

Lezione di Informatica Teorica: Macchine di Turing

Appunti da Trascrizione Automatica

30 giugno 2025

Indice

1 Introduzione alle Macchine di Turing

Ripassiamo brevemente i concetti fondamentali delle Macchine di Turing (MT) introdotti nella lezione precedente.

Definizione 1 (Macchina di Turing). *Una Macchina di Turing è un automa a stati finiti con le seguenti peculiarità:*

- Possiede un **nastro di input infinito** che può essere letto e scritto. A differenza degli automi a stati finiti che possono solo leggere l'input, le MT possono modificare il loro nastro.
- La **testina** della macchina può spostarsi sia a destra che a sinistra sul nastro. Gli automi a stati finiti, invece, si muovono solo a destra.
- Il **programma** (funzione di transizione) di una MT è fissato e non può cambiare durante l'esecuzione. Una volta che la macchina è "accesa" con un determinato programma, esso rimane invariato. Questo, pur sembrando una limitazione rispetto ai computer moderni, vedremo che non lo è, poiché le MT sono modelli di calcolo universali.

Queste caratteristiche rendono le Macchine di Turing un modello di calcolo universale, capace di simulare qualsiasi algoritmo computabile.

1.1 Linguaggi: Decidibilità vs. Accettazione

È fondamentale distinguere due concetti chiave relativi ai linguaggi che una Macchina di Turing può riconoscere.

Definizione 2 (Macchina che Decide un Linguaggio). *Una Macchina di Turing **decide** un linguaggio L se per ogni stringa w in input:*

- Se $w \in L$, la macchina si ferma in uno stato accettante.
- Se $w \notin L$, la macchina si ferma in uno stato non accettante.

In sintesi, una macchina che decide un linguaggio è garantita fermarsi su ogni input, fornendo una risposta "sì" (accettazione) o "no" (rifiuto).

Definizione 3 (Macchina che Accetta un Linguaggio). *Una Macchina di Turing **accetta** un linguaggio L se:*

- Per ogni stringa $w \in L$, la macchina si ferma in uno stato accettante.
- Per ogni stringa $w \notin L$, la macchina non accetta w . Questo significa che la macchina può fermarsi in uno stato non accettante oppure non fermarsi affatto (loop infinito).

Le macchine che accettano linguaggi non danno una garanzia di terminazione per input che non fanno parte del linguaggio.

Per gli esercizi odierni, progetteremo macchine che **decidono** i linguaggi specificati.

2 Tecniche di Programmazione delle Macchine di Turing

La programmazione delle Macchine di Turing richiede una visualizzazione chiara del loro funzionamento. Il suggerimento principale è "vedere il filmato" del movimento della testina e delle modifiche sul nastro nella propria mente, e poi tradurre questo filmato in una sequenza di transizioni di stato.

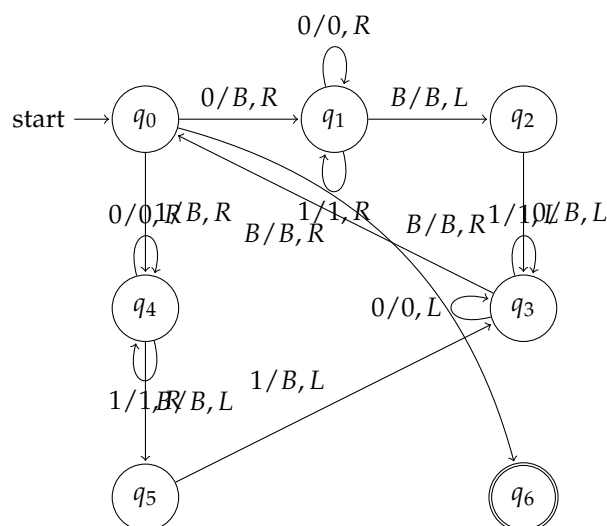
Un'altra tecnica importante è **evitare di cancellare i simboli di input** (sostituendoli con blank) se non strettamente necessario, specialmente per stringhe complesse. Questo può creare "buchi" sul nastro, rendendo difficile tenere traccia della posizione relativa della testina e delle parti rimanenti della stringa. È preferibile **marcare i simboli di input** con simboli speciali dell'alfabeto del nastro (Γ), mantenendo l'input compatto.

2.1 Esercizio 1: Linguaggio dei Palindromi $L = \{ww^R \mid w \in \{0,1\}^*\}$

Vogliamo progettare una MT che decide il linguaggio delle stringhe palindromiche con due parti w e w^R concatenate. L'alfabeto è $\{0,1\}$. Esempi: ε , 00, 11, 0110, 1001.

Esempio 1 (Strategia). La strategia consiste nel confrontare il primo carattere di w con l'ultimo carattere di w^R , il secondo con il penultimo, e così via, fino a quando tutti i caratteri sono stati confrontati.

1. Partire dal primo simbolo non marcato del nastro.
2. Leggere il simbolo (es. 0 o 1), marcarlo (sostituirlo con un Blank B).
3. Spostare la testina fino alla fine della stringa (trovando il primo B a destra), quindi tornare indietro di un passo per posizionarsi sull'ultimo simbolo.
4. Verificare che l'ultimo simbolo sia uguale a quello letto inizialmente. Se sì, marcarlo (sostituirlo con B).
5. Spostare la testina all'inizio della parte non marcata della stringa.
6. Ripetere il processo fino a quando tutti i simboli sono stati marcati.
7. Se tutti i simboli sono stati marcati correttamente, la stringa è accettata.



Spiegazione delle transizioni:

- **Stato q_0 (Iniziale):**
 - Se legge 0: scrive B (blank), sposta a destra (R), va in q_1 .
 - Se legge 1: scrive B , sposta a destra (R), va in q_4 .
 - Se legge B : scrive B , sposta a destra (R), va in q_6 (stato accettante). Questo accetta la stringa vuota.
- **Stato q_1 (Cerca fine stringa dopo aver letto 0):**
 - Se legge 0 o 1: lascia il simbolo, sposta a destra (R), rimane in q_1 . Salta tutti i caratteri.
 - Se legge B : lascia B , sposta a sinistra (L), va in q_2 . Questo indica che la testina ha raggiunto la fine della stringa e si posiziona sull'ultimo carattere.
- **Stato q_2 (Confronta ultimo carattere per 0):**
 - Se legge 0: scrive B , sposta a sinistra (L), va in q_3 . Il 0 iniziale è stato confrontato con successo.
 - Se legge 1 o B : la stringa non è del formato corretto (implica rifiuto non fermandosi in stato accettante).
- **Stato q_3 (Torna all'inizio):**
 - Se legge 0 o 1: lascia il simbolo, sposta a sinistra (L), rimane in q_3 . Salta tutti i caratteri.
 - Se legge B : lascia B , sposta a destra (R), va in q_0 . Questo indica che la testina ha raggiunto l'inizio del nastro e si posiziona sul prossimo carattere non marcato.
- **Stato q_4 (Cerca fine stringa dopo aver letto 1):**
 - Se legge 0 o 1: lascia il simbolo, sposta a destra (R), rimane in q_4 . Salta tutti i caratteri.
 - Se legge B : lascia B , sposta a sinistra (L), va in q_5 . Raggiunto fine stringa, posiziona sull'ultimo.
- **Stato q_5 (Confronta ultimo carattere per 1):**
 - Se legge 1: scrive B , sposta a sinistra (L), va in q_3 . L'1 iniziale è stato confrontato con successo.
 - Se legge 0 o B : la stringa non è del formato corretto (implica rifiuto).
- **Stato q_6 (Accettante):** Se la macchina raggiunge questo stato, la stringa è accettata.

Questa MT decide il linguaggio, poiché su ogni input valido o meno, si fermerà in uno stato accettante o non accettante.

2.2 Esercizio 2: Linguaggio $L = \{a^n b^m \mid m > n > 0\}$

Vogliamo progettare una MT che decide il linguaggio di stringhe con sequenze di 'a' seguite da sequenze di 'b', dove il numero di 'b' è strettamente maggiore del numero di 'a', ed entrambi sono maggiori di 0. Esempio: aab^3, ab^2 .

Esempio 2 (Strategia). La strategia prevede di marcare un 'a' e poi la prima 'b' corrispondente, ripetendo il processo. Alla fine, si verifica che tutte le 'a' siano state marcate e che ci siano 'b' non marcate rimaste.

1. Marcare la prima 'a' non marcata (es. con 'X').
2. Spostarsi a destra, saltando altre 'a' e 'Y' (b già marcate) fino a trovare la prima 'b' non marcata.
3. Marcare questa 'b' (es. con 'Y').
4. Spostarsi a sinistra, saltando 'a', 'b', 'Y' fino a trovare la 'X' più a destra.
5. Spostarsi a destra di una posizione per trovare la prossima 'a' non marcata e ripetere.
6. Una volta che tutte le 'a' sono state marcate, si controlla che ci siano 'b' non marcate rimaste. Se sì, la stringa è accettata.

Alfabeto del nastro $\Gamma = \{a, b, X, Y, B\}$. **Stati:** q_0 (iniziale), q_1 (marcata 'a', cerca 'b'), q_2 (salta 'Y' per trovare 'b'), q_3 (marcata 'b', torna indietro), q_4 (torna indietro per 'X'), q_5 (accettazione 'b' residue), q_6 (accettante).

Transizioni:

- **Stato q_0 (Iniziale / Cerca 'a'):**
 - $q_0 \xrightarrow{a/X,R} q_1$ (Marca la prima 'a' con 'X', si muove a destra)
 - $q_0 \xrightarrow{Y/Y,R} q_5$ (Se trova una 'Y', significa che tutte le 'a' sono state marcate e si controllano le 'b' residue)
 - Se q_0 legge B (stringa vuota) o b (stringa non valida, $n \leq 0$) \rightarrow implicitamente rifiuta.
- **Stato q_1 (Marca 'a', cerca 'b'):**
 - $q_1 \xrightarrow{a/a,R} q_1$ (Salta altre 'a')
 - $q_1 \xrightarrow{Y/Y,R} q_2$ (Salta 'b' già marcate con 'Y')
 - $q_1 \xrightarrow{b/Y,L} q_3$ (Marca la prima 'b' con 'Y', va a sinistra. Questo è il caso in cui non ci sono 'Y' tra 'a' e 'b' non marcate)
- **Stato q_2 (Salto 'Y' intermedie):**
 - $q_2 \xrightarrow{Y/Y,R} q_2$ (Continua a saltare 'Y' già marcate)
 - $q_2 \xrightarrow{b/Y,L} q_3$ (Marca la 'b' con 'Y', va a sinistra)
- **Stato q_3 (Marca 'b', torna indietro):**
 - $q_3 \xrightarrow{a/a,L} q_3$ (Salta 'a' mentre torna indietro)

- $q_3 \xrightarrow{Y/Y,L} q_3$ (Salta 'Y' mentre torna indietro)
- $q_3 \xrightarrow{X/X,R} q_0$ (Ha trovato l'X della 'a' iniziale. Sposta a destra per la prossima 'a' da marcare)

• **Stato q_5 (Verifica 'b' residue):**

- $q_5 \xrightarrow{Y/Y,R} q_5$ (Salta 'Y' residue)
- $q_5 \xrightarrow{b/b,R} q_5$ (Salta 'b' residue non marcate - devono esistere per $m > n$)
- $q_5 \xrightarrow{B/B,R} q_6$ (Se trova B, significa che ci sono più 'b' delle 'a' ed è un blank. Stringa accettata)

• **Stato q_6 (Accettante):** Stringa accettata.

Questa MT decide il linguaggio, rifiutando stringhe che non soddisfano le condizioni (es. $n \geq m$, ordine sbagliato dei caratteri, ecc.).

2.3 Esercizio 3: Linguaggio $L = \{a^n b^m \mid n > 0, m = 2n\}$

Vogliamo progettare una MT che decide il linguaggio di stringhe con sequenze di 'a' seguite da sequenze di 'b', dove il numero di 'b' è il doppio del numero di 'a', ed n è maggiore di 0. Esempio: aab^4, ab^2 .

Esempio 3 (Strategia). *Molto simile al precedente, ma per ogni 'a' marcata, si devono marcare due 'b'.*

1. Marcare la prima 'a' non marcata (con 'X').
2. Spostarsi a destra, saltando 'a' e 'Y' fino a trovare la prima 'b' non marcata. Marcarla (con 'Y').
3. Spostarsi ancora a destra, trovare la seconda 'b' non marcata. Marcarla (con 'Y').
4. Spostarsi a sinistra, saltando tutti i simboli marcati e non, fino a trovare la 'X' più a destra.
5. Spostarsi a destra di una posizione per trovare la prossima 'a' non marcata e ripetere.
6. Una volta che tutte le 'a' sono state marcate, si controlla che tutte le 'b' siano state marcate. Se sì, la stringa è accettata.

Alfabeto del nastro $\Gamma = \{a, b, X, Y, B\}$. **Stati:** q_0 (iniziale), q_1 (marcata 'a', cerca prima 'b'), q_2 (marcata prima 'b', cerca seconda 'b'), q_3 (marcata seconda 'b', torna indietro), q_4 (torna indietro), q_5 (accettazione 'Y' residue), q_6 (accettante).

Transizioni:

• **Stato q_0 (Iniziale / Cerca 'a'):**

- $q_0 \xrightarrow{a/X,R} q_1$ (Marca la prima 'a' con 'X')
- $q_0 \xrightarrow{Y/Y,R} q_5$ (Se trova 'Y', tutte le 'a' sono state marcate, si controllano solo 'b' marcate)

• **Stato q_1 (Marca 'a', cerca prima 'b'):**

- $q_1 \xrightarrow{a/a,R} q_1$ (Salta 'a')

- $q_1 \xrightarrow{Y/Y,R} q_1$ (Salta 'Y' già marcate)
- $q_1 \xrightarrow{b/Y,R} q_2$ (Marca la prima 'b' con 'Y', si muove a destra per cercare la seconda 'b')

• **Stato q_2 (Cerca seconda 'b'):**

- $q_2 \xrightarrow{b/Y,L} q_3$ (Marca la seconda 'b' con 'Y', si muove a sinistra per tornare)

• **Stato q_3 (Marca seconda 'b', torna indietro):**

- $q_3 \xrightarrow{a/a,L} q_3$ (Salta 'a')
- $q_3 \xrightarrow{Y/Y,L} q_3$ (Salta 'Y')
- $q_3 \xrightarrow{X/X,R} q_0$ (Trovato 'X', sposta a destra per la prossima 'a')

• **Stato q_5 (Verifica 'b' residue):**

- $q_5 \xrightarrow{Y/Y,R} q_5$ (Salta 'Y' residue)
- $q_5 \xrightarrow{B/B,R} q_6$ (Se trova B, tutte le 'b' sono state marcate correttamente. Accetta)

• **Stato q_6 (Accettante):** Stringa accettata.

Condizioni di rifiuto implicite: se non si trovano due 'b' per ogni 'a', o se l'ordine dei simboli non è 'a' poi 'b'.

2.4 Esercizio 4: Linguaggio $L = \{w\#w \mid w \in \{a,b\}^+\}$

Vogliamo progettare una MT che decide il linguaggio di stringhe composte da una sequenza w , seguita da un separatore '#', e poi dalla stessa sequenza w . w deve avere almeno un simbolo. Esempi: $a\#a$, $ab\#ab$, $baba\#baba$.

Esempio 4 (Strategia). *La strategia si basa sul confronto carattere per carattere delle due metà della stringa, usando il '#' come punto di riferimento.*

1. Marcare il primo simbolo non marcato della prima w (es. 'a' o 'b' con 'X').
2. Spostarsi a destra, saltando tutti i simboli intermedi, fino a trovare il '#'.
3. Superare il '#', e poi saltare i simboli già marcati ('X') nella seconda w .
4. Trovare il simbolo corrispondente nella seconda w (deve essere uguale a quello marcato nel passo 1), marcarlo (con 'X').
5. Spostarsi a sinistra, saltando tutti i simboli intermedi, fino a trovare il '#'.
6. Superare il '#', e poi saltare i simboli già marcati ('X') nella prima w .
7. Posizionarsi sul prossimo simbolo non marcato nella prima w e ripetere il ciclo.
8. Se tutti i simboli sono stati marcati correttamente, la stringa è accettata.

Alfabeto del nastro $\Gamma = \{a, b, \#, X, B\}$. **Stati:** q_0 (iniziale), q_1 (marcata 'a', cerca match), q_2 (dopo '#', salta 'X', cerca 'a'), q_3 (match trovato, torna indietro), q_4 (dopo '#', torna indietro a 'X'), q_5 (check finale 'X' prima di '#'), q_6 (marcata 'b', cerca match), q_7 (dopo '#', salta 'X', cerca 'b'), q_8 (dopo '#', check finale 'X' dopo '#'), q_9 (accettante).

Transizioni:

- **Stato q_0 (Iniziale / Cerca simbolo in w_1):**

- $q_0 \xrightarrow{a/X,R} q_1$ (Marca 'a' con 'X', va a destra)
- $q_0 \xrightarrow{b/X,R} q_6$ (Marca 'b' con 'X', va a destra)
- $q_0 \xrightarrow{X/X,R} q_5$ (Tutte le prime w sono marcate, passa alla fase di controllo finale)

- **Stato q_1 (Marcata 'a', cerca 'a' in w_2):**

- $q_1 \xrightarrow{a/a,R} q_1$ (Salta 'a')
- $q_1 \xrightarrow{b/b,R} q_1$ (Salta 'b')
- $q_1 \xrightarrow{\#/\#,R} q_2$ (Passa il separatore '#')

- **Stato q_2 (Dopo '#', cerca 'a' in w_2):**

- $q_2 \xrightarrow{X/X,R} q_2$ (Salta 'X' già marcate)
- $q_2 \xrightarrow{a/X,L} q_3$ (Trova 'a', marca con 'X', va a sinistra per tornare)

- **Stato q_6 (Marcata 'b', cerca 'b' in w_2):**

- $q_6 \xrightarrow{a/a,R} q_6$ (Salta 'a')
- $q_6 \xrightarrow{b/b,R} q_6$ (Salta 'b')
- $q_6 \xrightarrow{\#/\#,R} q_7$ (Passa il separatore '#')

- **Stato q_7 (Dopo '#', cerca 'b' in w_2):**

- $q_7 \xrightarrow{X/X,R} q_7$ (Salta 'X' già marcate)
- $q_7 \xrightarrow{b/X,L} q_3$ (Trova 'b', marca con 'X', va a sinistra per tornare)

- **Stato q_3 (Match trovato, torna indietro):**

- $q_3 \xrightarrow{X/X,L} q_3$ (Salta 'X' già marcate)
- $q_3 \xrightarrow{a/a,L} q_3$ (Salta 'a')
- $q_3 \xrightarrow{b/b,L} q_3$ (Salta 'b')
- $q_3 \xrightarrow{\#/\#,L} q_4$ (Passa il separatore '#')

- **Stato q_4 (Dopo '#', torna all'inizio di w_1):**

- $q_4 \xrightarrow{a/a,L} q_4$ (Salta 'a')
- $q_4 \xrightarrow{b/b,L} q_4$ (Salta 'b')
- $q_4 \xrightarrow{X/X,R} q_0$ (Trovata 'X' della 'a' o 'b' iniziale, si posiziona a destra per la prossima iterazione)

• **Stato q_5 (Controllo finale, parte w_1):**

- $q_5 \xrightarrow{X/X,R} q_5$ (Salta 'X' nella prima metà)
- $q_5 \xrightarrow{\#/ \#,R} q_8$ (Passa il separatore '#', ora controlla seconda metà)

• **Stato q_8 (Controllo finale, parte w_2):**

- $q_8 \xrightarrow{X/X,R} q_8$ (Salta 'X' nella seconda metà)
- $q_8 \xrightarrow{B/B,R} q_9$ (Se trova B, tutte le parti sono state marcate, stringa accettata)

• **Stato q_9 (Accettante):** Stringa accettata.

Questa MT decide il linguaggio, rifiutando stringhe come $a\#b$ o $aa\#a$ o $a\#bb$.

2.5 Esercizio 5: Linguaggio $L = \{a^n b^n c^n \mid n > 0\}$

Vogliamo progettare una MT che decide il linguaggio di stringhe con un numero uguale di 'a', 'b', e 'c', in sequenza $a^*b^*c^*$. n deve essere maggiore di 0. Esempio: $abc, aabbcc$.

Esempio 5 (Strategia). La strategia è marcare un' 'a', poi una 'b', poi una 'c', e ripetere fino a quando tutti i caratteri sono stati marcati.

1. Marcare la prima 'a' non marcata (con 'X').
2. Spostarsi a destra, saltando altre 'a' e 'Y' (b già marcate), fino a trovare la prima 'b' non marcata. Marcarla (con 'Y').
3. Spostarsi a destra, saltando altre 'b' e 'Z' (c già marcate), fino a trovare la prima 'c' non marcata. Marcarla (con 'Z').
4. Spostarsi a sinistra, saltando tutti i simboli marcati e non, fino a trovare la 'X' più a destra.
5. Spostarsi a destra di una posizione per trovare la prossima 'a' non marcata e ripetere.
6. Una volta che tutte le 'a' sono state marcate, si controlla che tutte le 'b' e 'c' siano state marcate anch'esse.

Alfabeto del nastro $\Gamma = \{a, b, c, X, Y, Z, B\}$. **Stati:** q_0 (iniziale), q_1 (marcata 'a', cerca 'b'), q_2 (salta 'Y', cerca 'b'), q_3 (marcata 'b', cerca 'c'), q_4 (salta 'Z', cerca 'c'), q_5 (marcata 'c', torna indietro), q_6 (accettazione 'Y' residue), q_7 (accettazione 'Z' residue), q_8 (accettante).

Transizioni:

• **Stato q_0 (Iniziale / Cerca 'a'):**

- $q_0 \xrightarrow{a/X,R} q_1$ (Marca 'a' con 'X')

- $q_0 \xrightarrow{Y/Y,R} q_6$ (Se trova 'Y', tutte le 'a' sono state marcate, si controllano 'b' e 'c' marcate)
- **Stato q_1 (Marca 'a', cerca 'b'):**
 - $q_1 \xrightarrow{a/a,R} q_1$ (Salta 'a')
 - $q_1 \xrightarrow{Y/Y,R} q_2$ (Salta 'Y' già marcate)
 - $q_1 \xrightarrow{b/Y,R} q_3$ (Marca la prima 'b' con 'Y'. Caso iniziale senza 'Y' intermedie)
- **Stato q_2 (Salto 'Y' intermedie):**
 - $q_2 \xrightarrow{Y/Y,R} q_2$ (Continua a saltare 'Y' già marcate)
 - $q_2 \xrightarrow{b/Y,R} q_3$ (Marca 'b' con 'Y')
- **Stato q_3 (Marca 'b', cerca 'c'):**
 - $q_3 \xrightarrow{b/b,R} q_3$ (Salta 'b')
 - $q_3 \xrightarrow{Z/Z,R} q_4$ (Salta 'Z' già marcate)
 - $q_3 \xrightarrow{c/Z,L} q_5$ (Marca la prima 'c' con 'Z'. Caso iniziale senza 'Z' intermedie)
- **Stato q_4 (Salto 'Z' intermedie):**
 - $q_4 \xrightarrow{Z/Z,R} q_4$ (Continua a saltare 'Z' già marcate)
 - $q_4 \xrightarrow{c/Z,L} q_5$ (Marca 'c' con 'Z')
- **Stato q_5 (Marca 'c', torna indietro):**
 - $q_5 \xrightarrow{Z/Z,L} q_5$ (Salta 'Z')
 - $q_5 \xrightarrow{c/c,L} q_5$ (Salta 'c')
 - $q_5 \xrightarrow{Y/Y,L} q_5$ (Salta 'Y')
 - $q_5 \xrightarrow{b/b,L} q_5$ (Salta 'b')
 - $q_5 \xrightarrow{a/a,L} q_5$ (Salta 'a')
 - $q_5 \xrightarrow{X/X,R} q_0$ (Trovato 'X', sposta a destra per la prossima 'a')
- **Stato q_6 (Verifica 'Y' residue):**
 - $q_6 \xrightarrow{Y/Y,R} q_6$ (Salta 'Y' residue)
 - $q_6 \xrightarrow{Z/Z,R} q_7$ (Se trova 'Z', passa a controllare le 'c')
- **Stato q_7 (Verifica 'Z' residue):**
 - $q_7 \xrightarrow{Z/Z,R} q_7$ (Salta 'Z' residue)

– $q_7 \xrightarrow{B/B,R} q_8$ (Se trova B , tutte le 'a', 'b', 'c' sono state marcate correttamente. Accetta)

- **Stato q_8 (Accettante):** Stringa accettata.

Questa MT decide il linguaggio. Implicitamente rifiuta stringhe con ordine errato dei caratteri (es. acb) o conteggi non corrispondenti (es. $aabc$).