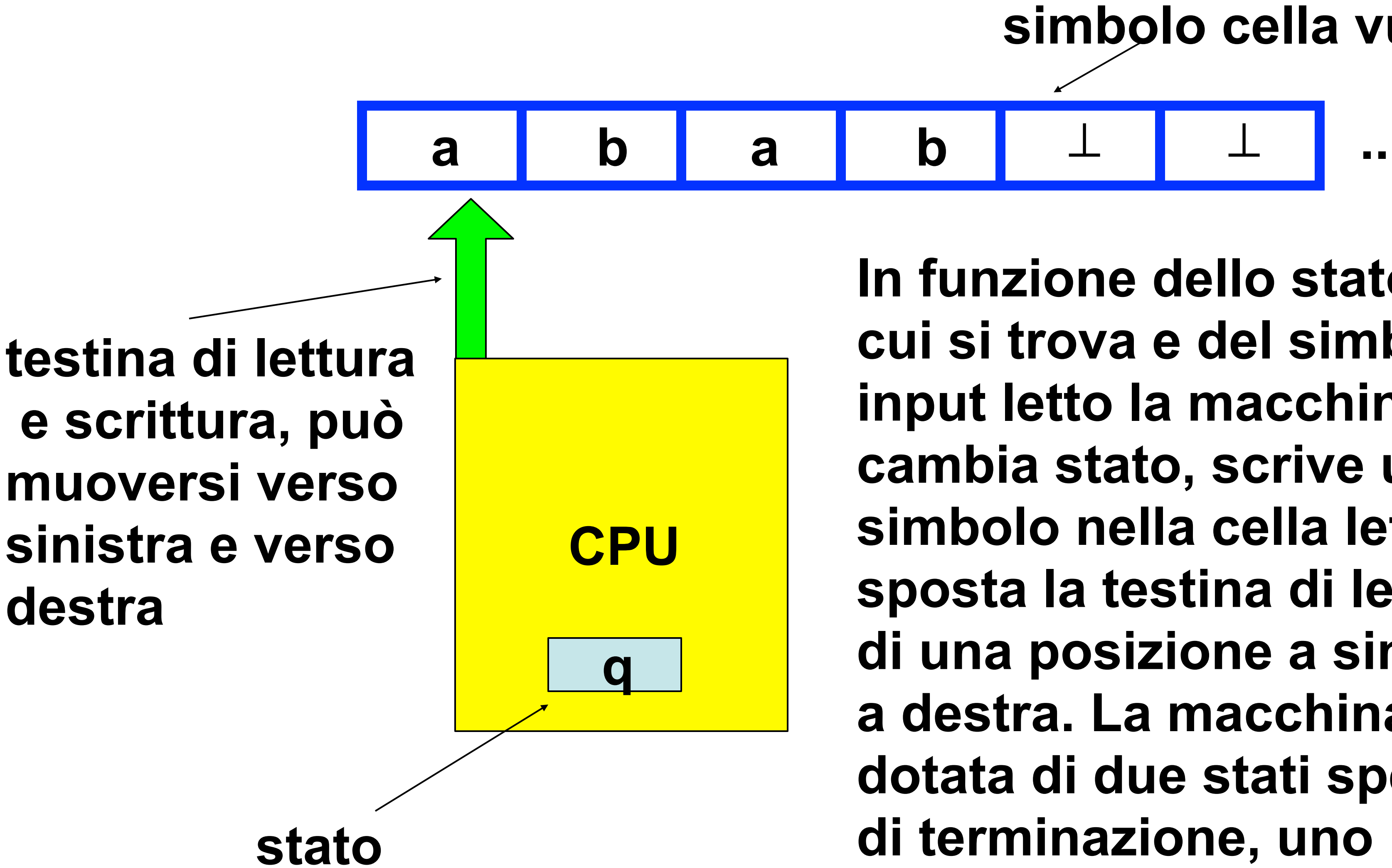
Sommario

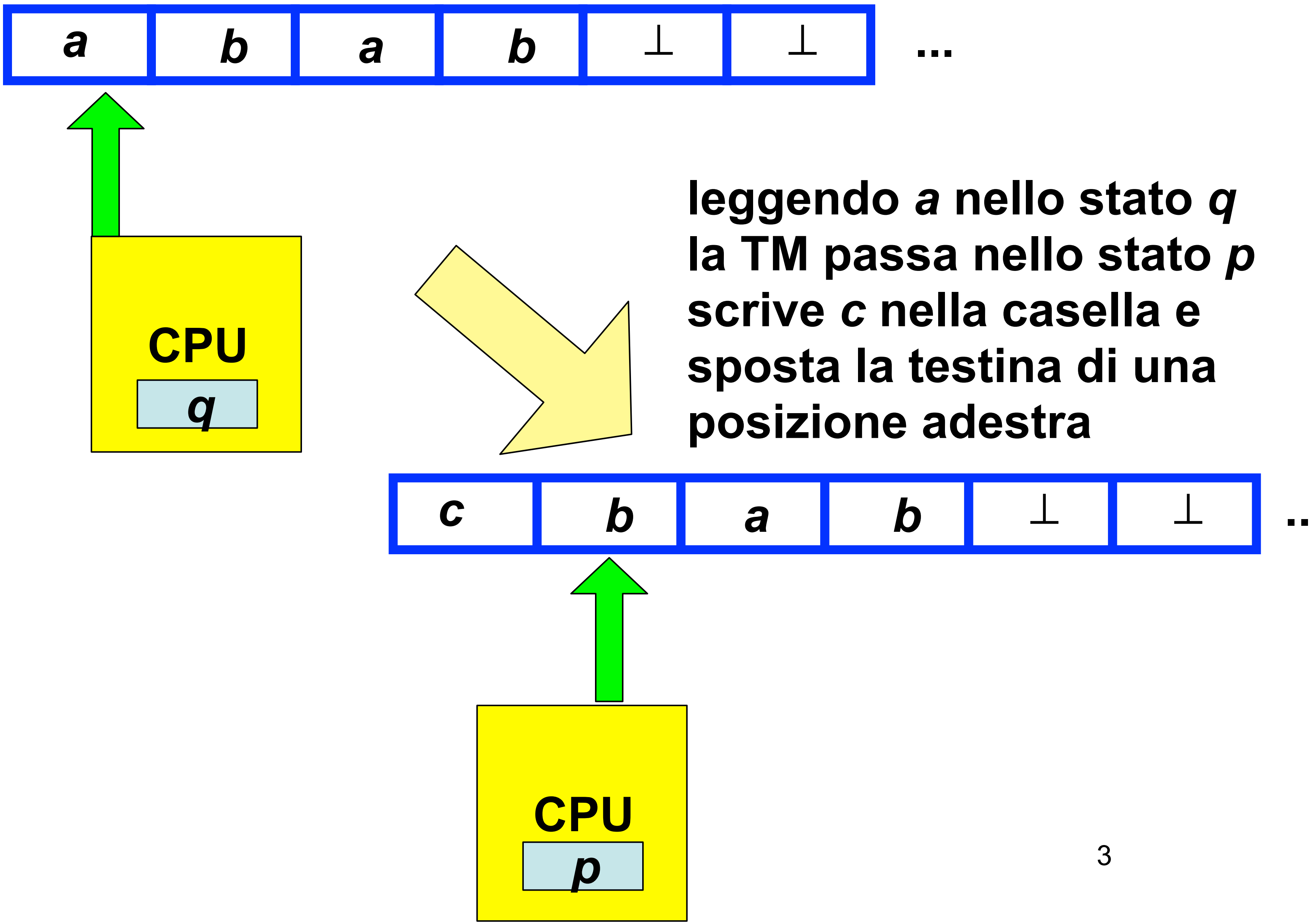
- Definizione Macchine di Turing, TM
- esempi
- Tesi di Church-Turing
- Proprietà elementari delle TM

Macchine di Turing: il modello mentale



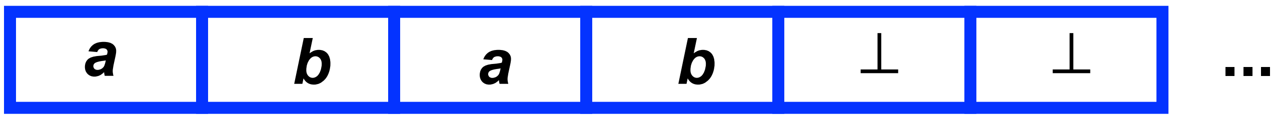
In funzione dello stato in cui si trova e del simbolo input letto la macchina cambia stato, scrive un simbolo nella cella letta, poi sposta la testina di lettura di una posizione a sinistra o a destra. La macchina è dotata di due stati speciali di terminazione, uno per l’accettazione e uno per il rifiuto.

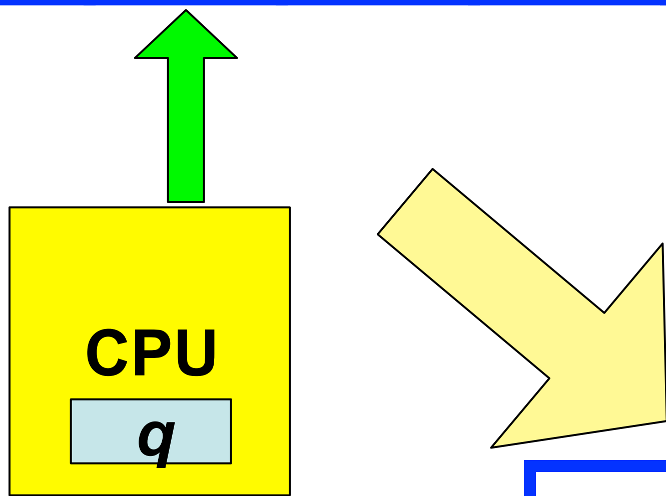
Macchine di Turing: una mossa a destra



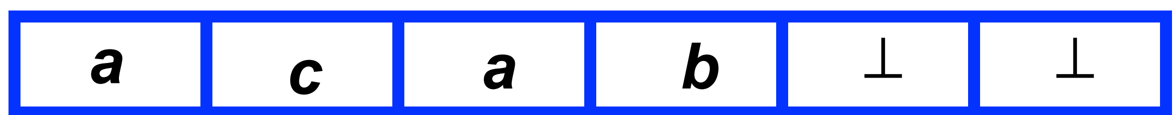
leggendo *a* nello stato *q* la TM passa nello stato *p* scrive *c* nella casella e sposta la testina di una posizione adestra

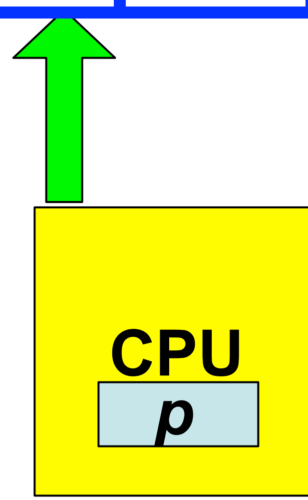
Macchine di Turing: una mossa a sinistra





leggendo *b* nello stato *q* la TM passa nello stato *p* scrive *c* nella casella e sposta la testina di una posizione a sinistra





La macchina termina se entra in uno stato speciale di accettazione o in uno stato speciale di rifiuto, dai quali non ci sono mosse da eseguire.

Esempio di TM 1

Una TM che accetta $L = \{0^n1^n \mid n > 0\}$

L'idea:

passo 1: se si legge uno 0 lo si sostituisce con X , se si legge 1 si rifiuta, se si legge Y si va al 4.

passo 2: si scorre il nastro verso destra, se si trova un 1 lo si sostituisce con Y, altrimenti si rifiuta

passo 3: si scorre il nastro verso sinistra fino al primo X si torna a destra e si ripete dal passo 1

passo 4: si scorre a destra, se si leggono solo Y e il simbolo di cella vuota, allora si accetta altrimenti si rifiuta.

000111

X00111

X00Y11

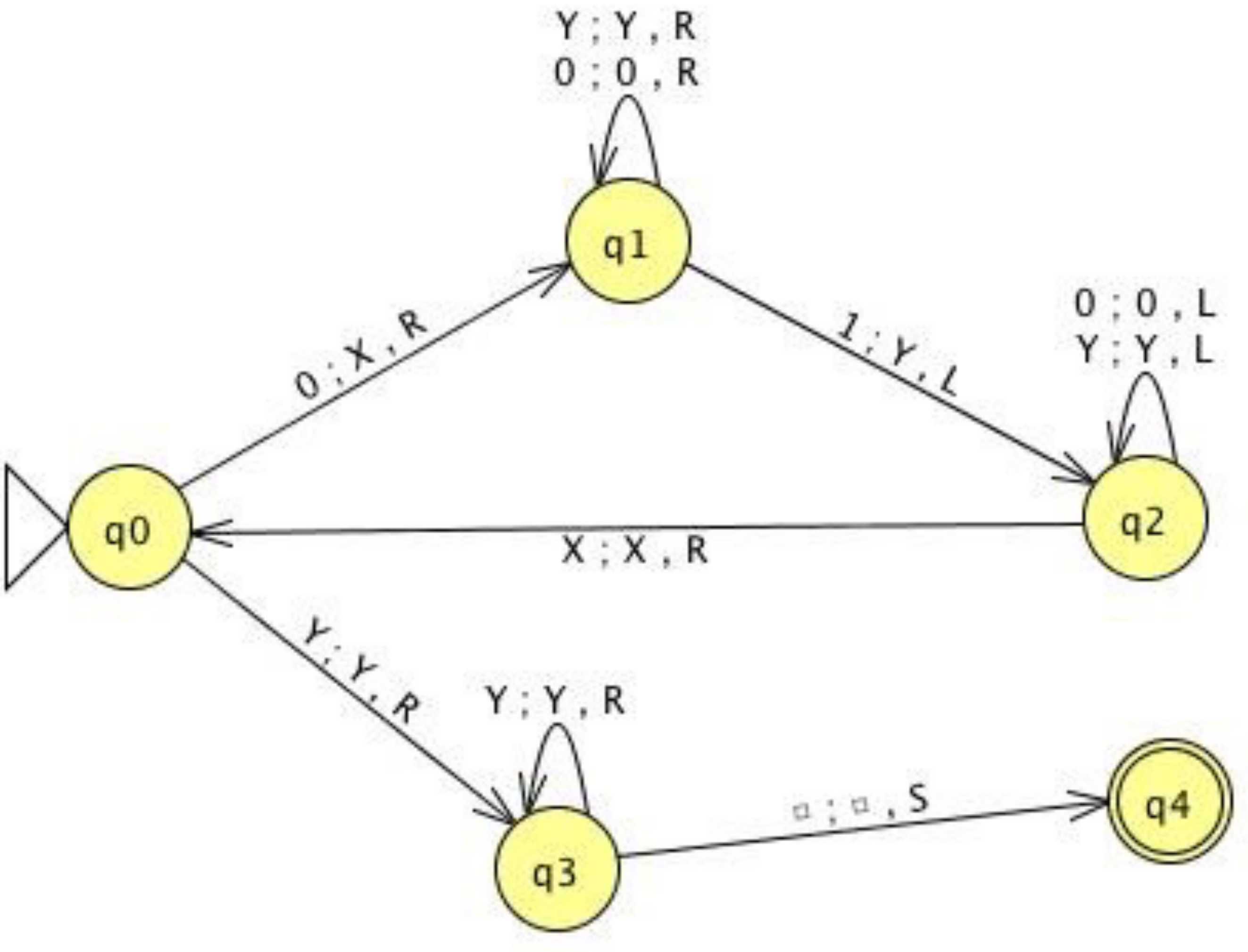
XX0Y11

XX0YY1

XXXYY1

XXXYYY

La TM nel dettaglio



Ogni mossa non rappresentata porta a uno stato di rifiuto

Esempio di TM 2

La TM deve decidere $L = \{0^{2^n}\ n \geq 0\}$.

L' idea si basa sull'osservazione che un numero è una potenza di 2 sse diviso per 2 dà ancora una potenza di due fino a ottenere 1.

passo 1: si marca il primo 0 e si scorre il nastro verso destra rimpiazzando uno 0 sì e uno no con X, facendo in modo che per ogni 0 che diventa X, ce ne sia uno che resta (così il numero è diviso a metà) e trascurando eventuali X incontrate, poi si ritorna all'inizio del nastro

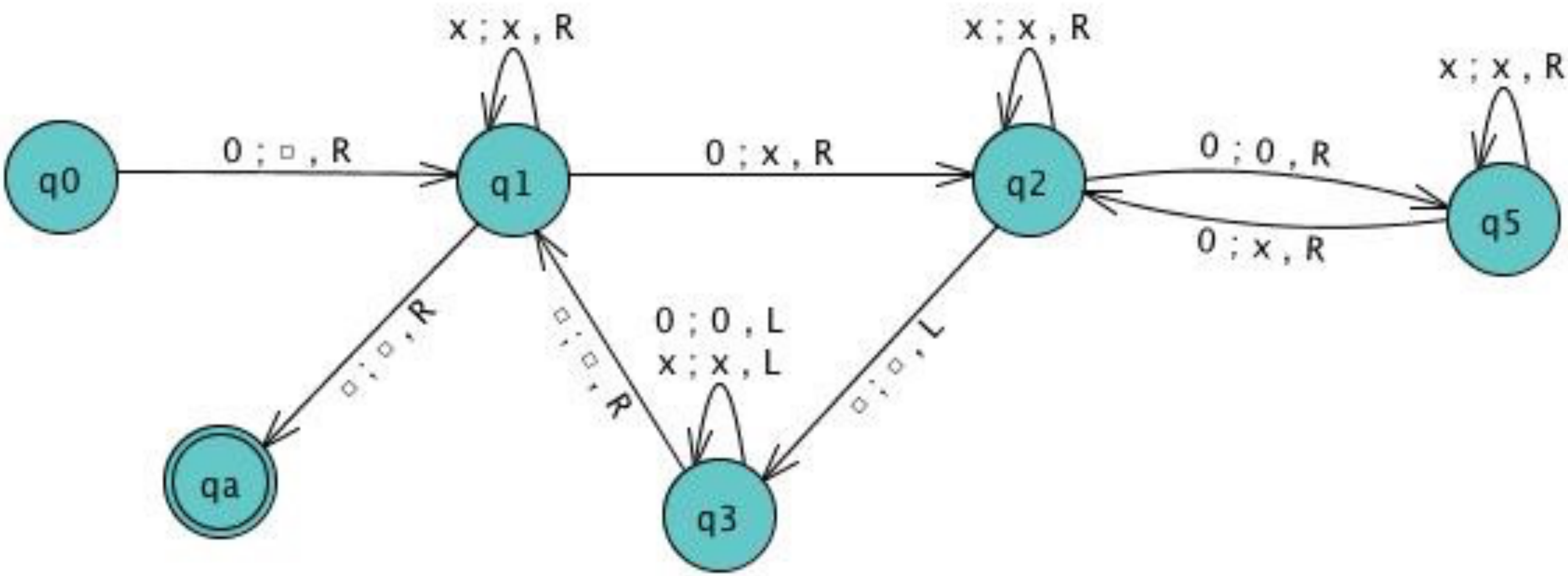
$\underline{0}0000000$

$\underline{0}X0X0X0X$

$\underline{0}XXX0XXX$

$\underline{0}XXXXXXX$

passo 2: se, dopo l'esecuzione una o più volte del passo 1 o fin dall'inizio scorrendo a destra si incontra un solo 0, allora si accetta.



Qui lo 0 “marcato” all’inizio è rimpiazzato con un blank. Ogni mossa non rappresentata porta a uno stato di rifiuto

Esempio di TM 3

La TM deve decidere L = {w#w | w è in {0,1}*e |w|>0}.

passo 1: si marca come cella vuota il primo 0 o il primo 1, i passi successivi sono divisi in due casi, a seconda se si è letto uno 0 o un 1.

<u>caso 0</u>

passo 2: si scorre il nastro a destra fino a trovare #

passo 3: si scorre ancora a destra, trascurando eventuali simboli già marcati, se si trova uno 0 lo si marca a X, altrimenti si rifiuta

passo 4: si torna all'inizio del nastro,

passo 5: si ritorna verso destra, trascurando eventuali simboli già marcati, e se si trova uno 0 lo si marca come X e si torna al passo 2,se si trova un 1 si va la passo 2 del caso 1, altrimenti si controlla se anche dopo l'occorrenza di # ci sono solo X e in tal caso si accetta.

<u>caso 1</u>

identico, scambiando 0 in 1

010#010

B10#010

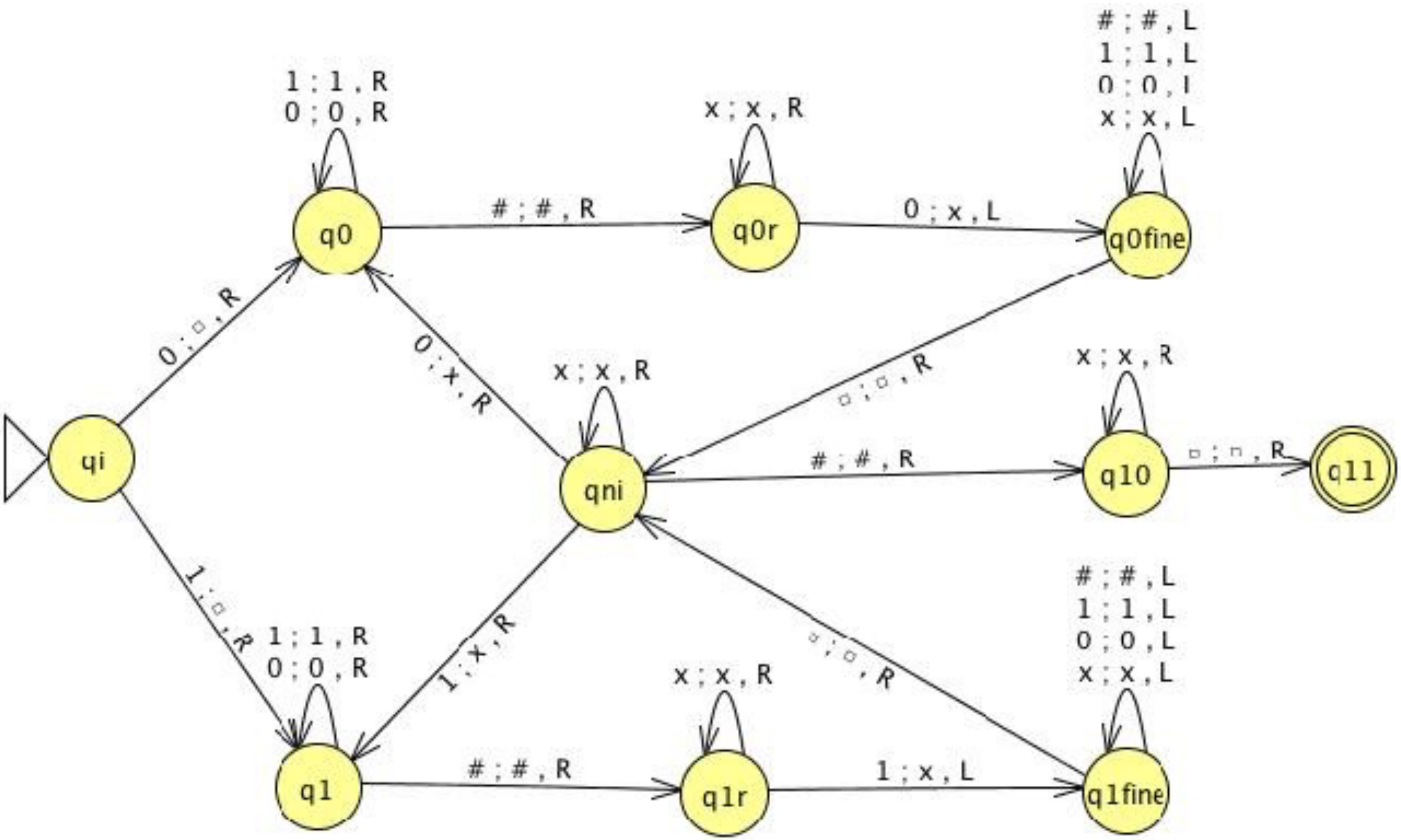
B10#X10

BX0#X10

BX0#XX0

BXX#XXX

La TM nel dettaglio



Ogni mossa non rappresentata porta a uno stato di rifiuto

Modello formale

Una **Machina di Turing** (**T**uring **M**achine) **deterministica**, è una settupla $M = (Q,\Sigma, \Gamma,\delta,q_0, q_a , q_r)$ dove

- Q, Σ e $\Gamma \supset \Sigma$ sono insiemi finiti, rispettivamente degli stati, dei simboli di input e dell'alfabeto di nastro; lo speciale simbolo di cella vuota $\perp$ è in Γ, ma <u>non</u> in Σ

- $\delta : Q - \{q_a,q_r\} \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ è la funzione di transizione;

- q_0 è lo stato iniziale;

- q_a e q_r sono rispettivamente lo stato di accettazione e quello di rifiuto, con $q_a \neq q_r$

Configurazioni

Una configurazione deve informare sul contenuto del nastro, lo stato della macchina e la posizione della testina di lettura.

Queste informazioni si possono ottenere sinteticamente da una sequenza del tipo

$$\alpha q a \beta \text{ in } \Gamma^*Q\Gamma^*$$

dove α e β sono stringhe su Γ, a è un simbolo di Γ, $\alpha a \beta$ è il contenuto del nastro, a è il simbolo in lettura e q è lo stato della macchina.

La sequenza

$$q_0 x$$

dove x è una stringa input, e q_0 è lo stato iniziale è detta configurazione iniziale.

Mosse

dove $\Rightarrow_T$
si legge
"porta a"

a destra:

se $\delta(q,a) = (p,b,R)$ allora

$$\alpha qac\beta \Rightarrow_T \alpha bpc\beta$$

a sinistra:

se $\delta(q,c) = (p,b,L)$ allora

$$\alpha aqc\beta \Rightarrow_T \alpha pab\beta$$

Una configurazione del tipo $\alpha q_a\beta$ è detta di accettazione.

Una configurazione del tipo $\alpha q_r\beta$ è detta di rifiuto.

Queste configurazioni sono di terminazione

Linguaggio accettato

Sia $M = (Q,\Sigma, \Gamma,\delta,q_0, q_a , q_r)$, e sia $C(x)$ l'insieme delle configurazioni raggiungibili da quella iniziale $c_0 = q_0\, x$.

Il linguaggio accettato (riconosciuto) è

$$L(M) = \{x \mid x \in \Sigma^* \text{ e } \exists\, c \in C(x) \text{ e } c \text{ è di accettazione}\}$$

$$= \{x \mid x \in \Sigma^* \text{ e } q_0 x \Rightarrow^* \alpha q_a \beta \text{ con } \alpha,\beta \in \Gamma^*\}$$

Il linguaggio rifiutato è

$$R(M) = \{x \mid x \in \Sigma^* \text{ e } \exists\, c \in C(x) \text{ e } c \text{ è di rifiuto}\}$$

$$= \{x \mid x \in \Sigma^* \text{ e } q_0 x \Rightarrow^* \alpha q_r \beta \text{ con } \alpha,\beta \in \Gamma^*\}$$

In generale $L(M) \cup R(M) \subseteq \Sigma^*$!
Se $L(M) \cup R(M) = \Sigma^*$ allora vuol dire che la TM si ferma sempre, in tal caso $L(M)$ è il linguaggio deciso dalla TM.

Linguaggio accettato

In generale

il linguaggio accettato, L(M), da una TM è detto Turing-riconoscibile o semplicemente riconoscibile (ricorsivamente enumerabile, semidecidibile, in questo caso $L(M) \cup R(M) \subseteq \Sigma^*$)

Se

la TM si ferma sempre L(M) è il linguaggio deciso dalla TM ed è detto Turing- decidibile o semplicemente decidibile (in questo caso $L(M) \cup R(M) = \Sigma^*$)

Classe dei linguaggi accettati

L'insieme di tutti i linguaggi che sono riconosciuti da una TM è così definito:

$$\mathcal{L}(TM) = \{L \mid \exists\ M \in TM \text{ e } L(M) = L\}$$

Tesi di Church - Turing

Tesi di Church - Turing:

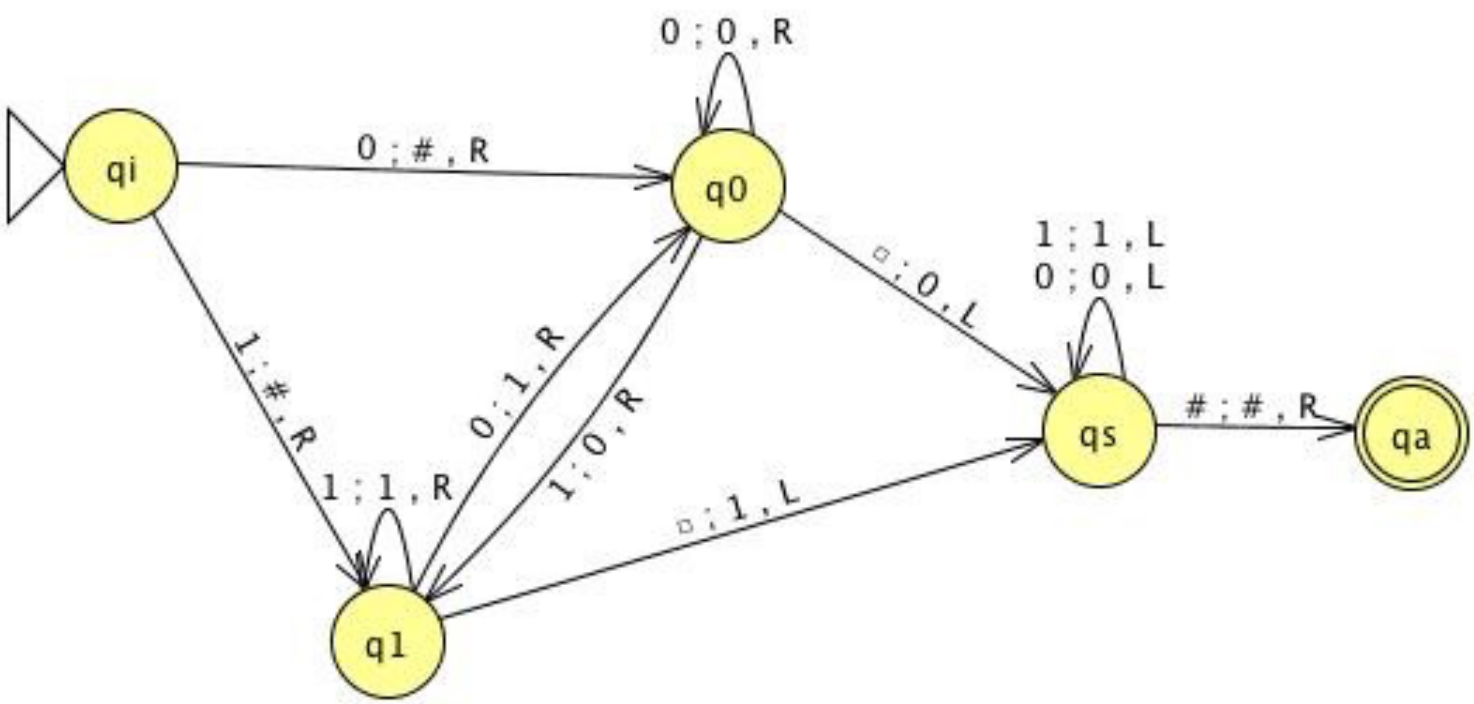
La classe delle funzioni calcolate da una TM è la classe delle funzioni che informalmente sono considerate <u>effettivamente</u> calcolabili, o equivalentemente la classe dei linguaggi riconosciuti da una TM corrisponde alla classe dei problemi intuitivamente <u>effettivamente</u> risolvibili.

Quindi possiamo identificare l'idea di algoritmo con quella di una TM che si ferma sempre.

Mentre un semialgoritmo corrisponde a una TM che non sempre si ferma.

Esempio di modulo TM

Una TM che trasforma una stringa binaria x nella stringa #x, riposizionando la testina di lettura sulla prima cella di nastro



Esempio: Una TM che genera la successiva stringa binaria in ordine canonico

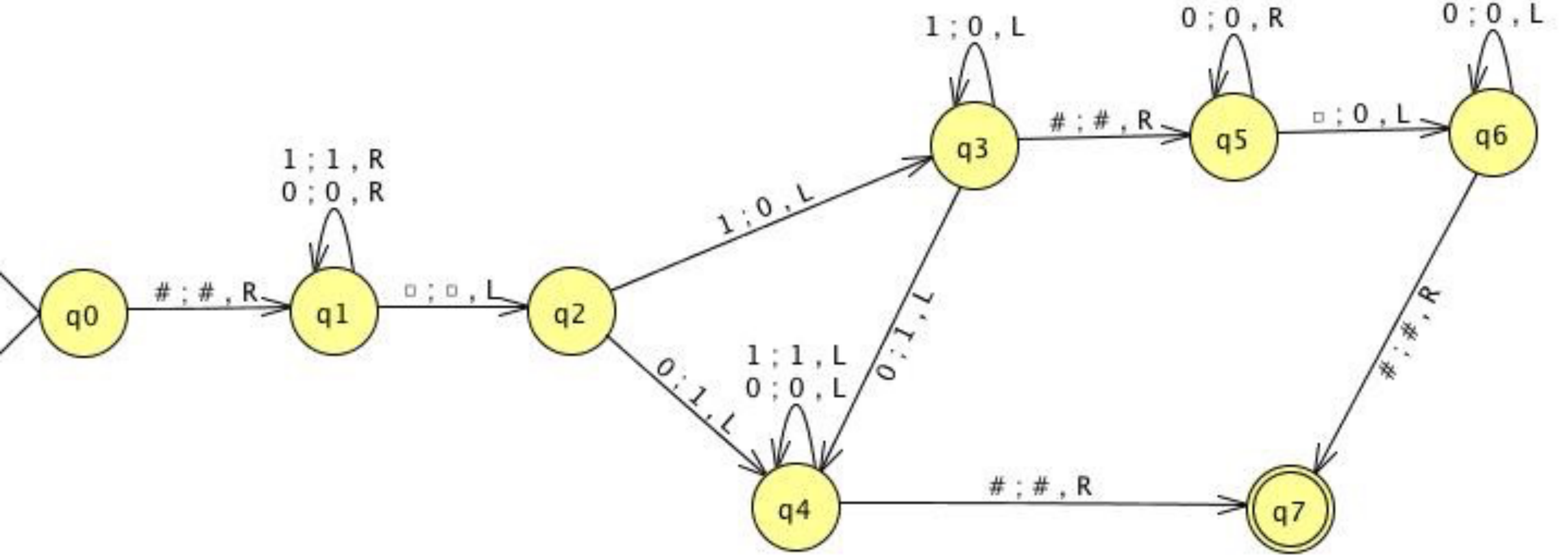
Le stringhe binarie elencate in ordine canonico :

0 1 00 01 10 11 000 001 010 011 100 101 110 111 ...

1. **Data una sequenza x01…1, con x in {0,1}*, la successiva si ottiene trasformando tutti gli 1 finali in 0 e lo 0 in 1, ottenendo x10...0**
2. **Data una sequenza x0, con x in {0,1}*, la successiva si ottiene trasformando lo 0 in 1, ottenendo x1**
3. **data una sequenza 1...1 la successiva si ottiene trasformando ogni 1 in 0 e concatenando un altro 0**

Ne diamo una descrizione dettagliata come file JFLAP, supponendo di avere utilizzato il modulo di TM che ha inserito il marcatore di fine nastro nella prima cella.

Esempio: Una TM che genera la successiva stringa binaria in ordine canonico



Esempio di TM che non si ferma