai a 7; se leggi a scrivi X , vai a destra e passa a 2
2. Fin quando leggi a o Y vai a destra; se leggi b scrivi Y , vai a destra e passa a 3.
3. Fin quando leggi b o Z vai a destra; se leggi c scrivi Z , vai a sinistra e passa a 4.
4. Fin quando leggi A, b, Y o Z vai a sinistra; se leggi X vai a destra e passa a 5.
5. Se leggi a scrivi X , vai a destra e passa a 2; se leggi Y vai a destra e passa a 6.
6. Fin quando leggi Y o Z vai a destra; se leggi \sqcup , vai a 7.
7. Accetta.



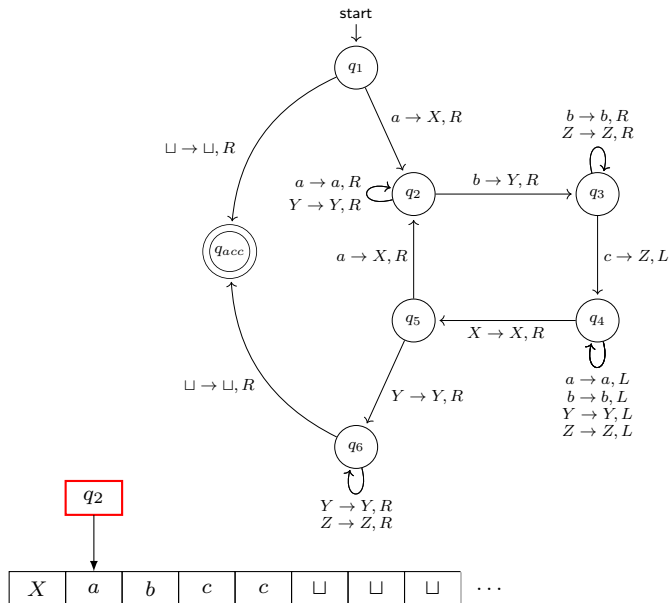
Esempio di stringa accettata: *aabbcc*



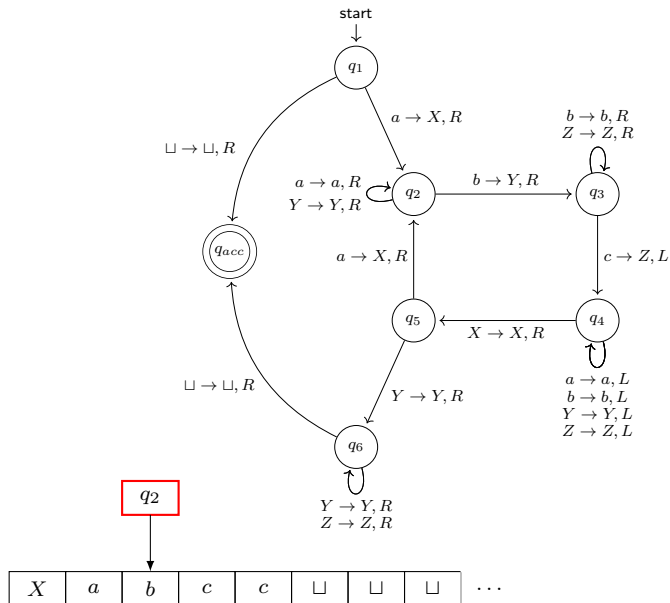
Esempio di stringa non accettata: *aabcc*



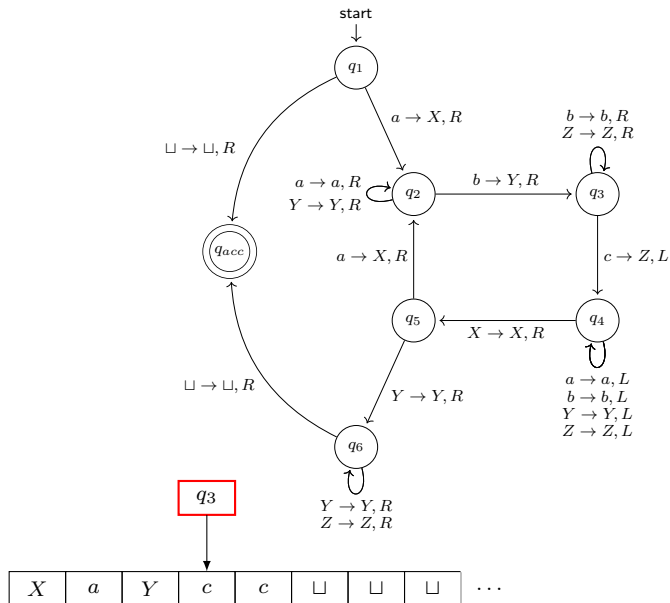
Esempio di stringa non accettata: *aabcc*



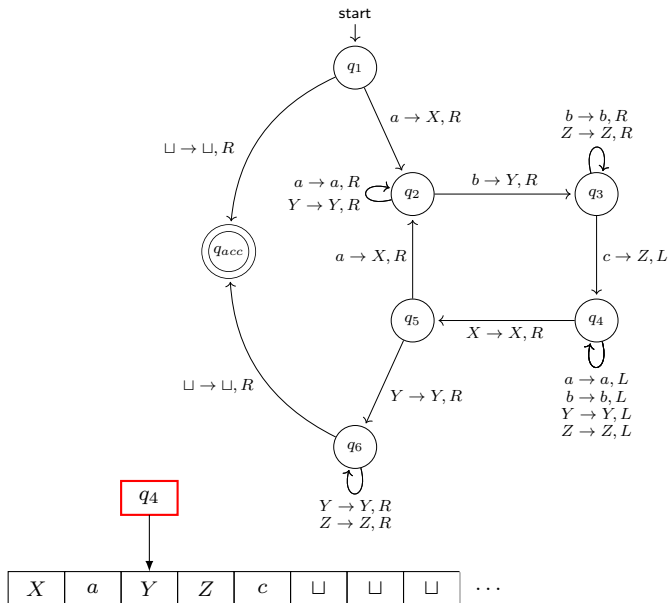
Esempio di stringa non accettata: *aabcc*



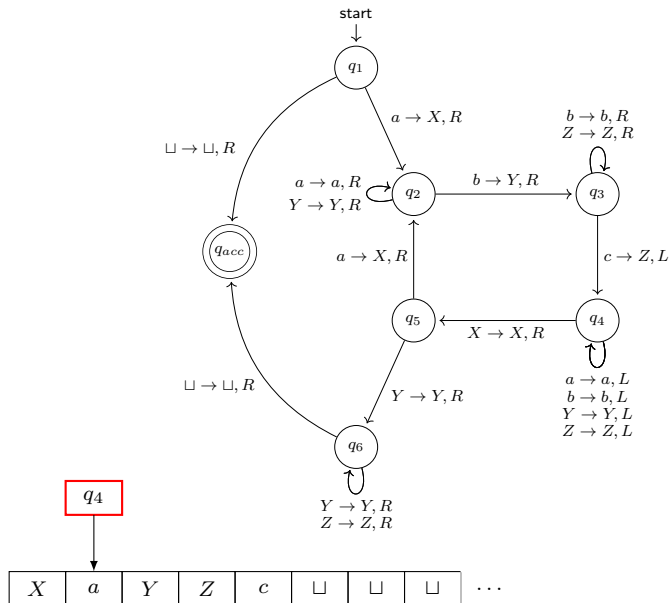
Esempio di stringa non accettata: *aabcc*



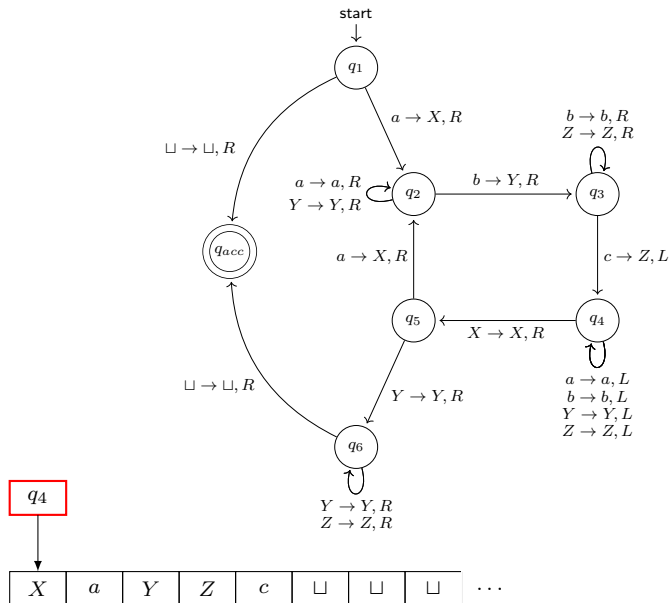
Esempio di stringa non accettata: *aabcc*



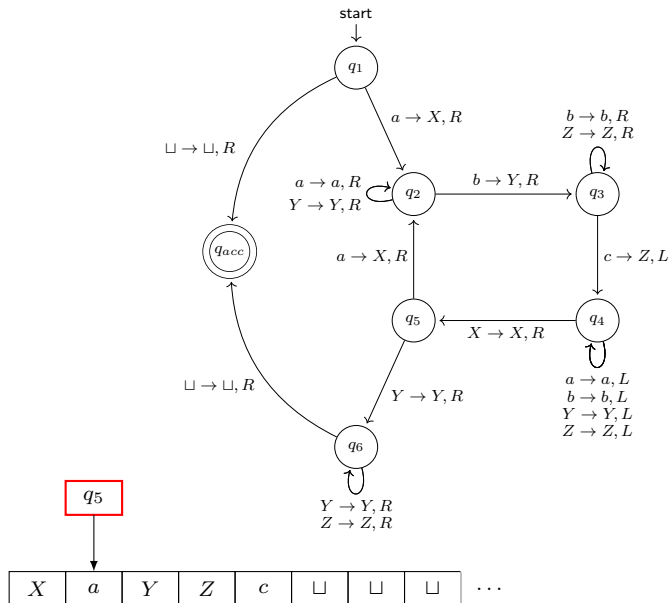
Esempio di stringa non accettata: *aabcc*



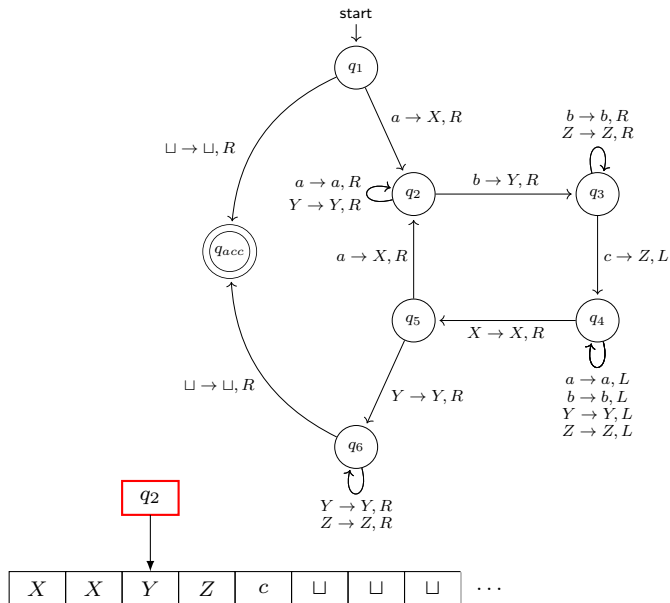
Esempio di stringa non accettata: *aabcc*



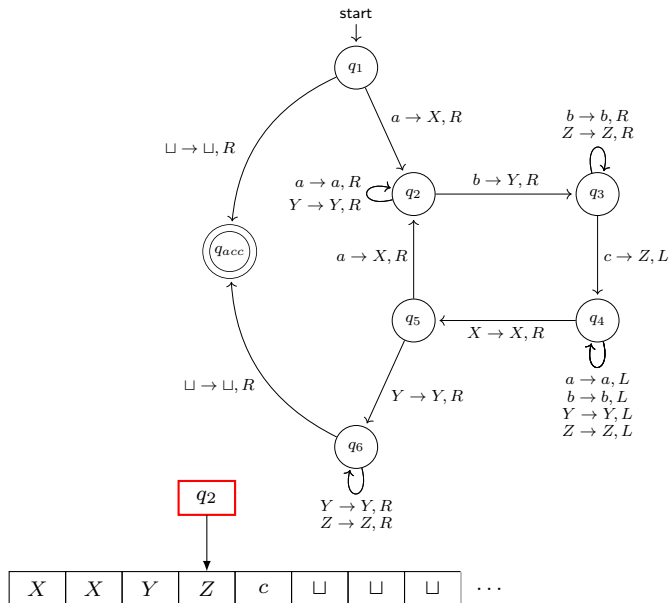
Esempio di stringa non accettata: *aabcc*



Esempio di stringa non accettata: *aabcc*



Esempio di stringa non accettata: *aabcc*



Esempio di stringa non accettata: *aabcc*

