

Macchine di TURING



Alan Mathison Turing (1912 – 1954)

Macchine di TURING



Alan Mathison Turing (1912 – 1954)

matematico, logico e crittanalista britannico, considerato uno dei padri dell'informatica e uno dei più grandi matematici del Novecento.

Introdusse la macchina ideale ed il test omonimi.

Fu uno dei più brillanti decrittatori che operavano in Inghilterra, durante la seconda guerra mondiale, per decifrare i codici delle Potenze dell'Asse.

Decifrò Enigma, il codice usato dai sottomarini tedeschi

Morì mangiando una mela al cianuro, in seguito ad una persecuzione omofobica condotta nei suoi confronti.

Macchine di TURING



Alan Mathison Turing (1912 – 1954)

matematico, logico e crittanalista britannico, considerato uno dei padri dell'informatica e uno dei più grandi matematici del Novecento.

Introdusse la macchina ideale ed il test omonimi.

Fu uno dei più brillanti decrittatori che operavano in Inghilterra, durante la seconda guerra mondiale, per decifrare i codici delle Potenze dell'Asse.

Decifrò Enigma, il codice usato dai sottomarini tedeschi

Morì mangiando una mela al cianuro, in seguito ad una persecuzione omofobica condotta nei suoi confronti.

Biografia: Andrew Hodges, Alan Turing: una biografia, Torino, Bollati Boringhieri, 2006

Macchine di TURING

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

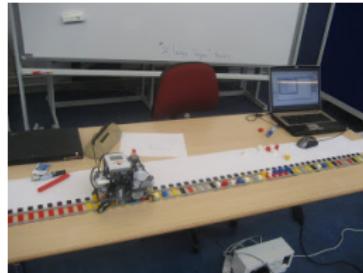
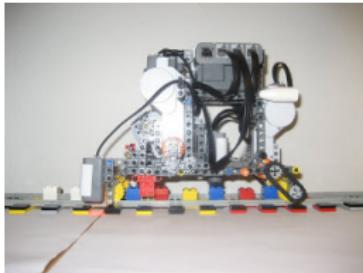
Vedremo le Macchine di Turing (MdT) e discuteremo alcune delle loro proprietà

Turing Machine è una macchina a stati finiti con un nastro semi-infinito

La testina può muoversi in entrambe le direzioni.

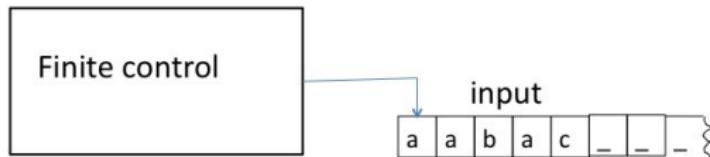
Può leggere, scrivere in ogni cella del nastro

Quando la MdT raggiunge uno stato accept/reject allora accetta/rifiuta immediatamente



MdT in Lego – Aharus University

Schema di MdT



Testina muove in entrambe le direzioni
Il simbolo _ segna la fine dell' input.

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

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 ($_ \notin \Sigma$)

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 ($_ \notin \Sigma$)
- ▶ Γ : **Alfabeto del nastro** ($_ \in \Gamma$, $\Sigma \subset \Gamma$)

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

Funzione di transizione di 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

Funzione di transizione di 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 un quadrato del nastro contenente un qualche simbolo $\gamma \in \Gamma$:

Funzione di transizione di 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 un quadrato del nastro contenente un qualche simbolo $\gamma \in \Gamma$:
- ▶ la funzione 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 MdT

Il range della funzione di transizione sono triple (q', γ', d) con

- ▶ $q' \in Q$

Funzione di transizione di MdT

Il range della funzione di transizione sono triple (q', γ', d) con

- ▶ $q' \in Q$
- ▶ $\gamma' \in \Gamma$ è simbolo scritto dalla testina sulla cella del nastro su cui la testina si trova ALL' INIZIO della transizione

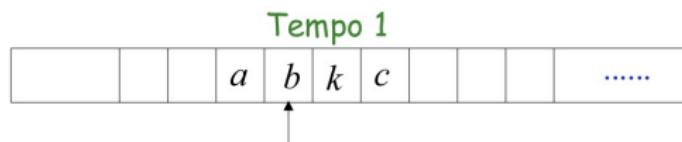
Funzione di transizione di MdT

Il range della funzione di transizione sono triple (q', γ', d) con

- ▶ $q' \in Q$
- ▶ $\gamma' \in \Gamma$ è simbolo scritto dalla testina sulla cella del nastro su cui la testina si trova ALL' INIZIO della transizione
- ▶ $d \in \{L, R\}$ è la direzione in cui la testina muove un passo

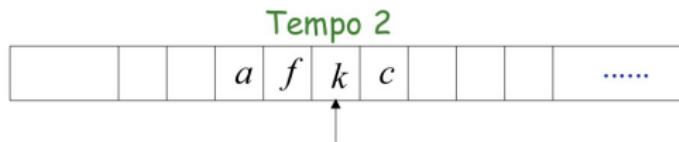
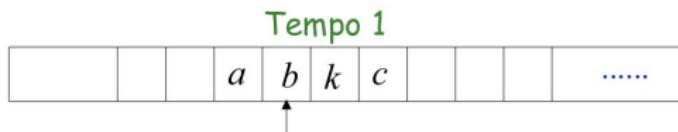
Funzione di transizione di MdT

Esempio:



1. Legge a
2. Scrive k
3. Si muove a sinistra

Funzione di transizione di MdT

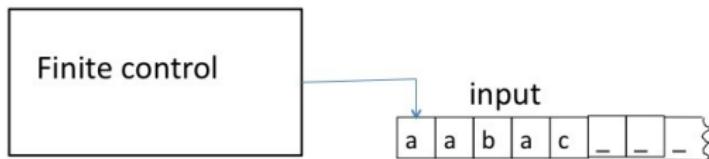


1. Legge b
2. Scrive f
3. Si muove a destra

Computazione di MdT

La computazione parte sempre

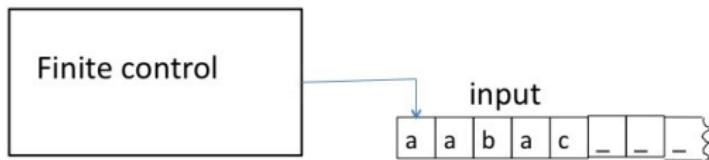
- ▶ da stato iniziale q_0



Computazione di MdT

La computazione parte sempre

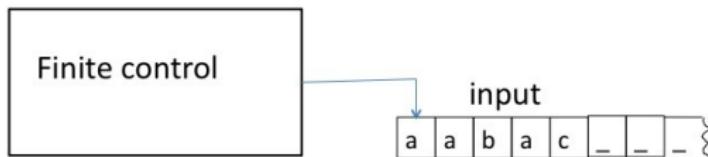
- ▶ da stato iniziale q_0
- ▶ con input posizionato sulla parte più a sinistra del nastro: le prime n celle a sinistra, se n è lunghezza input



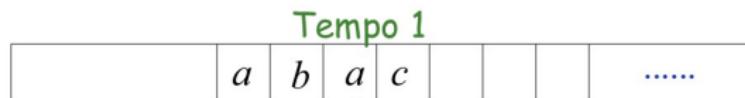
Computazione di MdT

La computazione parte sempre

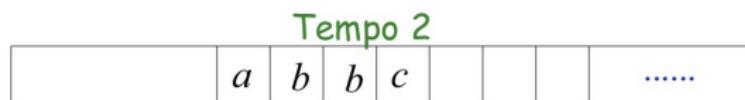
- ▶ da stato iniziale q_0
- ▶ con input posizionato sulla parte più a sinistra del nastro: le prime n celle a sinistra, se n è lunghezza input
- ▶ la testina si trova in prima cella a sinistra del nastro (cella 0)



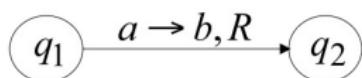
Funzione di transizione di MdT



q_1



q_2

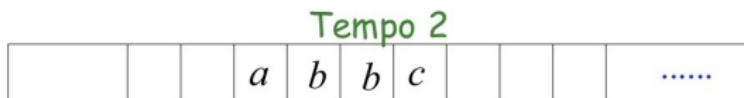


Funzione di transizione di MdT

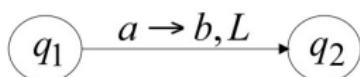
Esempio:



q_1



q_2



Computazione di MdT

La computazione termina quando M raggiunge

Computazione di MdT

La computazione termina quando M raggiunge

- ▶ stato accettazione q_{accept} : **Computazione Accept**

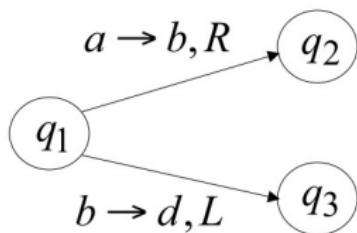
Computazione di MdT

La computazione termina quando M raggiunge

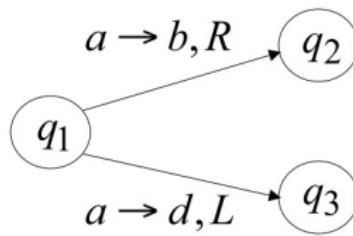
- ▶ stato accettazione q_{accept} : **Computazione Accept**
- ▶ stato rifiuto q_{reject} : **Computazione Reject**

Macchina di Turing deterministica

Permesso



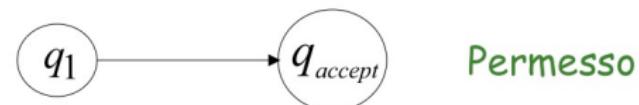
Non permesso



Non ci sono epsilon-transizioni

Esempio

Stati di arresto

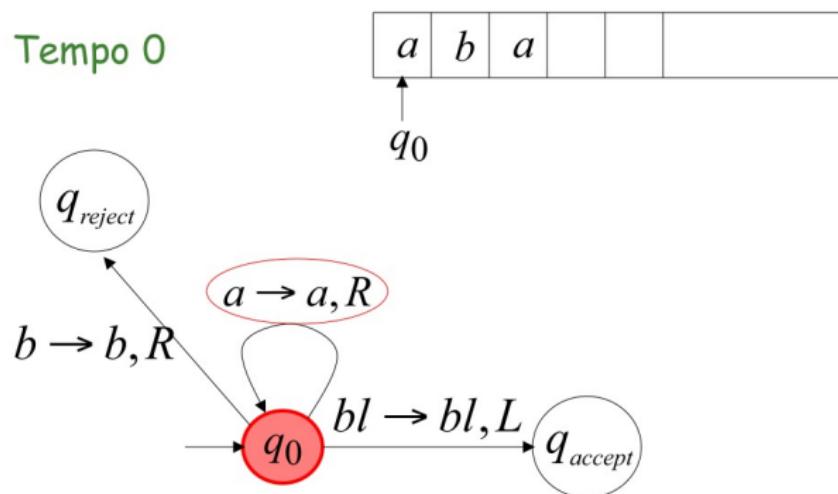


- Gli stati di arresto non hanno archi uscenti
- In uno stato di arresto la computazione termina

Esempio

Esempio di stringa non accettata

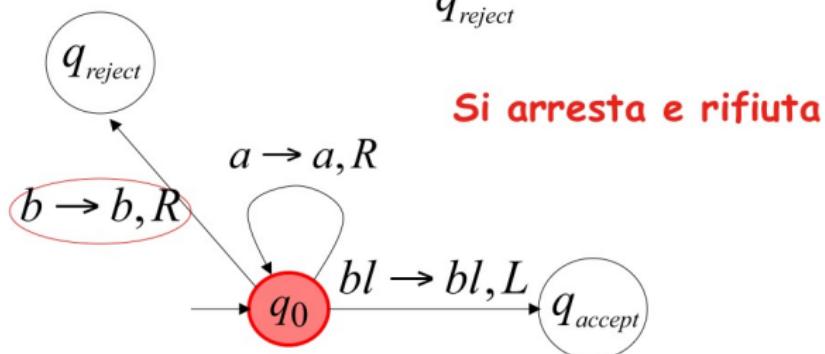
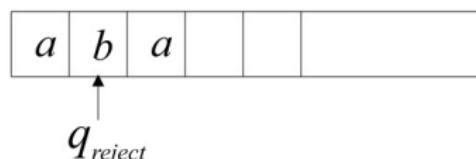
Tempo 0



Esempio

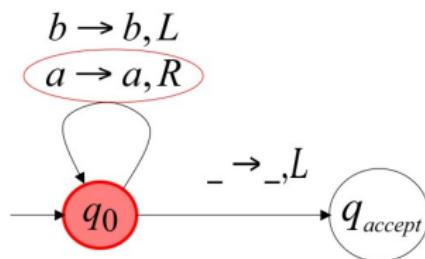
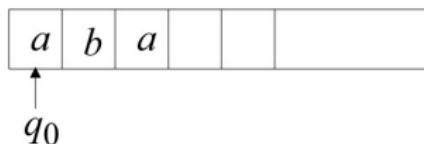
Esempio di stringa non accettata

Tempo 1



Esempio di non terminazione

Tempo 0



11

Lo stato q_{accept} non raggiungibile

La computazione non termina. L' input non viene accettato

Esempio: Strategia per accettare $\{a^n b^n\}$

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

Esempio: Strategia per accettare $\{a^n b^n\}$

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 è a, scrivi $_$ e vai a 2.

Esempio: Strategia per accettare $\{a^n b^n\}$

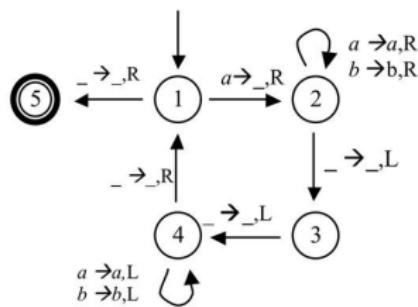
Cancella ripetutamente: prima occorrenza di (a) e ultima di (b)
se la stringa era del tipo $a^n b^n$, non rimangono simboli
Cinque passi

1. Se leggi $_$, vai a 5. Se è a, scrivi $_$ e vai a 2.
2. Spostati a destra (R) di tutti a e b. All'ultimo b, muovi L e vai a 3
3. se simbolo è b, scrivi $_$ e vai a 4.
4. Spostati a sinistra (R) di tutti a e b. All'ultimo $_$, muovi R e vai a 1.
5. Accept.

Funzione di transizione di MdT

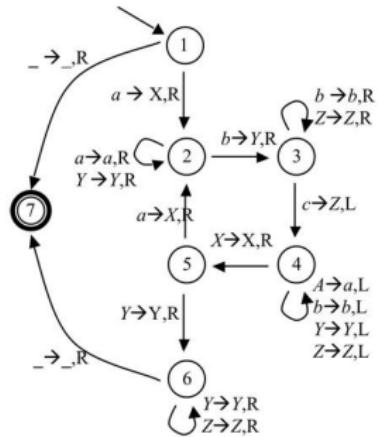
1. Se leggi $_$, vai a 5. Se è a, scrivi $_$ e vai a 2.
2. Spostati a destra (R) di tutti a e b. All'ultimo b, muovi L e vai a 3
3. se simbolo è b, scrivi $_$ e vai a 4.
4. Spostati a sinistra (R) di tutti a e b. All'ultimo $_$, muovi R e vai a 1.
5. Accept.

$$\{a^n b^n\}$$

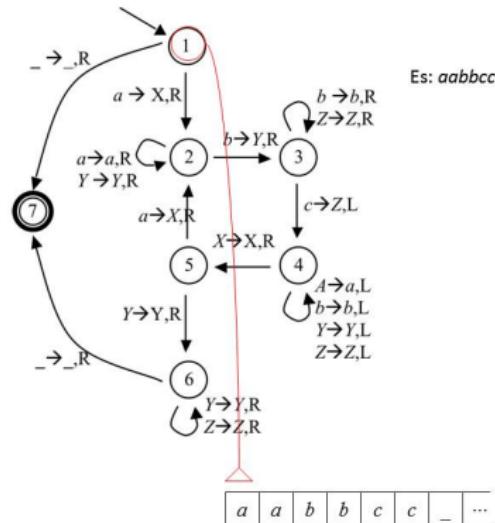


Funzione di transizione di MdT

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

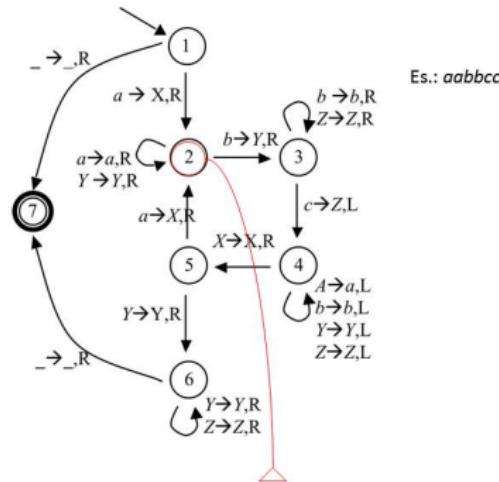


Funzione di transizione di MdT



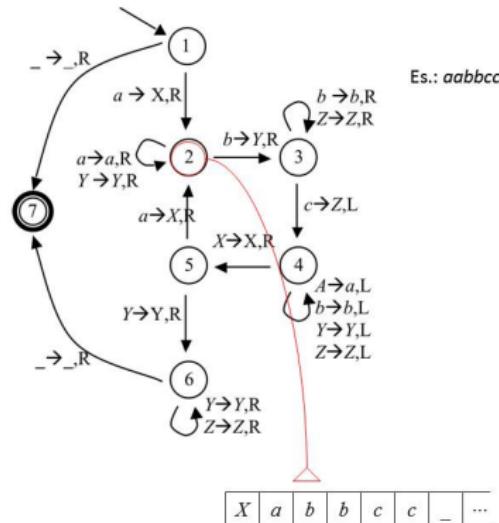
Esempio: $aabbcc$

Funzione di transizione di MdT

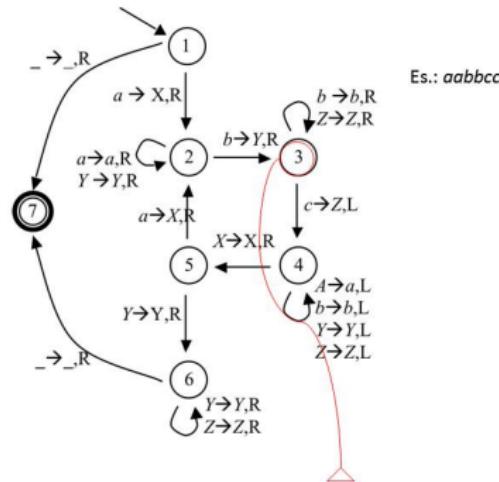


Es.: $aabbcc$

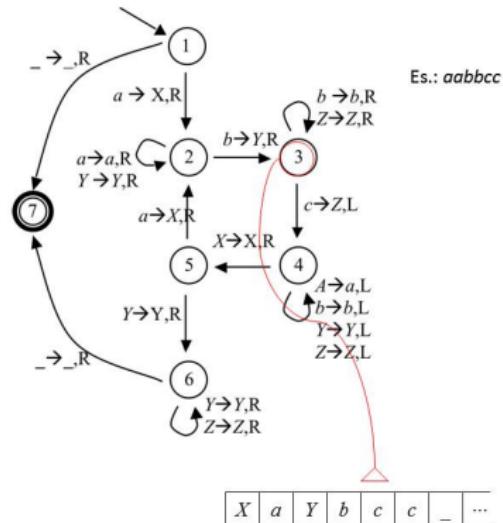
Funzione di transizione di MdT



Funzione di transizione di MdT

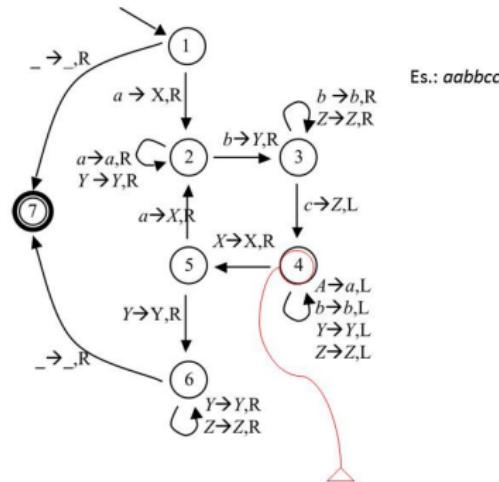


Funzione di transizione di MdT



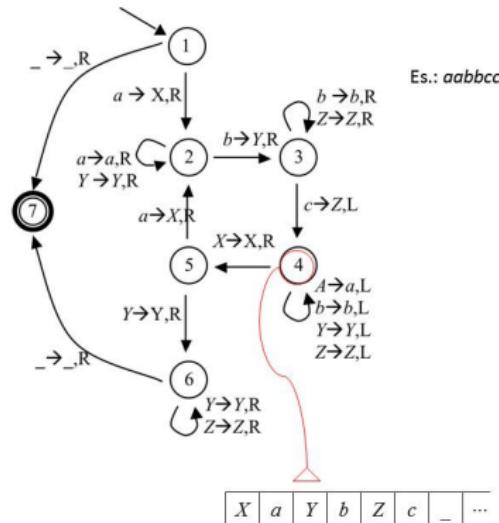
Es.: *aabbcc*

Funzione di transizione di MdT



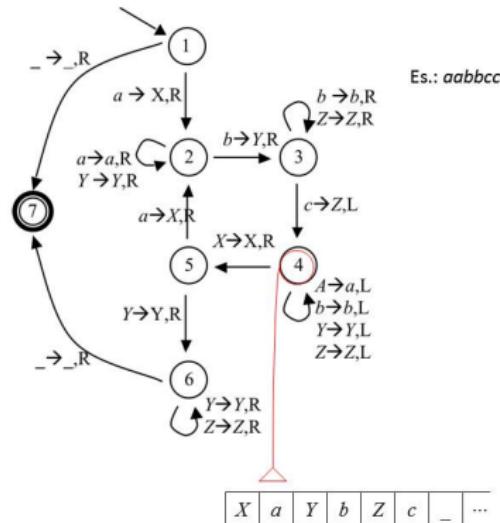
Es.: $aabbcc$

Funzione di transizione di MdT

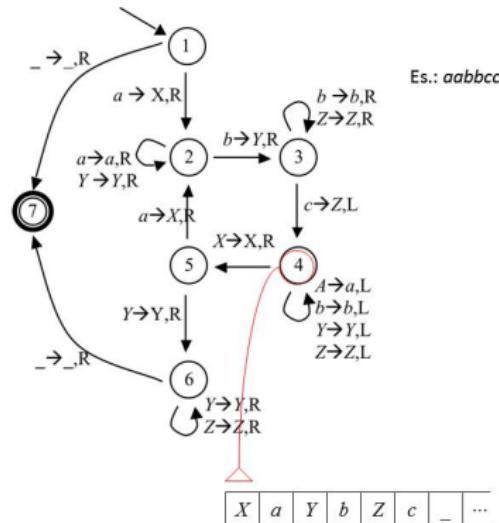


Es.: $aabbcc$

Funzione di transizione di MdT

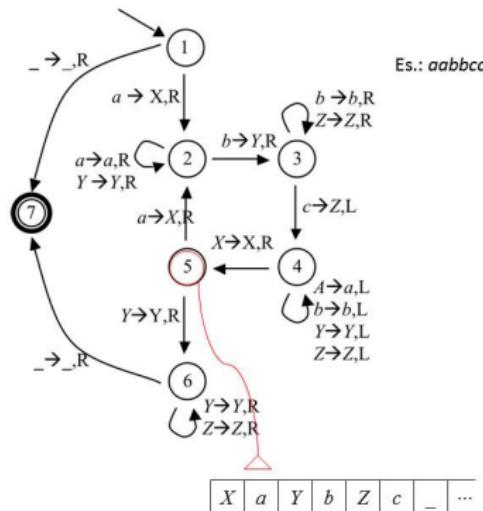


Funzione di transizione di MdT

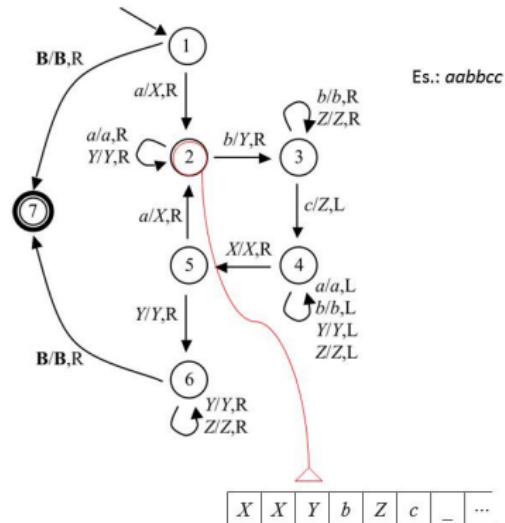


Es.: $aabbcc$

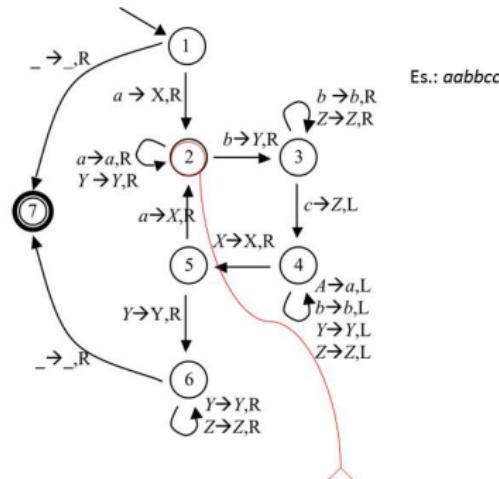
Funzione di transizione di MdT



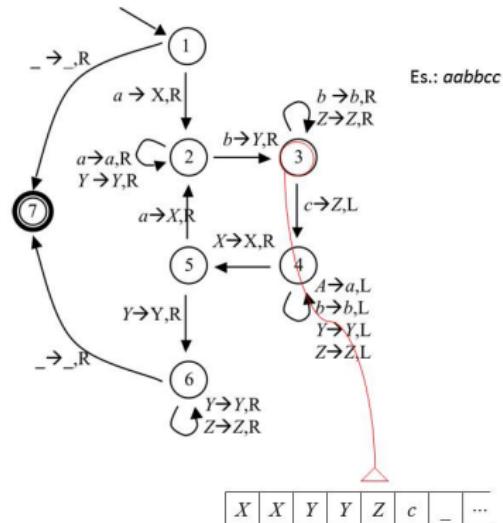
Funzione di transizione di MdT



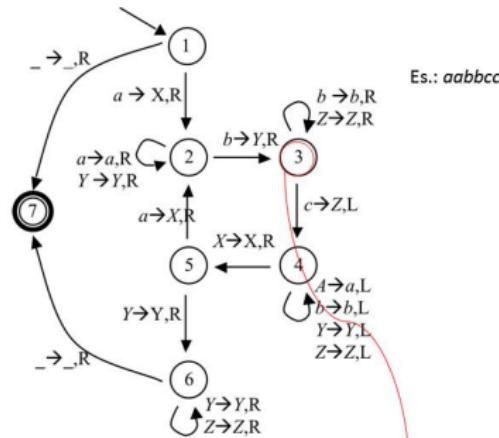
Funzione di transizione di MdT



Funzione di transizione di MdT



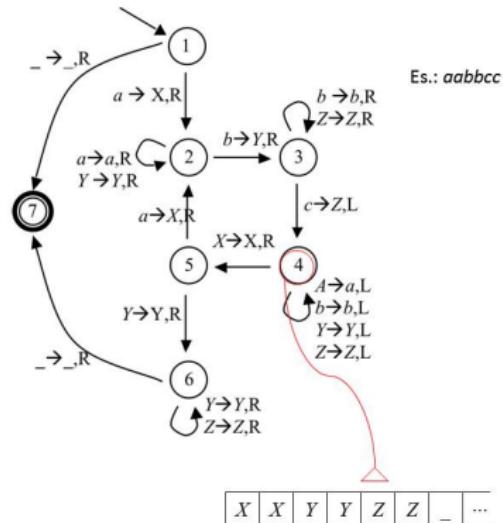
Funzione di transizione di MdT



Es.: $aabbcc$

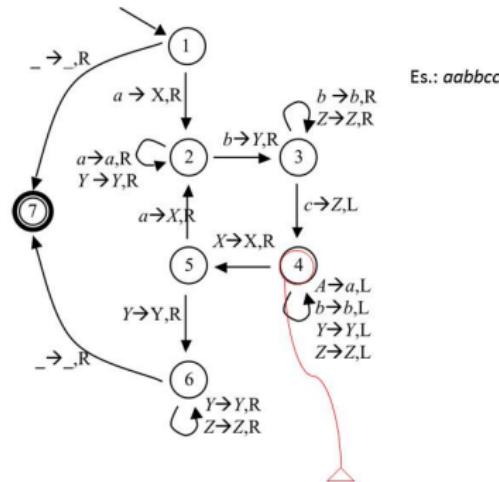
X	X	Y	Y	Z	c	$-$	\dots
-----	-----	-----	-----	-----	-----	-----	---------

Funzione di transizione di MdT

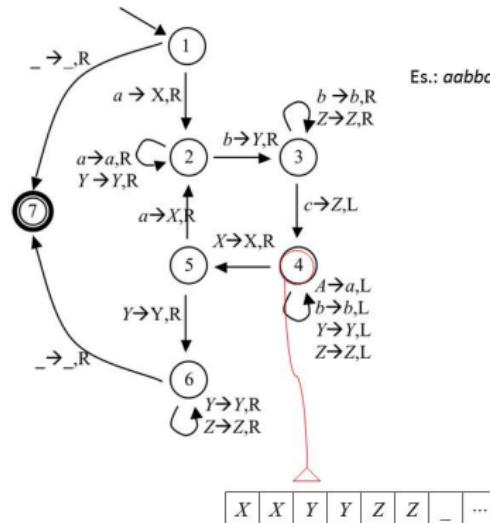


Es.: $aabbcc$

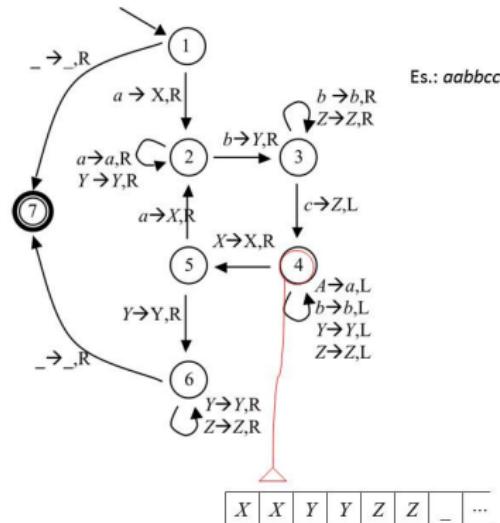
Funzione di transizione di MdT



Funzione di transizione di MdT

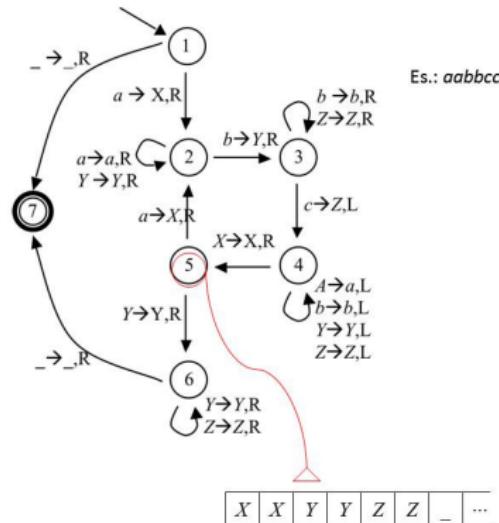


Funzione di transizione di MdT

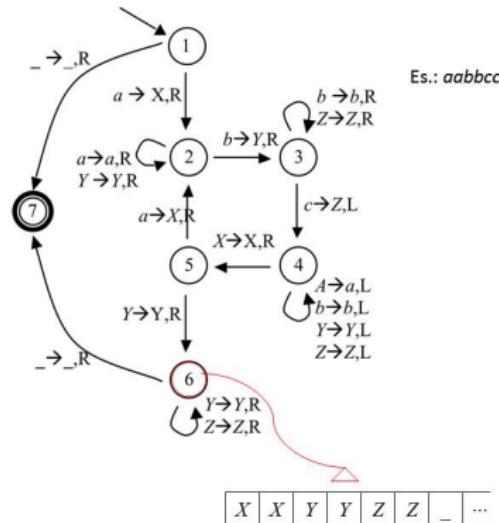


Es.: $aabbcc$

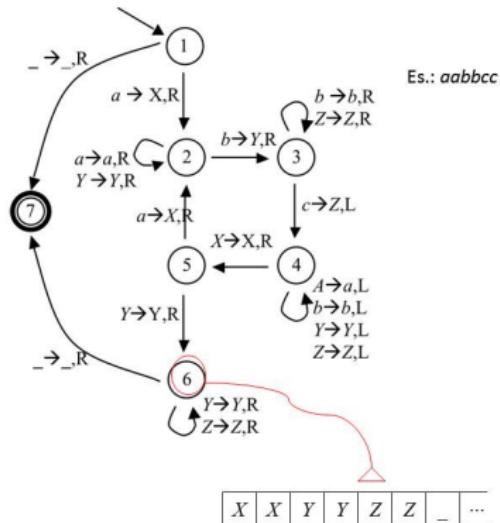
Funzione di transizione di MdT



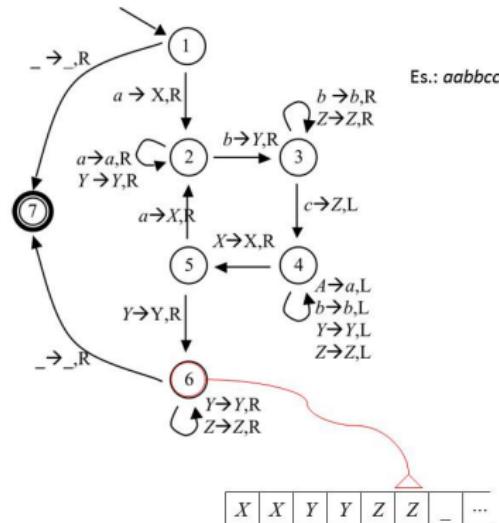
Funzione di transizione di MdT



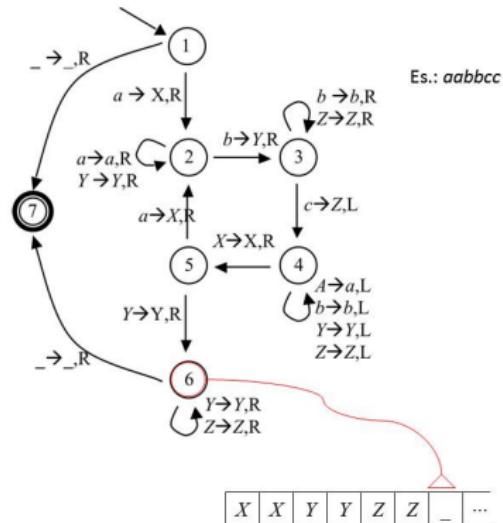
Funzione di transizione di MdT



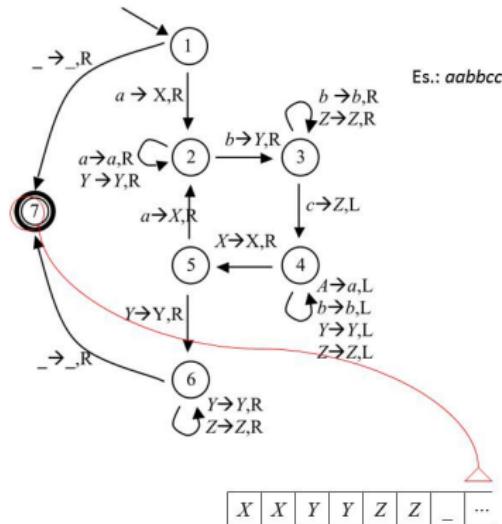
Funzione di transizione di MdT



Funzione di transizione di MdT



Funzione di transizione di MdT



Configurazioni

Configurazione MdT \equiv descrizione concisa di stato e contenuto
nastro

$$C = uqv$$

- ▶ stato di M è q

Configurazioni

Configurazione MdT \equiv descrizione concisa di stato e contenuto
nastro

$$C = uqv$$

- ▶ stato di M è q
- ▶ contenuto nastro (sinistra – destra) è uv

Configurazioni

Configurazione MdT \equiv descrizione concisa di stato e contenuto nastro

$$C = uqv$$

- ▶ stato di M è q
- ▶ contenuto nastro (sinistra – destra) è uv
- ▶ testina sul primo (cioè più a sinistra) simbolo di v
(su primo blank _ se $v = \epsilon$)

Configurazioni

Configurazione MdT \equiv descrizione concisa di stato e contenuto
nastro

$$C = uqv$$

- ▶ stato di M è q
- ▶ contenuto nastro (sinistra – destra) è uv
- ▶ testina sul primo (cioè più a sinistra) simbolo di v
(su primo blank _ se $v = \epsilon$)
- ▶ dopo v solo blanks _

Configurazioni

Configurazione cambia ad ogni mossa

$$C_1 \text{ produce } C_2 \quad (C_1 \rightarrow C_2)$$

se una mossa della MdT può far andare la macchina da C_1 a C_2

Es. Se $a, b, c \in \Gamma$, $u, v \in \Gamma^*$, $q_i, q_j \in Q$

$uaq_i bv \rightarrow uq_j acv$ se $\delta(q_i, b) = (q_j, c, L)$ **Mossa a sinistra**
oppure

$uaq_i bv \rightarrow uacq_j v$ se $\delta(q_i, b) = (q_j, c, R)$ **Mossa a destra**

Configurazioni:casi particolari

Configurazione $q_i bv \rightarrow q_j cv$
se testina ad inizio nastro e **Mossa a sinistra**:
non si può andare oltre inizio nastro

Configurazioni:casi particolari

Configurazione $q_i bv \rightarrow q_j cv$
se testina ad inizio nastro e **Mossa a sinistra**:
non si può andare oltre inizio nastro

Configurazione uaq_i equivalente a $uaq_i \underline{\quad}$:
parte vuota nastro \equiv riempita con $\underline{\quad}$

Configurazioni

Configurazione di start di M su input w

$$q_0 w$$

\Leftrightarrow stato q_0 , nastro contiene w , testina su prima cella del nastro

Configurazioni

Configurazione di start di M su input w

$$q_0 w$$

\Leftrightarrow stato q_0 , nastro contiene w , testina su prima cella del nastro

Configurazione Accept: configurazione che raggiunge stato q_{accept}

Configurazioni

Configurazione di start di M su input w

$$q_0 w$$

\Leftrightarrow stato q_0 , nastro contiene w , testina su prima cella del nastro

Configurazione Accept: configurazione che raggiunge stato q_{accept}

Configurazione Reject: configurazione che raggiunge stato q_{reject}

Configurazioni

Configurazione di start di M su input w

$$q_0 w$$

\Leftrightarrow stato q_0 , nastro contiene w , testina su prima cella del nastro

Configurazione Accept: configurazione che raggiunge stato q_{accept}

Configurazione Reject: configurazione che raggiunge stato q_{reject}

Configurazione di Halt : qualsiasi Configurazione Accept o Reject

Computazioni

MdT M accetta parola w se esiste una computazione (sequenza di configurazioni) di M ,

$$C_1, C_2, \dots, C_k$$

tali che

Computazioni

MdT M accetta parola w se esiste una computazione (sequenza di configurazioni) di M ,

$$C_1, C_2, \dots, C_k$$

tali che

1. $C_1 = q_0 w$ è la config. iniziale di M con input w

Computazioni

MdT M accetta parola w se esiste una computazione ([sequenza di configurazioni](#)) di M ,

$$C_1, C_2, \dots, C_k$$

tali che

1. $C_1 = q_0 w$ è la config. iniziale di M con input w
2. C_i produce C_{i+1} per ogni $i = 1, \dots, k - 1$

Computazioni

MdT M accetta parola w se esiste una computazione (sequenza di configurazioni) di M ,

$$C_1, C_2, \dots, C_k$$

tali che

1. $C_1 = q_0 w$ è la config. iniziale di M con input w
2. C_i produce C_{i+1} per ogni $i = 1, \dots, k - 1$
3. C_k è la config. accept

Risultato Computazioni

Tre possibili **Risultati computazione:**

Risultato Computazioni

Tre possibili **Risultati computazione**:

1. M **accetta** – se si ferma in q_{accept}

Risultato Computazioni

Tre possibili **Risultati computazione**:

1. M **accetta** – se si ferma in q_{accept}
2. M **rifiuta** – se si ferma in q_{reject}

Risultato Computazioni

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.

Linguaggio

Linguaggio di M , è l'insieme delle stringhe che M accetta;
denotato con $L(M)$

Linguaggio

Linguaggio di M , è l'insieme delle stringhe che M accetta; denotato con $L(M)$

Linguaggio Turing riconoscibile se esiste una macchina di Turing che lo riconosce

Deciders

E' difficile dire se MdT è in loop

Preferiamo

deciders: macchine che si fermano su **ogni input**

Deciders

E' difficile dire se MdT è in loop

Preferiamo

deciders: macchine che si fermano su **ogni input**

Decider **decide** linguaggio L se riconosce L

Linguaggio si dice **Turing riconoscibile** se esiste una MdT che lo riconosce

Deciders

E' difficile dire se MdT è in loop

Preferiamo

deciders: macchine che si fermano su **ogni input**

Decider **decide** linguaggio L se riconosce L

Linguaggio si dice **Turing riconoscibile** se esiste una MdT che lo riconosce

Un linguaggio si dice **Turing decidibile** se esiste una MdT che lo decide.

Esempio

Consideriamo il linguaggio

$$L = \{0^{2^n} \mid n > 0\}$$

insieme stringhe di 0 la cui lunghezza è potenza di 2

Nota. Il linguaggio non è regolare

Vogliamo costruire una MdT M_2 che lo decide.

Esempio

Sia w l'input, M_2 deve:

1. Scorrere il nastro da sinistra a destra cancellando ogni SECONDO 0

Esempio

Sia w l'input, M_2 deve:

1. Scorrere il nastro da sinistra a destra cancellando ogni SECONDO 0
2. Se rimane solo uno 0, ACCEPT

Esempio

Sia w l'input, M_2 deve:

1. Scorrere il nastro da sinistra a destra cancellando ogni SECONDO 0
2. Se rimane solo uno 0, ACCEPT
3. Se rimane un numero di 0 dispari ≥ 3 , REJECT

Esempio

Sia w l'input, M_2 deve:

1. Scorrere il nastro da sinistra a destra cancellando ogni SECONDO 0
2. Se rimane solo uno 0, ACCEPT
3. Se rimane un numero di 0 dispari ≥ 3 , REJECT
4. Riporta la testina all'inizio del nastro

Esempio

Sia w l'input, M_2 deve:

1. Scorrere il nastro da sinistra a destra cancellando ogni SECONDO 0
2. Se rimane solo uno 0, ACCEPT
3. Se rimane un numero di 0 dispari ≥ 3 , REJECT
4. Riporta la testina all'inizio del nastro
5. Go to passo 1.

Esempio – spiegazione

n è potenza di due sse ripetute divisioni per 2 danno resto 1

Esempio – spiegazione

n è potenza di due sse ripetute divisioni per 2 danno resto 1

M_2 :

1. Scorre il nastro da sinistra a destra cancellando ogni SECONDO 0

Divide per due il numero di 0

Esempio – spiegazione

n è potenza di due sse ripetute divisioni per 2 danno resto 1

M_2 :

1. Scorre il nastro da sinistra a destra cancellando ogni SECONDO 0

Divide per due il numero di 0

2. Se rimane solo uno 0, ACCEPT

3. Se rimane un numero di 0 dispari ≥ 3 , REJECT

Se rimane uno 0, OK

Se rimane numero dispari di 0, NON e' potenza

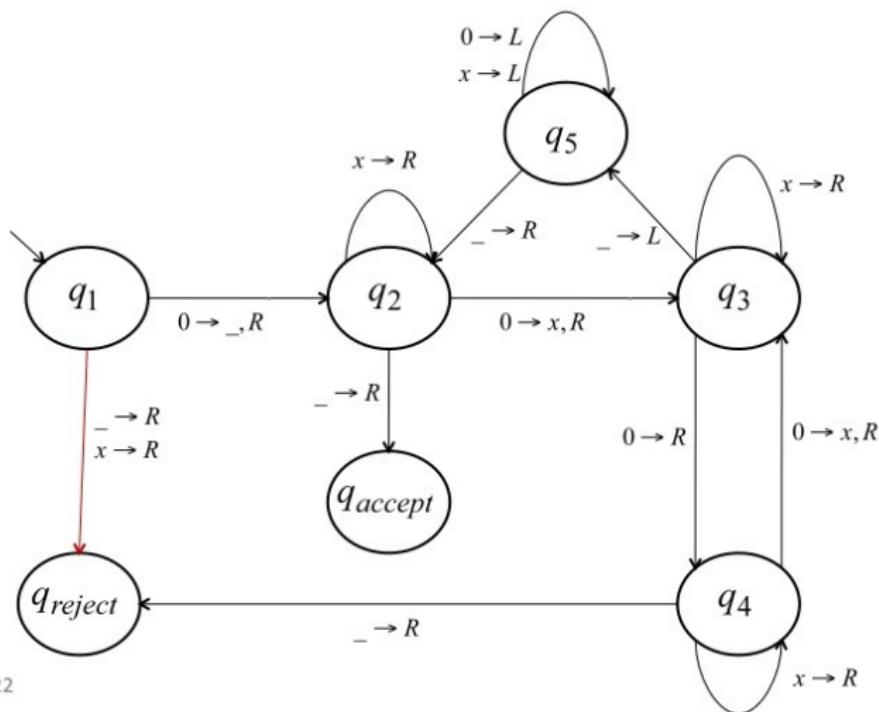
Esempio – spiegazione

n è potenza di due sse ripetute divisioni per 2 danno resto 1

M_2 :

1. Scorre il nastro da sinistra a destra cancellando ogni SECONDO 0
Divide per due il numero di 0
2. Se rimane solo uno 0, ACCEPT
3. Se rimane un numero di 0 dispari ≥ 3 , REJECT
Se rimane uno 0, OK
Se rimane numero dispari di 0, NON e' potenza
4. Riporta la testina all'inizio del nastro
5. Go to passo 1
Se rimane un numero pari: bisogna iterare

Costruiamo M_2 : $\Sigma = \{0\}$, $\Gamma = \{0, x, _\}$



Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0, 1\}^*\}$$

Idea per verificare se stringa il L :

Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0, 1\}^*\}$$

Idea per verificare se stringa il L :

- ▶ Leggi primo carattere

Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Idea per verificare se stringa il L :

- ▶ Leggi primo carattere
- ▶ memorizzalo e cancellalo

Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Idea per verificare se stringa il L :

- ▶ Leggi primo carattere
- ▶ memorizzalo e cancellalo
- ▶ cerca $\#$ e guarda carattere successivo

Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Idea per verificare se stringa il L :

- ▶ Leggi primo carattere
- ▶ memorizzalo e cancellalo
- ▶ cerca $\#$ e guarda carattere successivo
- ▶ se uguale al carattere memorizzato allora cancellalo

Esempio 2

Consideriamo il linguaggio

$$L = \{w\#w \mid w \in \{0, 1\}^*\}$$

Idea per verificare se stringa il L :

- ▶ Leggi primo carattere
- ▶ memorizzalo e cancellalo
- ▶ cerca $\#$ e guarda carattere successivo
- ▶ se uguale al carattere memorizzato allora cancellalo
- ▶ ritorna all'inizio al primo carattere non cancellato

Esempio 2

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Idea cont:

- ▶ ripeti fino a considerare tutta la stringa input
- ▶ Se si trova qualcosa di "inatteso" allora REJECT;
altrimenti ACCEPT

Esempio 2

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Idea cont:

- ▶ ripeti fino a considerare tutta la stringa input
- ▶ Se si trova qualcosa di "inatteso" allora REJECT;
altrimenti ACCEPT

Vediamo ora come costruire MdT M_1 per L

Esempio 2

Stringa input w , M_1 deve:

Esempio 2

Stringa input w , M_1 deve:

1. memorizzare simbolo piu' a sinistra e cancellarlo (scrive x)

Esempio 2

Stringa input w , M_1 deve:

1. memorizzare simbolo piu' a sinistra e cancellarlo (scrive x)
2. avanzare su nastro fino a superare $\#$, se non si trova allora REJECT

Esempio 2

Stringa input w , M_1 deve:

1. memorizzare simbolo piu' a sinistra e cancellarlo (scrive x)
2. avanzare su nastro fino a superare $\#$, se non si trova allora REJECT
3. Confronta il primo simbolo $\neq x$ con il simbolo memorizzato; se diverso REJECT

Esempio 2

Stringa input w , M_1 deve:

1. memorizzare simbolo piu' a sinistra e cancellarlo (scrive x)
2. avanzare su nastro fino a superare $\#$, se non si trova allora REJECT
3. Confronta il primo simbolo $\neq x$ con il simbolo memorizzato; se diverso REJECT
4. Cancella il simbolo confrontato e ritorna a inizio nastro

Esempio 2

Stringa input w , M_1 deve:

1. memorizzare simbolo piu' a sinistra e cancellarlo (scrive x)
2. avanzare su nastro fino a superare $\#$, se non si trova allora REJECT
3. Confronta il primo simbolo $\neq x$ con il simbolo memorizzato; se diverso REJECT
4. Cancella il simbolo confrontato e ritorna a inizio nastro
5. Vai a 1

Esempio 2

Vediamo ora la funzione di transizione per M_2

Nota: $\Sigma = \{0, 1, \#\}$ $\Gamma = \{0, 1, \#, x, _\}$

Nella descrizione, lo stato q_{reject} e tutte le sue transizioni in ingresso sono state omesse.

Ovunque vi sia una transizione mancante, va in q_{reject}

Esempio 2

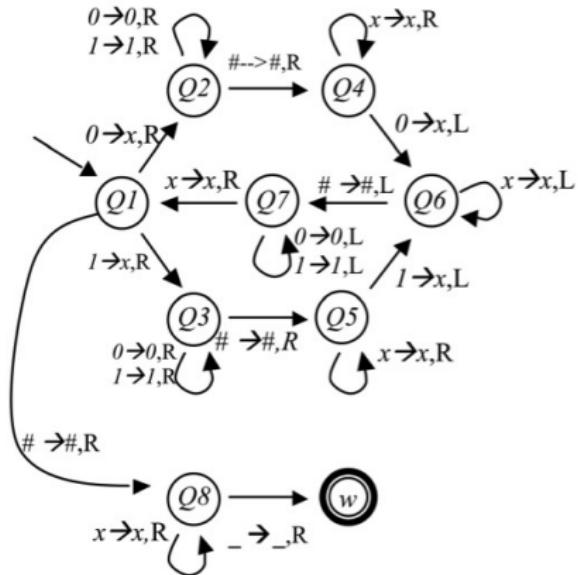
Nota.

gli stati q_2 e q_3 memorizzano il bit 0, mentre

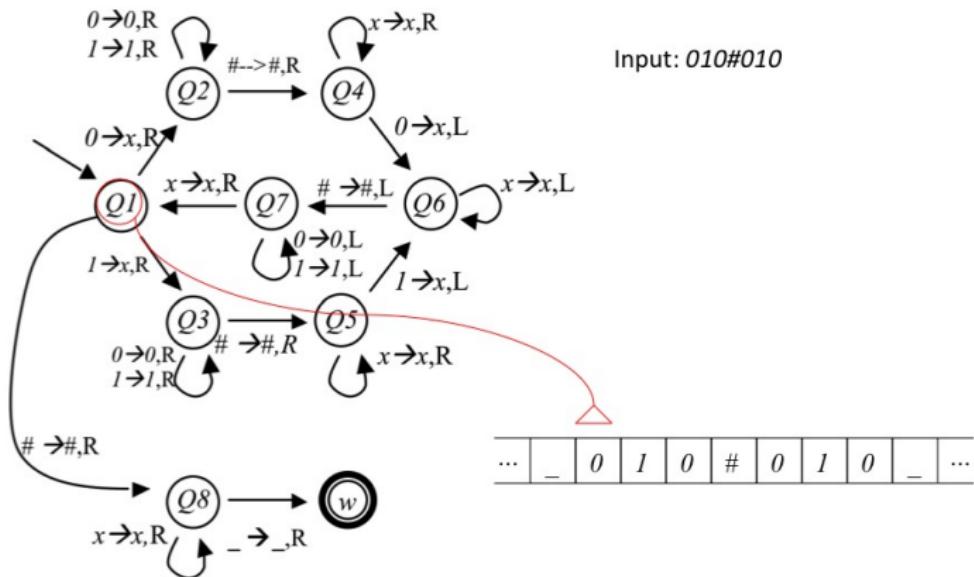
gli stati q_4 e q_5 memorizzano il bit 1

In altre parole: questi due segmenti sono identici, ogni segmento si utilizza il valore memorizzato.

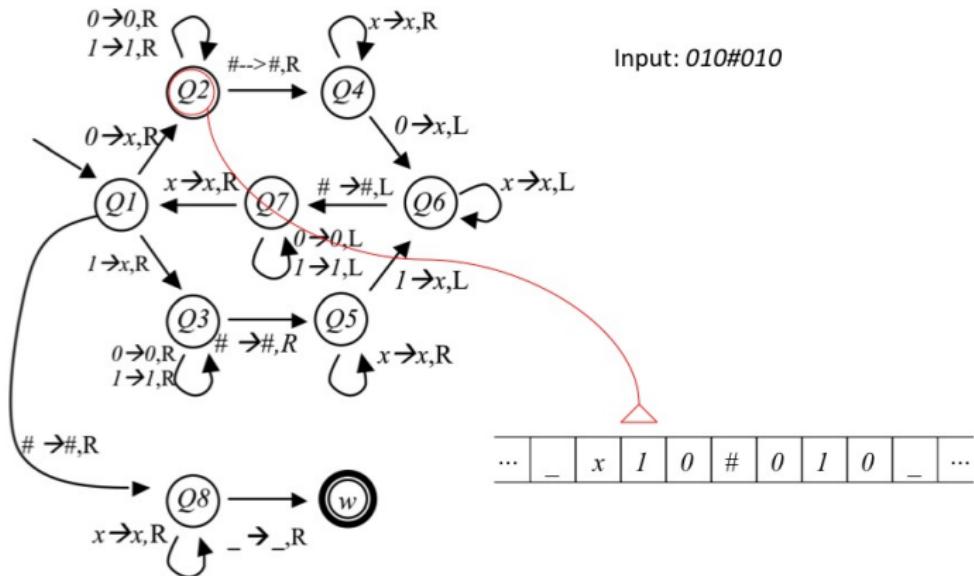
M_1 : $L = \{w\#w \mid w \in \{0,1\}^*\}$, $\Sigma = \{0,1,\#\}$ $\Gamma = \{0,1,\#,x,_\}$



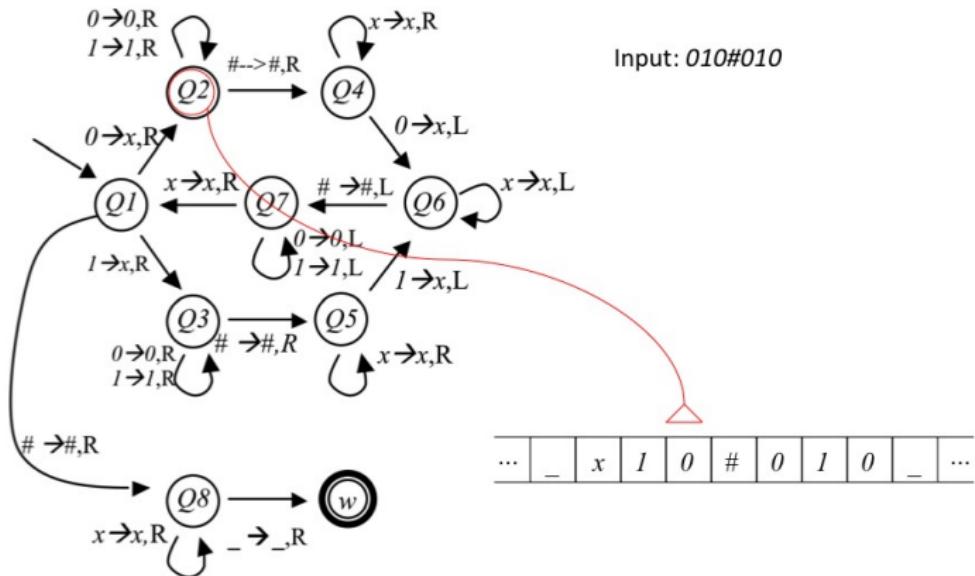
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



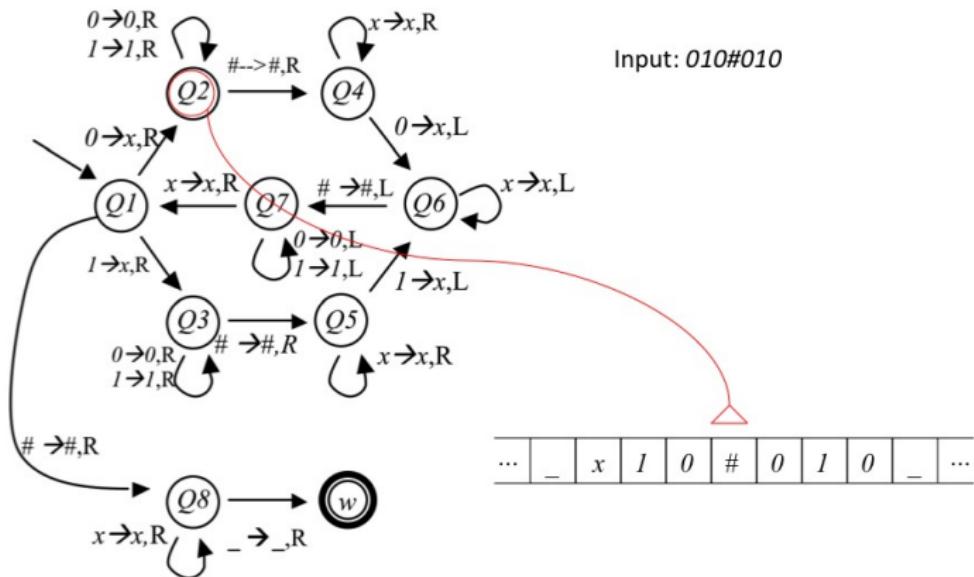
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



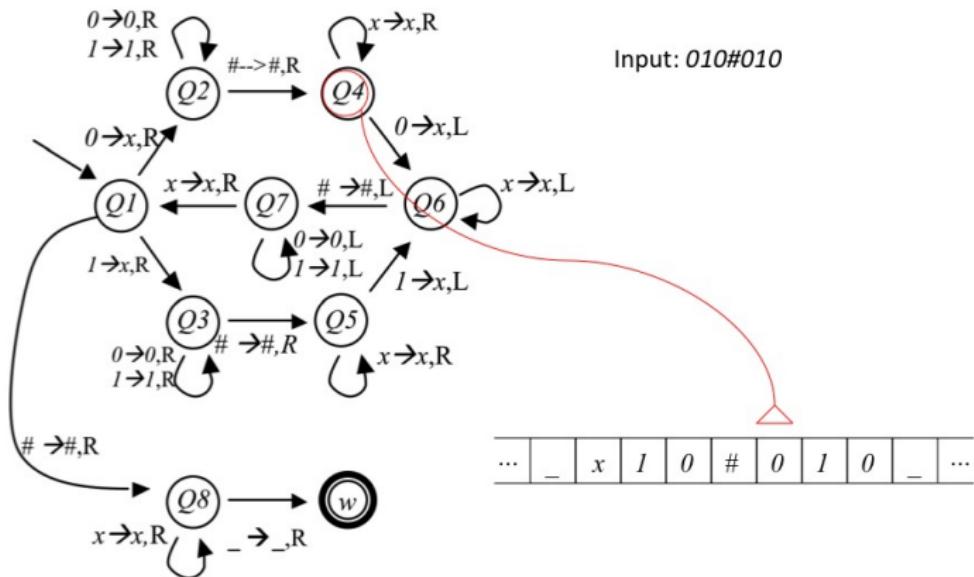
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



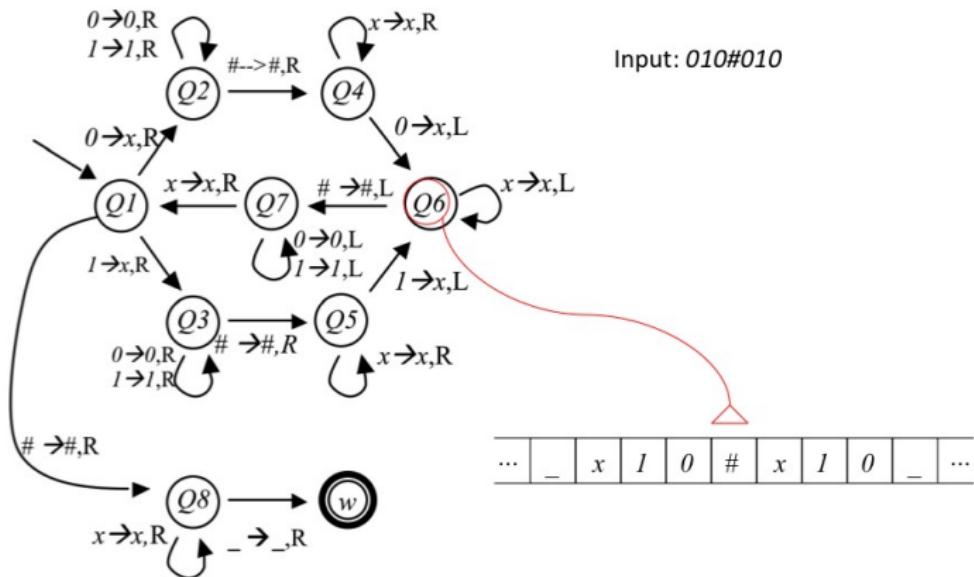
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



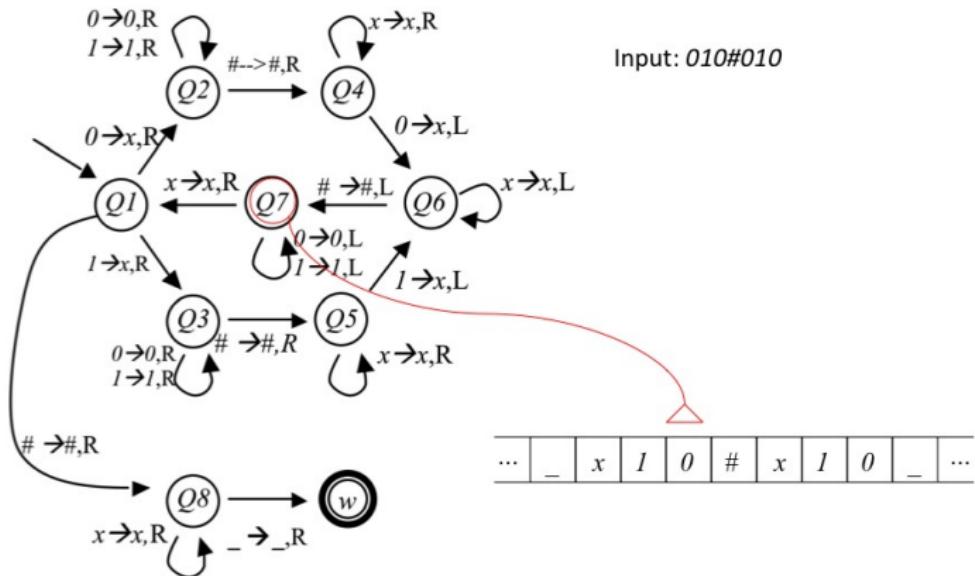
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



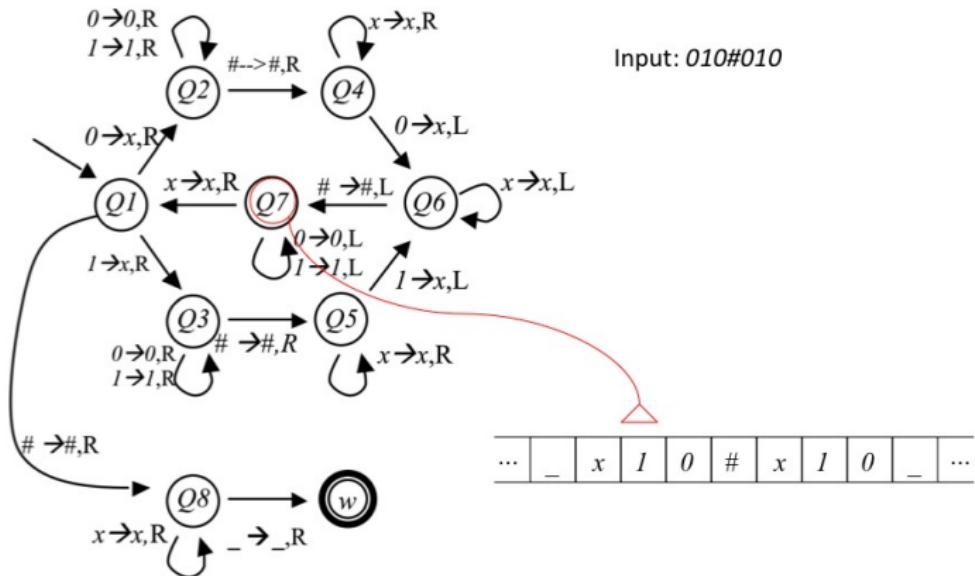
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



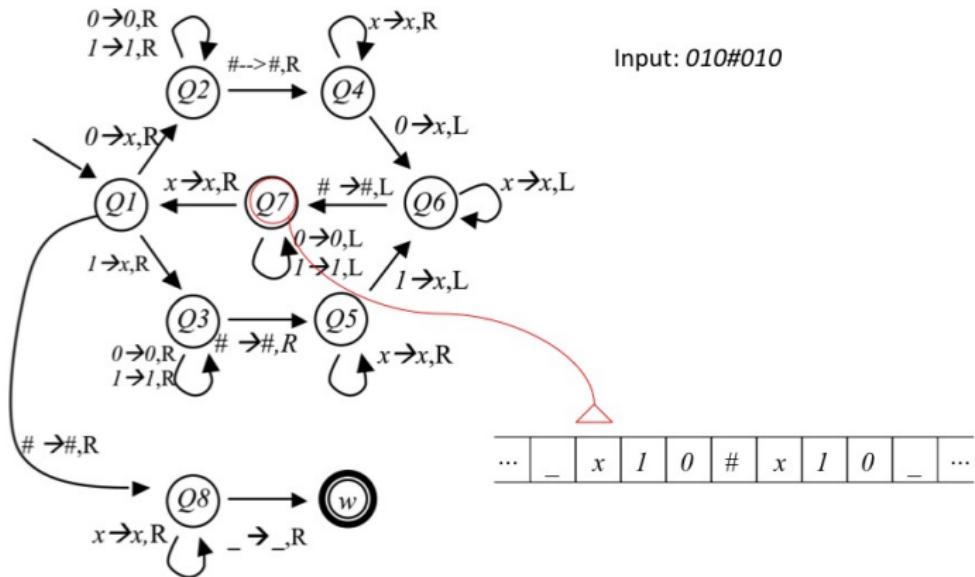
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



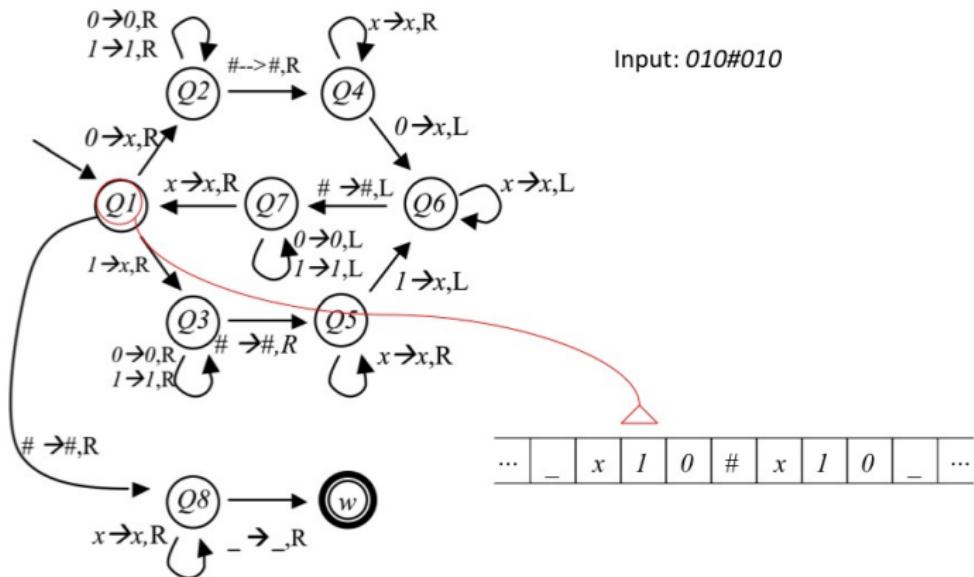
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



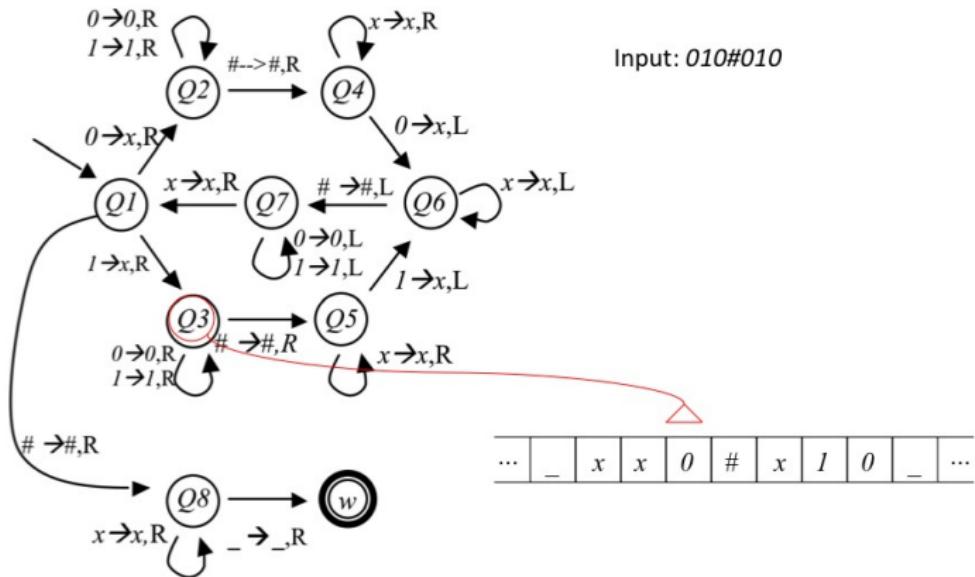
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



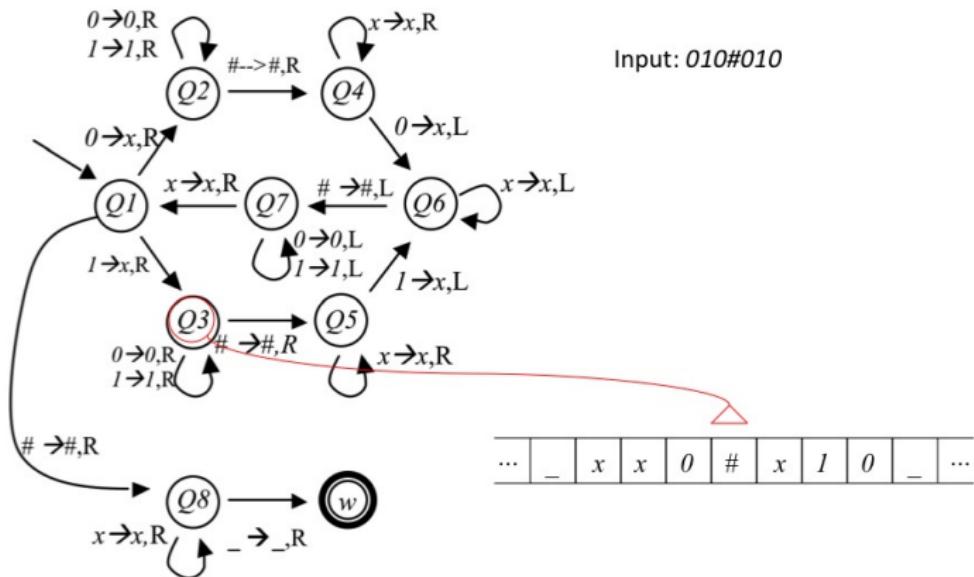
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



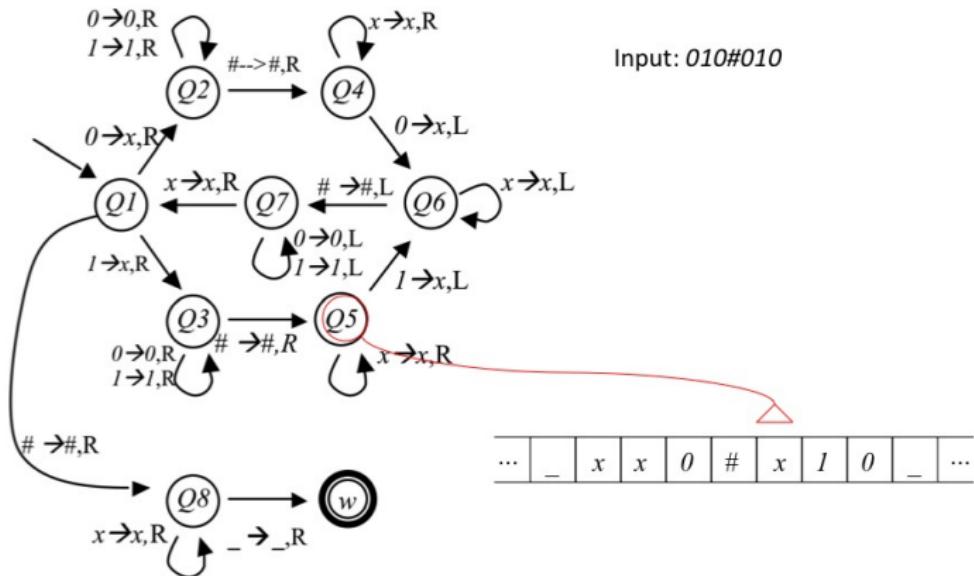
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



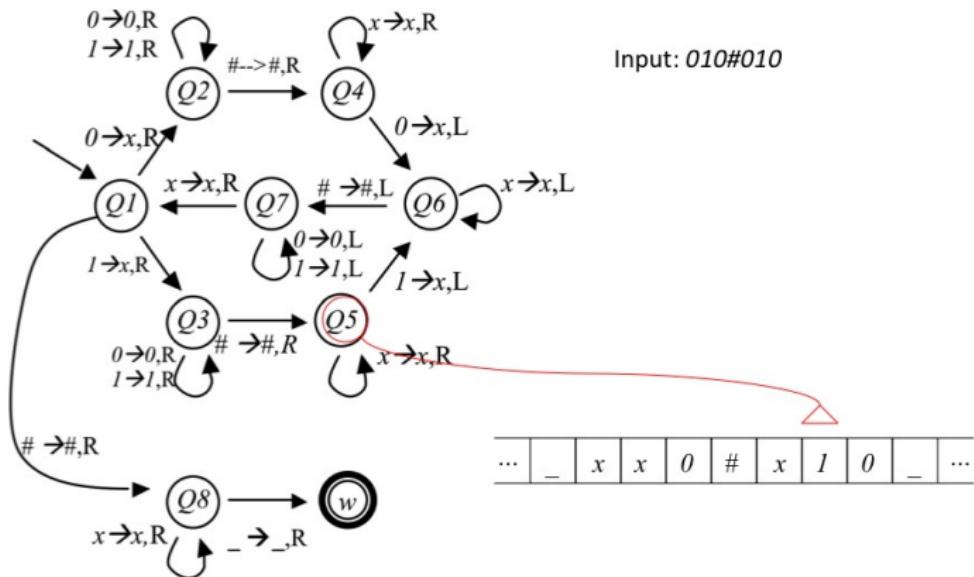
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



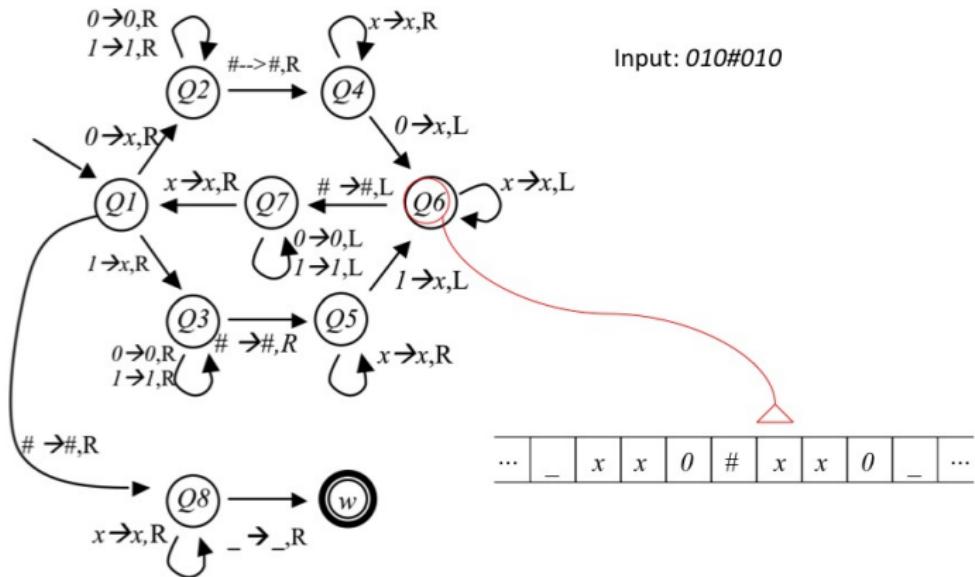
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



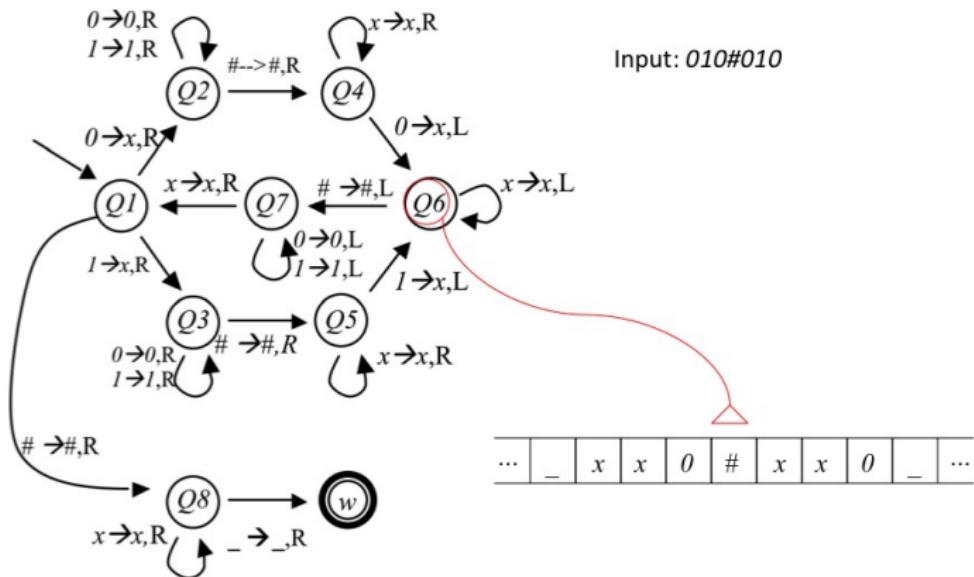
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



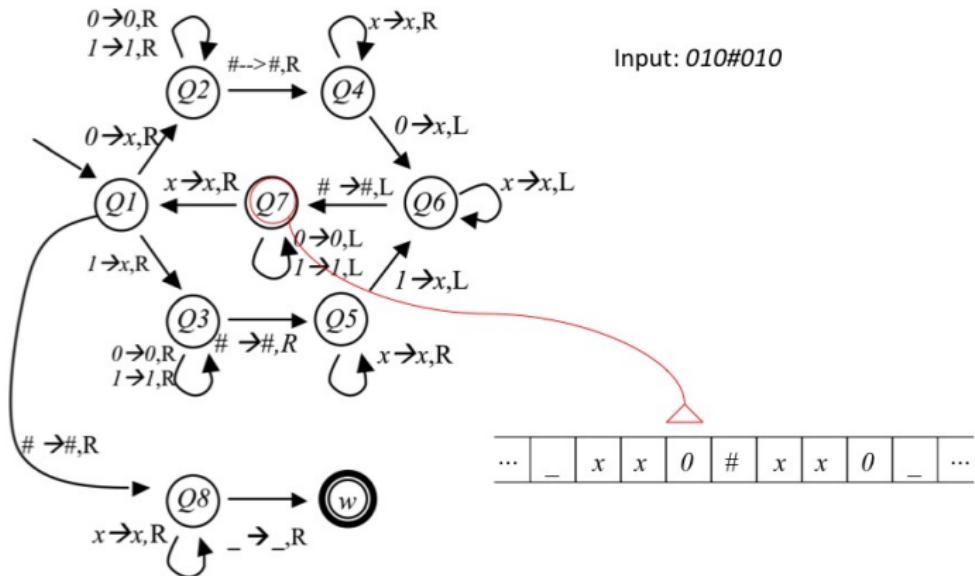
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



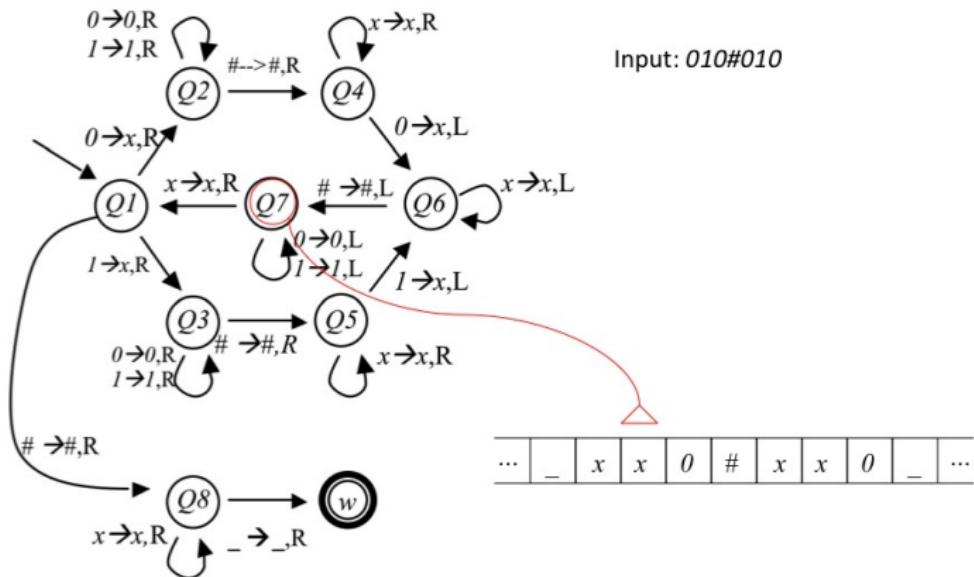
$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$



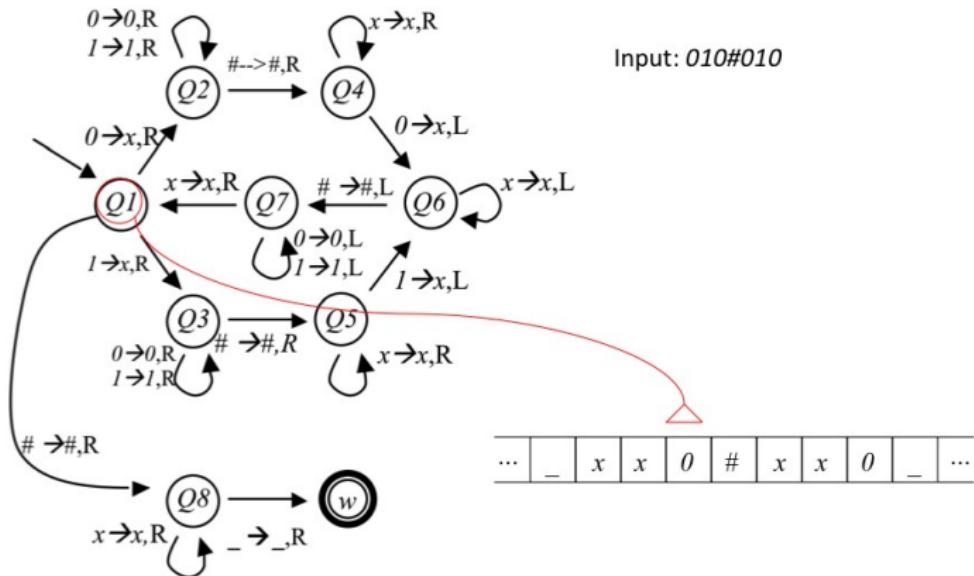
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



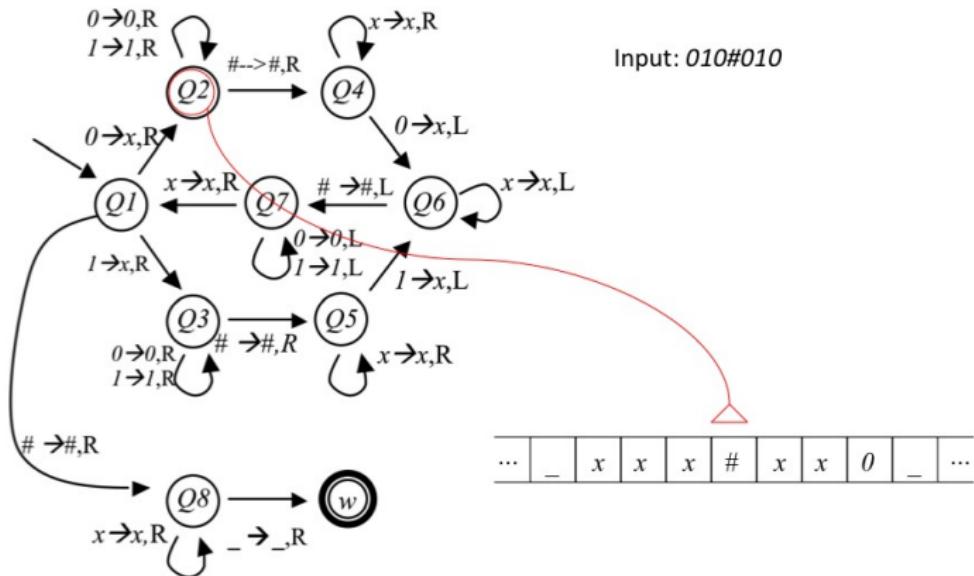
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



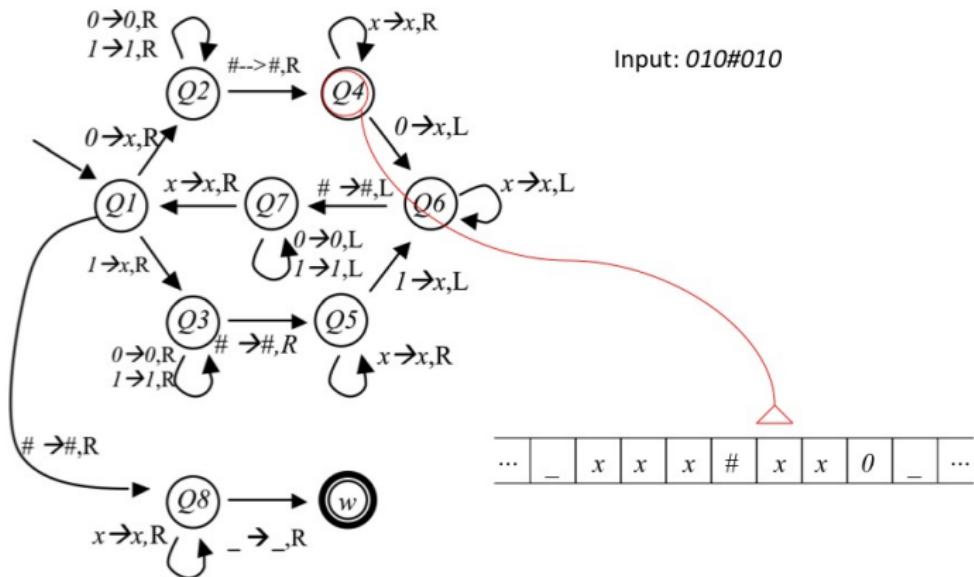
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



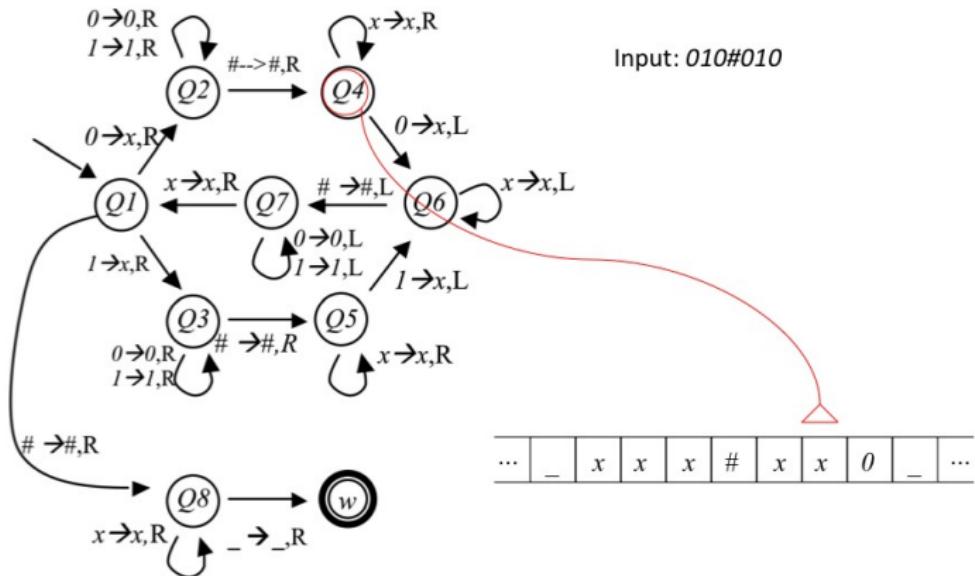
$M_1: L = \{w\#w \mid w \in \{0, 1\}^*\}$



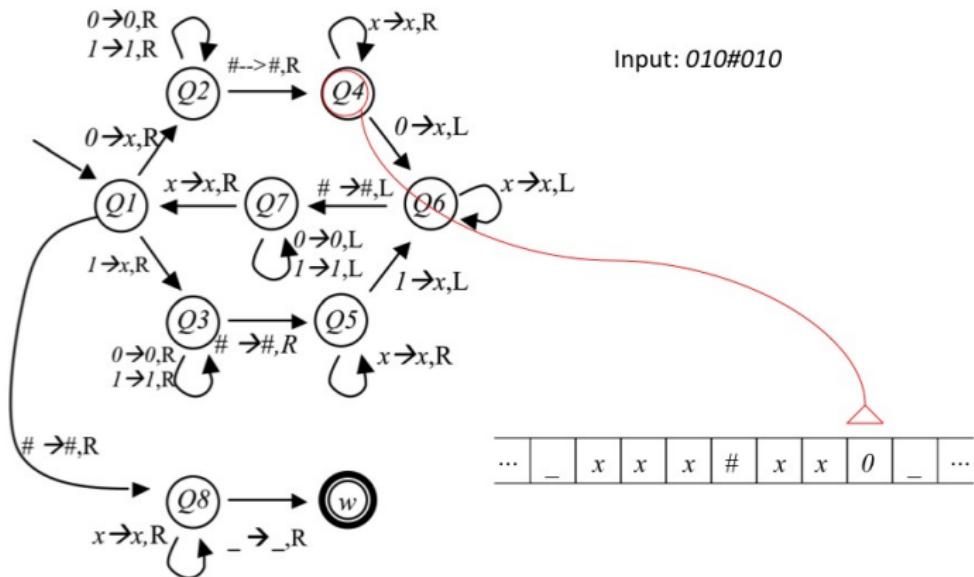
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



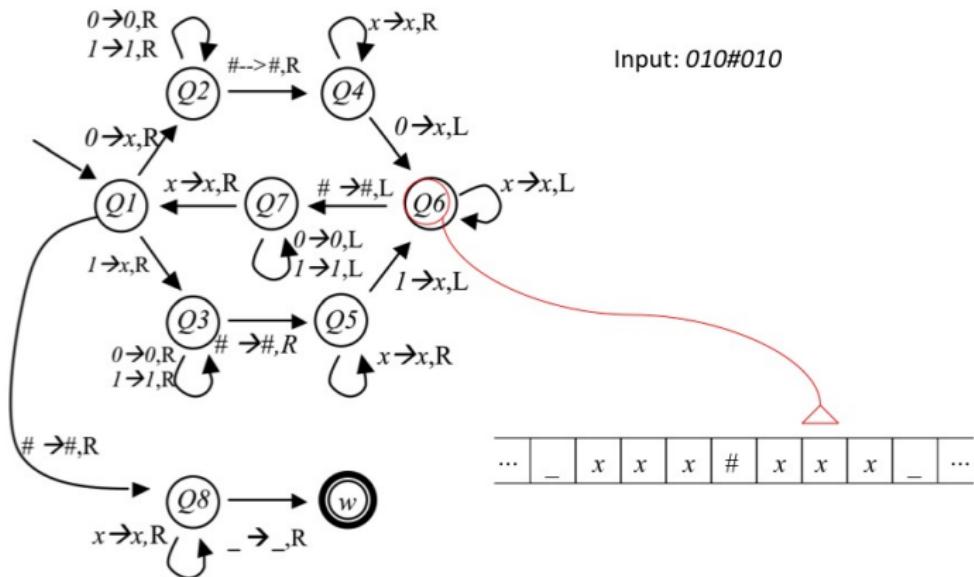
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



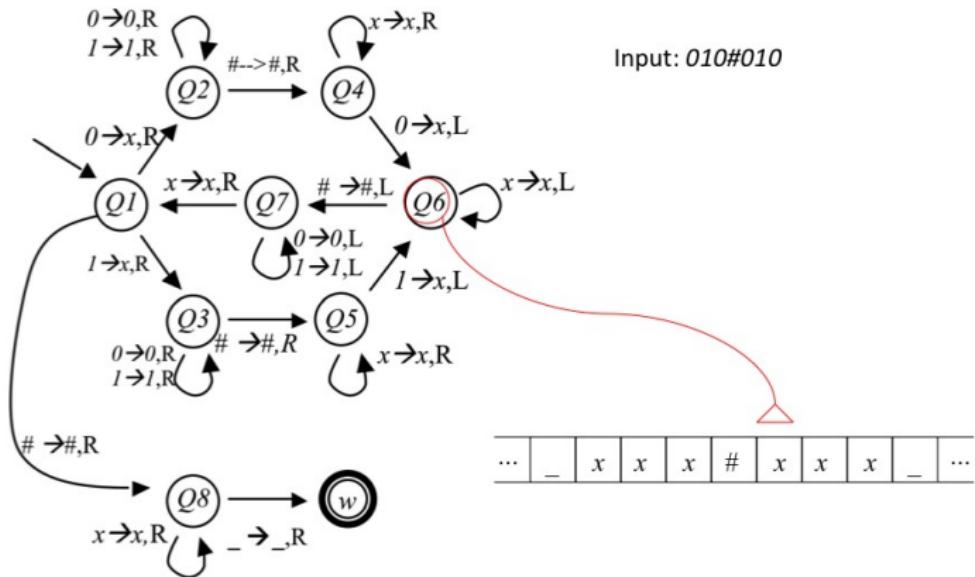
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



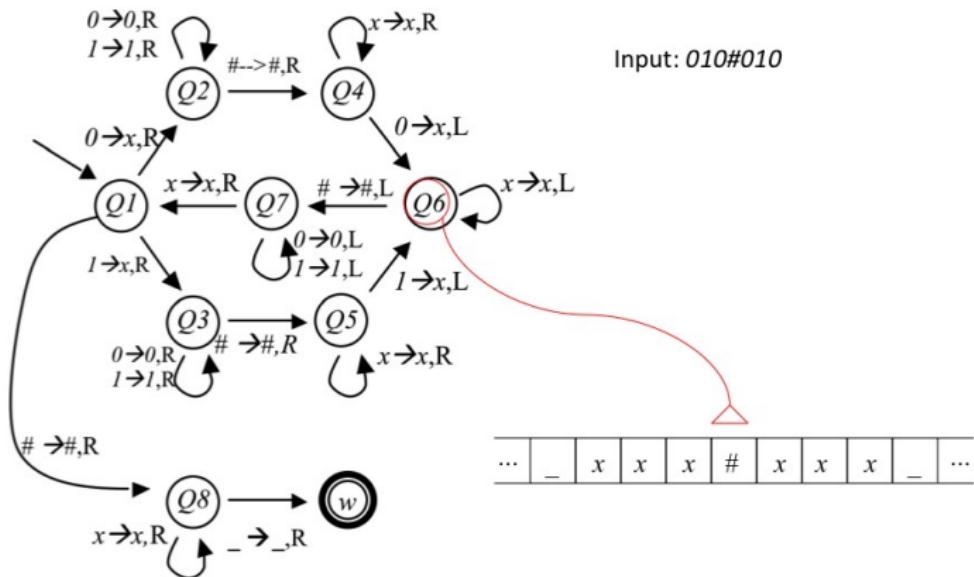
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



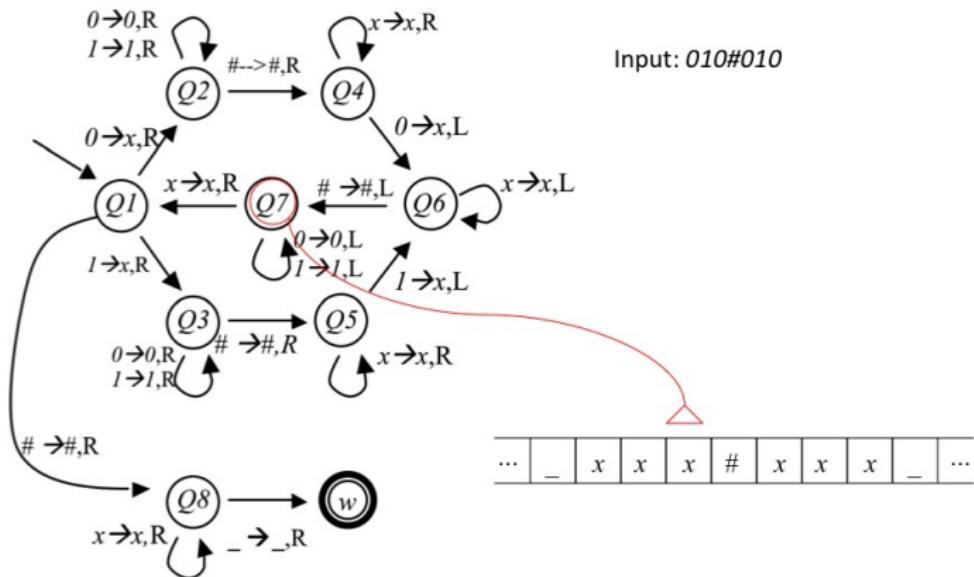
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



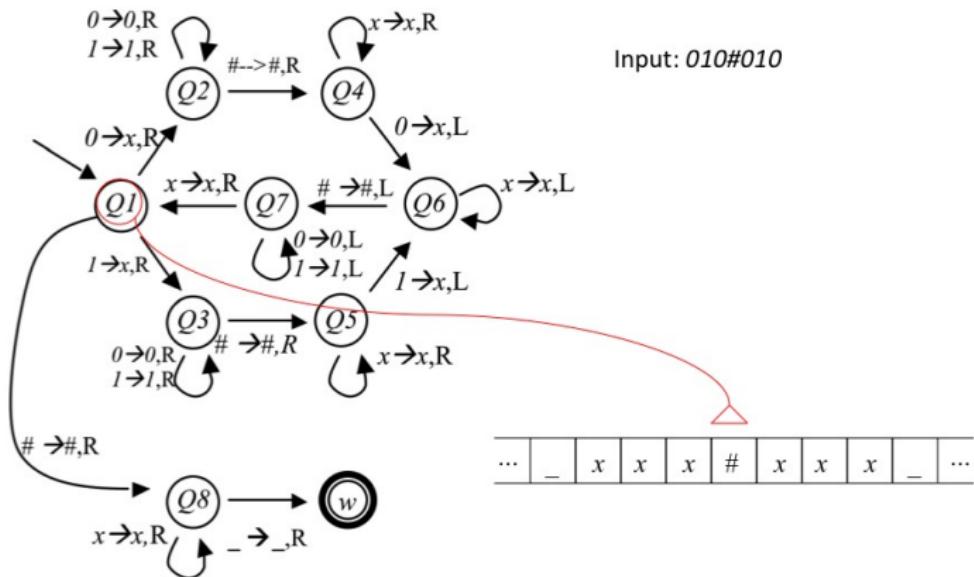
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



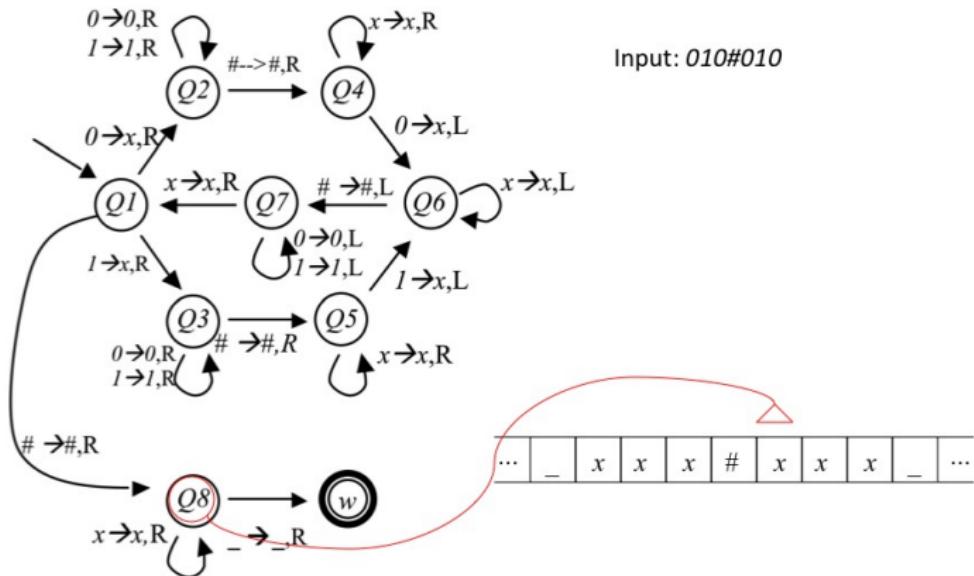
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



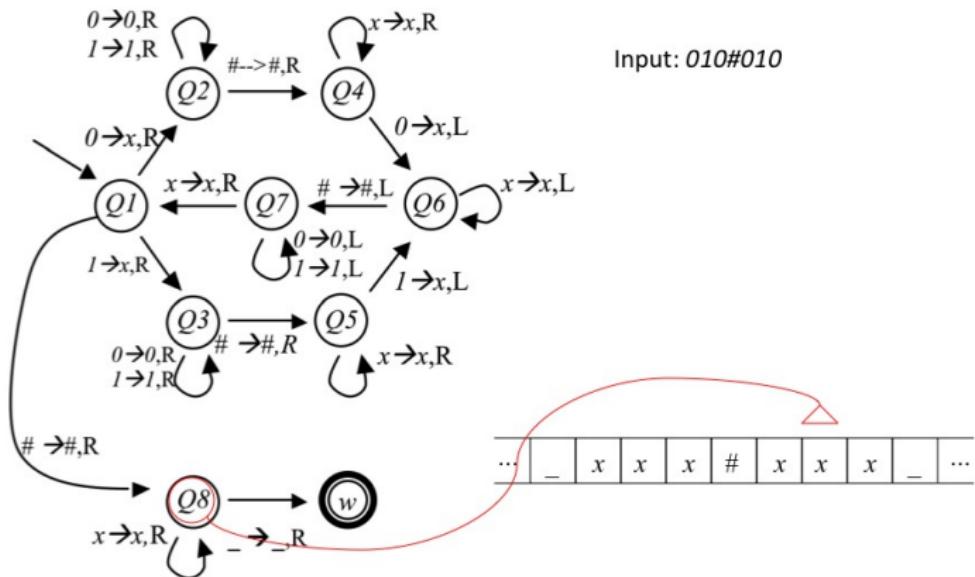
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



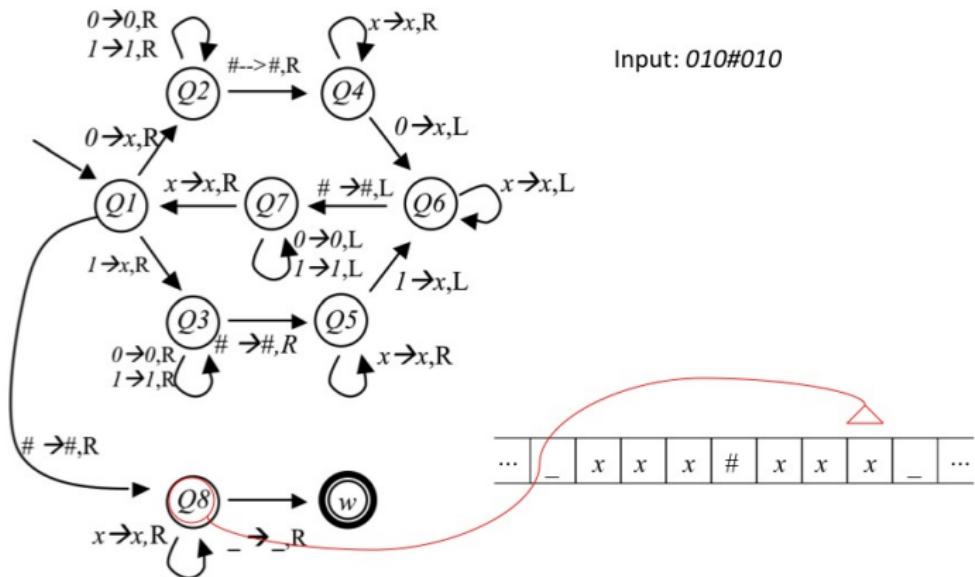
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



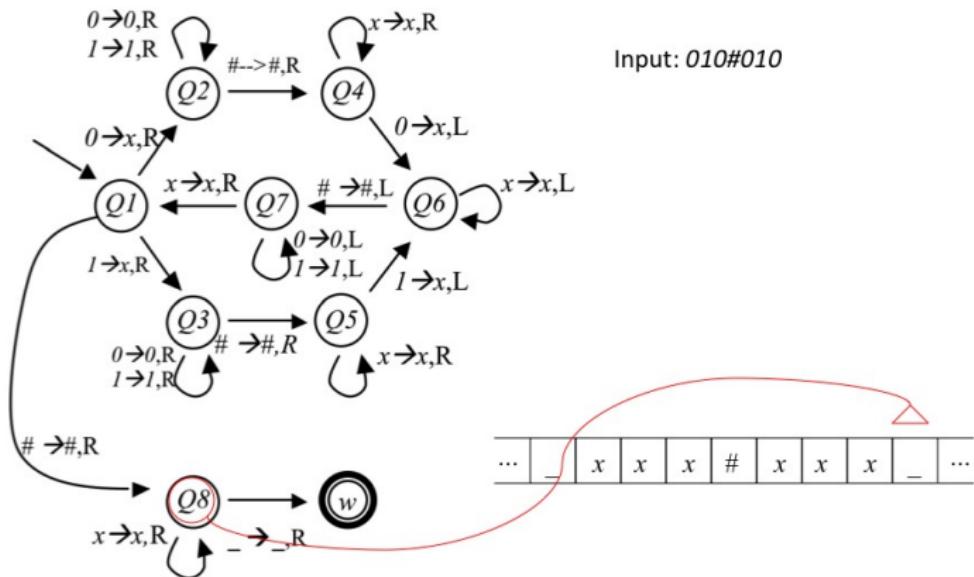
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



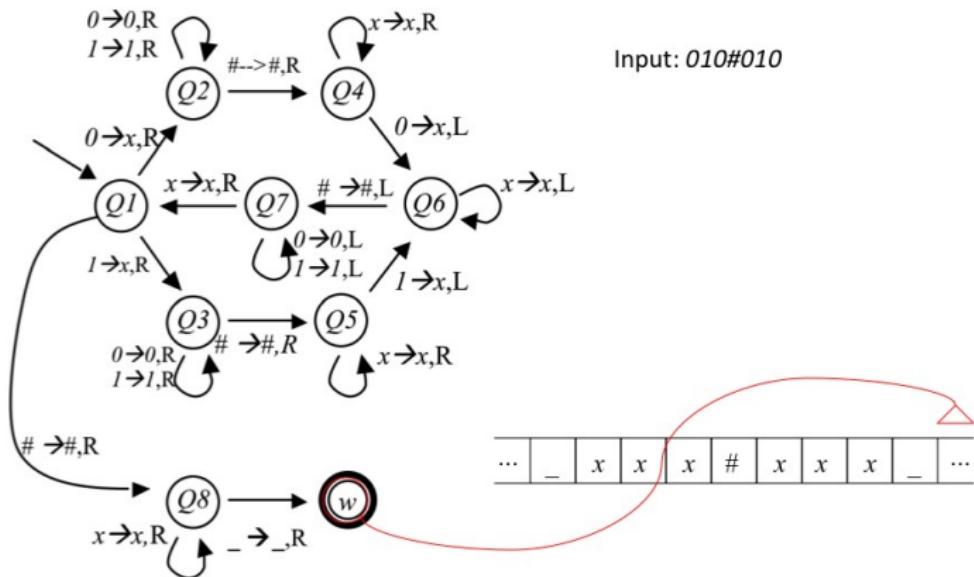
$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



$$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$$



$M_1: L = \{w\#w \mid w \in \{0,1\}^*\}$



Rappresentazione tabellare di una MdT

- ▶ 5 colonne e 1 riga per ogni etichetta
 - ▶ Prima colonna: stato attuale
 - ▶ Seconda colonna: simbolo letto
 - ▶ Terza colonna: nuovo stato
 - ▶ Quarta colonna: simbolo da scrivere
 - ▶ Quinta colonna: movimento testina
- ▶ Esempio:

stato	simbolo	stato	simbolo	movimento
q0	0	q0	1	R
q0	1	q0	0	R
q0	-	q1	-	L
q1	0	q1	0	L
q1	1	q1	1	L
q1	-	q2	-	R

Rappresentazione tabellare di una MdT

- ▶ 5 colonne e 1 riga per ogni etichetta
 - ▶ Prima colonna: stato attuale
 - ▶ Seconda colonna: simbolo letto
 - ▶ Terza colonna: nuovo stato
 - ▶ Quarta colonna: simbolo da scrivere
 - ▶ Quinta colonna: movimento testina
- ▶ Esempio:

stato	simbolo	stato	simbolo	movimento
q0	0	q0	1	R
q0	1	q0	0	R
q0	-	q1	-	L
q1	0	q1	0	L
q1	1	q1	1	L
q1	-	q2	-	R

COSA FA?