

# Macchine di Turing Non Deterministiche: Esercitazioni

Appunti da Trascrizione Automatica

30 giugno 2025

## Indice

<b>1</b>	<b>Introduzione alle Macchine di Turing Non Deterministiche (NDTM)</b>	<b>2</b>
<b>2</b>	<b>Esercizi</b>	<b>2</b>
2.1	Esercizio 1: Sottostringa Palindroma di Lunghezza Variabile . . . . .	2
2.1.1	Strategia della Macchina di Turing . . . . .	2
2.1.2	Descrizione degli Stati e delle Transizioni . . . . .	3
2.2	Esercizio 2: Coppie di $W_i$ . . . . .	5
2.2.1	Strategia della Macchina di Turing . . . . .	5
2.2.2	Descrizione degli Stati e delle Transizioni . . . . .	5
2.3	Esercizio 3: Parità di Lunghezza di $W_i$ . . . . .	7
2.3.1	Strategia della Macchina di Turing . . . . .	7
2.3.2	Descrizione degli Stati e delle Transizioni . . . . .	7
2.4	Esercizio 4: $A$ o $B^R$ in $W_i$ in base alla Parità dell'Indice $i$ . . . . .	9
2.4.1	Strategia della Macchina di Turing . . . . .	9
2.4.2	Descrizione degli Stati e delle Transizioni . . . . .	9
<b>3</b>	<b>Conclusioni</b>	<b>11</b>

## 1 Introduzione alle Macchine di Turing Non Deterministiche (NDTM)

Riprendiamo dalla discussione sulle Macchine di Turing Non Deterministiche (NDTM). Una caratteristica fondamentale delle NDTM è la loro capacità di "indovinare" o "guessare" porzioni di stringa o decisioni computazionali. Questa non è una capacità intrinseca della macchina di indovinare nel senso umano, né implica una computazione parallela di tutte le possibilità. È una metafora per descrivere il fatto che se esiste almeno una sequenza di scelte che porta all'accettazione, allora la macchina accetterà.

Tuttavia, per garantire che una NDTM accetti solo le stringhe che appartengono al linguaggio desiderato, la fase di guess deve essere sempre seguita da una fase di check (controllo). La fase di check è cruciale per filtrare le "scelte sbagliate" fatte dalla macchina non deterministica, assicurando che solo le computazioni corrette portino a uno stato di accettazione.

Un esempio pratico di questa capacità delle NDTM è la possibilità di scrivere in anticipo su un nastro ausiliario delle stringhe che saranno necessarie in un momento successivo della computazione. Questo comportamento è particolarmente utile quando si analizzano le classi di complessità computazionale (es. NP).

## 2 Esercizi

Analizziamo alcuni esercizi per applicare i concetti delle NDTM.

### 2.1 Esercizio 1: Sottostringa Palindroma di Lunghezza Variabile

**Definizione 1** (Linguaggio  $L_1$ ). Sia  $L_1$  il linguaggio definito come:  $L_1 = \{X^N \# W_1 \# W_2 \# \dots \# W_N \mid N > 0, W_i \in \{a, b, c, d\}^+, \forall i \in [1, N], \exists S_i \subseteq W_i \text{ t.c. } |S_i| = i \text{ e } S_i = S_i^R\}$  dove  $X^N$  indica  $N$  occorrenze del simbolo  $X$ .

#### 2.1.1 Strategia della Macchina di Turing

La macchina utilizzerà cinque nastri:

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Conteggio  $N$ ):** Per memorizzare il numero di blocchi  $W_i$ .
- **Nastro 3 (Conteggio  $i$ ):** Per memorizzare il valore corrente dell'indice  $i$ .
- **Nastro 4 (Guess  $S_i$ ):** Per scrivere la stringa  $S_i$  "indovinata".
- **Nastro 5 (Guess  $S_i$  per  $S_i^R$ ):** Una copia di  $S_i$  per il controllo di palindromia.

L'idea principale è che per ogni  $W_i$ , la macchina non deterministicamente "indovina" la stringa  $S_i$  sul Nastro 4 e 5. Successivamente, verifica che  $S_i$  sia palindroma (confrontando Nastro 4 e Nastro 5) e che sia effettivamente una sottostringa di  $W_i$  (confrontando Nastro 4 con il Nastro 1). La lunghezza di  $S_i$  viene controllata usando il Nastro 3.

### 2.1.2 Descrizione degli Stati e delle Transizioni

Sia  $\Sigma_I = \{X, \#, a, b, c, d\}$  l'alfabeto di input e  $\Gamma_T = \Sigma_I \cup \{B, \text{ausiliari}\}$  l'alfabeto del nastro. Usiamo  $\alpha$  per un simbolo generico da  $\{a, b, c, d\}$ .

- **Stato  $Q_0$  (Inizio):**
  - **Transizione per  $X^N$ :** Legge  $X$  dal Nastro 1, lo riscrive e si muove a destra. Scrive  $X$  sul Nastro 2 (inizialmente vuoto) e si muove a destra. Passa allo stato  $Q_1$ . Questo ciclo si ripete per tutti gli  $X$  iniziali.
  - $(X, B, B, B, B) \rightarrow (X, X, B, B, B), (R, R, S, S, S)$
- **Stato  $Q_1$  (Copia  $N$  e Inizio  $W_1$ ):**
  - **Transizione per  $\#$ :** Quando legge  $\#$  sul Nastro 1, lo riscrive e si muove a destra (posizionandosi all'inizio di  $W_1$ ). Sul Nastro 2 (che contiene gli  $X$  per  $N$ ), legge un  $X$ , lo cancella ( $B$ ) e si muove a sinistra (si posiziona sull'ultimo  $X$  restante o  $B$ ). Sul Nastro 3 (inizialmente vuoto), scrive  $X$  e si muove a destra (questo inizia il conteggio di  $i = 1$ ). Passa a  $Q_2$ .
  - $(\#, X, B, B, B) \rightarrow (\#, B, X, B, B), (R, L, R, S, S)$
- **Stato  $Q_2$  (Guess  $S_i$ ):** Questo stato gestisce la generazione non deterministica di  $S_i$  sui nastri 4 e 5. Il loop avviene basandosi sul Nastro 3 (che contiene  $i$ ).
  - **Transizione (Loop per guess  $S_i$ ):** Mentre Nastro 3 contiene  $X$  (cioè,  $S_i$  non ha ancora raggiunto la lunghezza  $i$ ):
    - \* Nastro 1: Legge  $\alpha$  (qualsiasi simbolo di input), lo riscrive e si muove a destra (continua a leggere  $W_i$ ).
    - \* Nastro 2: Non modificato.
    - \* Nastro 3: Legge  $X$ , lo riscrive e si muove a destra (avanza nel conteggio di  $i$ ).
    - \* Nastro 4: Legge  $B$ , scrive  $\alpha$  (un simbolo non deterministico dall'alfabeto di input) e si muove a destra.
    - \* Nastro 5: Legge  $B$ , scrive  $\alpha$  (lo stesso simbolo di Nastro 4) e si muove a destra.
 Questo loop rimane in  $Q_3$  (si passa a  $Q_3$  nella descrizione, il diagramma usa  $Q_2 \rightarrow Q_3 \rightarrow Q_3$ ).
  - $(\alpha, \text{any}, X, B, B) \rightarrow (\alpha, \text{any}, X, \alpha, \alpha), (R, S, R, R, R)$  (Questo  $\alpha$  su Nastro 1 è solo lettura,  $\alpha$  è il simbolo guessed).
  - **Transizione (Fine guess  $S_i$ ):** Quando Nastro 3 legge  $B$  (ha scritto  $i$  simboli su Nastro 4 e 5):
    - \* Nastro 1: Legge  $\alpha$ , lo riscrive e si muove a destra.
    - \* Nastro 2: Non modificato.
    - \* Nastro 3: Legge  $B$ , lo riscrive e si muove a sinistra (riavvolge il conteggio di  $i$ ).
    - \* Nastro 4: Legge  $B$ , lo riscrive e si muove a sinistra (riavvolge Nastro 4).
    - \* Nastro 5: Legge  $B$ , lo riscrive e si muove a sinistra (riavvolge Nastro 5).
 Passa a  $Q_4$ .
  - $(\alpha, \text{any}, B, B, B) \rightarrow (\alpha, \text{any}, B, B, B), (R, S, L, L, L)$

- **Stato  $Q_4$  (Riavvolgimento):** Questo stato riavvolge i nastri 3, 4 e 5 per prepararsi al controllo. Si continua a muoversi a sinistra su Nastro 3, 4 e 5 finché non si raggiunge il B iniziale.

- **Transizione (Loop di riavvolgimento):** Mentre Nastro 4 non è B:

- \* Nastro 1: Ignorato (ma continua a leggere  $W_i$ ).
- \* Nastro 2: Ignorato.
- \* Nastro 3: Legge  $X$ , lo riscrive, si muove a sinistra.
- \* Nastro 4: Legge  $\alpha$ , lo riscrive, si muove a sinistra.
- \* Nastro 5: Legge  $\alpha$ , lo riscrive, si muove a sinistra.

Questo loop rimane in  $Q_4$ .

- $(\text{any}, \text{any}, X, \alpha, \alpha) \rightarrow (\text{any}, \text{any}, X, \alpha, \alpha), (S, S, L, L, L)$

- **Transizione (Fine riavvolgimento):** Quando Nastro 4 e Nastro 5 leggono B (sono all'inizio di  $S_i$ ):

- \* Nastro 1: Ignorato.
- \* Nastro 2: Ignorato.
- \* Nastro 3: Legge B, lo riscrive, si muove a destra.
- \* Nastro 4: Legge B, lo riscrive, si muove a destra.
- \* Nastro 5: Legge B, lo riscrive, si muove a destra (posiziona il capo del Nastro 5 sulla fine di  $S_i$  per il reverse check).

Passa a  $Q_5$ .

- $(\text{any}, \text{any}, B, B, B) \rightarrow (\text{any}, \text{any}, B, B, B), (S, S, R, R, R)$

- **Stato  $Q_5$  (Check  $S_i$  e  $S_i^R$  in  $W_i$ ):** Questo stato non deterministicamente cerca l'inizio di  $S_i$  in  $W_i$  sul Nastro 1 e contemporaneamente verifica la palindromia di  $S_i$  tra Nastro 4 (leggendo in avanti) e Nastro 5 (leggendo all'indietro).

- **Transizione (Salto in  $W_i$ ):** Nondeterministicamente salta caratteri in  $W_i$  sul Nastro 1 fino a trovare un possibile inizio di  $S_i$ .

- \* Nastro 1: Legge  $\alpha$ , lo riscrive, si muove a destra.

Loop su  $Q_5$ .

- $(\alpha, \text{any}, \text{any}, \text{any}, \text{any}) \rightarrow (\alpha, \text{any}, \text{any}, \text{any}, \text{any}), (R, S, S, S, S)$

- **Transizione (Confronto e Palindromia):** Quando si decide di iniziare il confronto:

- \* Nastro 1: Legge  $\alpha$ , lo riscrive, si muove a destra (confronta con  $S_i$ ).
- \* Nastro 4: Legge  $\alpha$ , lo cancella (B), si muove a destra (consuma  $S_i$ ).
- \* Nastro 5: Legge  $\alpha$ , lo cancella (B), si muove a sinistra (consuma  $S_i^R$ ).

Passa a  $Q_6$ . Questo loop continua in  $Q_6$ .

- $(\alpha, \text{any}, \text{any}, \alpha, \alpha) \rightarrow (\alpha, \text{any}, \text{any}, B, B), (R, S, S, R, L)$

- **Stato  $Q_6$  (Continuazione del Check):**

- **Transizione (Fine del confronto):** Quando Nastro 4 e Nastro 5 leggono B (hanno verificato tutta  $S_i$ ):

- \* Nastro 1: Ignorato.

\* Nastro 4: Legge B, lo riscrive, si ferma.

\* Nastro 5: Legge B, lo riscrive, si ferma.

Passa a  $Q_7$ .

– (any, any, any, B, B)  $\rightarrow$  (any, any, any, B, B), (S, S, S, S, S)

• **Stato  $Q_7$  (Controllo Prossima  $W_i$  o Accettazione):**

– **Transizione (Prossima  $W_i$ ):** Se Nastro 2 contiene ancora X (ci sono altre  $W_i$  da processare):

\* Nastro 1: Continua a leggere  $\alpha$  e si sposta a destra (fino a fine  $W_i$  o #).

\* Nastro 2: Legge X, lo riscrive, si ferma (la cancellazione avverrà tornando a  $Q_1$ ).

\* Nastro 3: Riavvolge a sinistra fino a B.

Questa è una transizione composta:  $\{(\alpha, X, \alpha, B, B) \rightarrow (\alpha, X, \alpha, B, B), (R, S, L, S, S)\}$  (Loop per riavvolgere Nastro 3) Al raggiungimento del B su Nastro 3 e # su Nastro 1:  $(\#, X, B, B, B) \rightarrow (\#, X, B, B, B), (R, S, R, S, S)$  e torna a  $Q_1$  (per processare il prossimo # e la prossima  $W_i$ , decrementando  $N$  su Nastro 2).

– **Transizione (Accettazione):** Se Nastro 2 legge B (non ci sono più X per  $N$ , quindi tutte le  $W_i$  sono state processate):

\* Nastro 1: Controlla che sia finito (legga B).

\* Nastro 2: Legge B, lo riscrive, si ferma.

Passa a  $Q_{acc}$ .

– (B, B, any, B, B)  $\rightarrow$  (B, B, any, B, B), (S, S, S, S, S) (e si sposta in  $Q_{acc}$ ).

## 2.2 Esercizio 2: Coppie di $W_i$

**Definizione 2** (Linguaggio  $L_2$ ). Sia  $L_2$  il linguaggio definito come:  $L_2 = \{A\#B\#W_1W_1W_2W_2\dots W_NW_N \mid A, B, W_i \in \{0, 1\}^+, |A| > |B|, N = |A| - |B|, |W_i| \geq |B|\}$

### 2.2.1 Strategia della Macchina di Turing

- **Nastro 1 (Input):** Contiene  $A\#B\#W_1W_1\dots$
- **Nastro 2 (Conteggio  $N$ ):** Per memorizzare  $N = |A| - |B|$ .
- **Nastro 3 (Copia  $B$ ):** Per memorizzare la stringa  $B$ .
- **Nastro 4 (Copia  $W_i$ ):** Per memorizzare  $W_i$  e confrontarla con la sua seconda occorrenza.

### 2.2.2 Descrizione degli Stati e delle Transizioni

Sia  $\Sigma_I = \{0, 1, \#\}$  l'alfabeto di input e  $\Gamma_T = \Sigma_I \cup \{B, X\}$  l'alfabeto del nastro. Usiamo  $\alpha$  per un simbolo generico da  $\{0, 1\}$ .

• **Stato  $Q_0$  (Copia  $A$ ):**

– Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Scrive X su Nastro 2,  $R$ .

– Alla lettura di #: riscrive #,  $R$ . Su Nastro 2,  $S$ . Passa a  $Q_1$ .

- **Stato  $Q_1$  (Copia  $B$  e Calcolo  $N$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ .
  - Legge  $X$  da Nastro 2, lo cancella ( $B$ ),  $L$ . (Inizia a calcolare  $|A| - |B|$ ).
  - Scrive  $\alpha$  su Nastro 3,  $R$ . (Copia  $B$ ).
  - Loop in  $Q_2$ .
- **Stato  $Q_2$  (Fine Calcolo  $N$  e Copia  $B$ ):**
  - Alla lettura di  $\#$  (dopo  $B$ ): riscrive  $\#$ ,  $R$ .
  - Su Nastro 2: legge  $X$ , lo riscrive,  $S$ . (Controlla  $|A| > |B|$ : se Nastro 2 fosse  $B$  qui, allora  $|A| \leq |B|$ , quindi rifiuterebbe implicitamente).
  - Su Nastro 3: legge  $B$ , riscrive  $B$ ,  $L$ . (Riavvolge Nastro 3 per preparare  $B$  per confronti futuri).
  - Passa a  $Q_3$ .
- **Stato  $Q_3$  (Processa  $W_i$  - prima occorrenza):** Questo stato si occupa di copiare la prima occorrenza di  $W_i$  sul Nastro 4 e di verificare che  $|W_i| \geq |B|$ .
  - Nondeterministicamente, la macchina può saltare caratteri in  $W_i$  finché non decide di iniziare a copiare la sottostringa  $W_i$  e allo stesso tempo verificare  $|W_i| \geq |B|$ .
  - Loop in  $Q_3$ : Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Copia  $\alpha$  su Nastro 4,  $R$ .
  - Questo loop continua finché non si raggiunge la fine di  $W_i$  (prossimo  $\#$  o  $B$ ).
  - Alla lettura di  $\#$ : riscrive  $\#$ ,  $R$ .
  - Cancella un  $X$  dal Nastro 2,  $L$ . (Decrementa  $N$ ).
  - Riavvolge Nastro 4 a sinistra per preparare il confronto. Passa a  $Q_5$ .
- **Stato  $Q_5$  (Verifica  $W_iW_i$ ):** Confronta la  $W_i$  copiata sul Nastro 4 con la seconda occorrenza di  $W_i$  sul Nastro 1.
  - Legge  $\alpha$  da Nastro 1, lo riscrive,  $R$ .
  - Legge  $\alpha$  da Nastro 4, lo cancella ( $B$ ),  $R$ .
  - Se i simboli non corrispondono, la macchina si blocca e rifiuta (non c'è transizione definita per questa situazione).
  - Loop in  $Q_6$ .
  - Quando Nastro 4 è  $B$  (fine di  $W_i$  copiata):
    - \* Su Nastro 1: deve esserci l'inizio della seconda  $W_i$ .
    - \* Passa a  $Q_7$ .
- **Stato  $Q_7$  (Prossima Coppia  $W_iW_i$  o Accettazione):**
  - Se Nastro 2 contiene  $X$  (ci sono altre coppie  $W_iW_i$ ):
    - \* Nastro 1: Posiziona il capo lettura sull'inizio della prossima  $W_i$  (dopo il  $\#$ ).
    - \* Riavvolge Nastro 3 e 4.
    - \* Torna a  $Q_3$ .
  - Se Nastro 2 è  $B$  (tutte le coppie  $W_iW_i$  sono state processate):
    - \* Nastro 1: Controlla che sia  $B$  (fine input).
    - \* Accetta ( $Q_{acc}$ ).

## 2.3 Esercizio 3: Parità di Lunghezza di $W_i$

**Definizione 3** (Linguaggio  $L_3$ ). Sia  $L_3$  il linguaggio definito come:  $L_3 = \{A\#B\#W_1\#W_2\#\dots\#W_N \mid A, B, W_i \in \{a, b, c, d\}^+, |A| > |B|, N = |A| - |B|, (se |W_i| \text{ è pari, allora } B \subseteq W_i) \wedge (se |W_i| \text{ è dispari, allora } B^R \subseteq W_i)\}$

### 2.3.1 Strategia della Macchina di Turing

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Conteggio  $N$ ):** Per memorizzare  $N = |A| - |B|$ .
- **Nastro 3 (Copia  $B$ ):** Per memorizzare la stringa  $B$ . La stringa  $B^R$  sarà controllata rileggendo  $B$  dal Nastro 3 all'indietro.
- **Nastro 4:** Non esplicitamente usata per copie permanenti in questa strategia, ma potrebbe servire per un flag temporaneo di parità o per verificare sottostringhe temporanee. Il professore descrive un metodo che evita un nastro specifico per la parità, ma usa stati distinti.

### 2.3.2 Descrizione degli Stati e delle Transizioni

Sia  $\Sigma_I = \{a, b, c, d, \#\}$  l'alfabeto di input e  $\Gamma_T = \Sigma_I \cup \{B, X\}$  l'alfabeto del nastro. Usiamo  $\alpha$  per un simbolo generico da  $\{a, b, c, d\}$ .

- **Stato  $Q_0$  (Copia  $A$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Scrive  $X$  su Nastro 2,  $R$ . (Copia  $A$  come  $X$ s su Nastro 2).
  - Alla lettura di  $\#$ : riscrive  $\#$ ,  $R$ . Su Nastro 2,  $L$ . Passa a  $Q_1$ .
- **Stato  $Q_1$  (Copia  $B$  e Calcolo  $N$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ .
  - Legge  $X$  da Nastro 2, lo cancella ( $B$ ),  $L$ . (Inizia a calcolare  $|A| - |B|$ ).
  - Scrive  $\alpha$  su Nastro 3,  $R$ . (Copia  $B$ ).
  - Loop in  $Q_2$ .
- **Stato  $Q_2$  (Fine Calcolo  $N$  e Copia  $B$ ):**
  - Alla lettura di  $\#$  (dopo  $B$ ): riscrive  $\#$ ,  $R$ .
  - Su Nastro 2: legge  $X$ , lo riscrive,  $S$ . (Controlla  $|A| > |B|$ ).
  - Su Nastro 3: legge  $B$ , riscrive  $B$ ,  $L$ . (Riavvolge Nastro 3 all'inizio di  $B$ ).
  - Passa a  $Q_3$ .
- **Stato  $Q_3$  (Inizio Parity Check per  $W_i$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . (Legge il primo carattere di  $W_i$ ).
  - Su Nastro 2: legge  $X$ , lo cancella ( $B$ ),  $L$ . (Decrementa  $N$  per questa  $W_i$ ).
  - Su Nastro 3:  $S$ .

- Passa a  $Q_4$  (stato per lunghezza dispari, avendo letto il 1° carattere).
- **Stato  $Q_4$  (Lunghezza  $W_i$  dispari / Prossimo carattere):**
  - **Transizione (Prossimo carattere - pari):** Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Passa a  $Q_5$ . (Se siamo in  $Q_4$  con  $k$  caratteri letti, il prossimo carattere rende la lunghezza  $k + 1$ , che è pari).
  - **Transizione (Fine  $W_i$  - dispari):** Legge # o B da Nastro 1, riscrive,  $L$ . (Ritorna all'inizio di  $W_i$ ). Passa a  $Q_6$ . (La lunghezza di  $W_i$  è dispari, quindi cerchiamo  $B^R$ ).
- **Stato  $Q_5$  (Lunghezza  $W_i$  pari / Prossimo carattere):**
  - **Transizione (Prossimo carattere - dispari):** Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Passa a  $Q_4$ . (Se siamo in  $Q_5$  con  $k$  caratteri letti, il prossimo carattere rende la lunghezza  $k + 1$ , che è dispari).
  - **Transizione (Fine  $W_i$  - pari):** Legge # o B da Nastro 1, riscrive,  $L$ . (Ritorna all'inizio di  $W_i$ ). Passa a  $Q_9$ . (La lunghezza di  $W_i$  è pari, quindi cerchiamo  $B$ ).
- **Stato  $Q_6$  (Cerca  $B^R$  in  $W_i$  - Lunghezza dispari):**
  - Si muove non deterministicamente su Nastro 1 (leggendo  $\alpha$ , riscrivendo,  $L$ ) per trovare l'inizio della  $B^R$  in  $W_i$ .
  - Quando decide di iniziare il confronto: Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $L$ . Passa a  $Q_7$ . (Nastro 3 ha  $B$ , leggendolo  $L$  si legge  $B^R$ ).
- **Stato  $Q_7$  (Confronto  $B^R$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $L$ .
  - Loop in  $Q_7$ .
  - Quando Nastro 3 legge B (fine di  $B^R$ ): Nastro 1 continua a leggere  $\alpha$ , riscrive  $\alpha$ ,  $R$ . Passa a  $Q_8$ .
- **Stato  $Q_8$  (Pulizia dopo  $B^R$ ):**
  - Riavvolge Nastro 1 a destra fino al prossimo # o B. Riavvolge Nastro 3 a destra fino al B iniziale.
  - Quando # o B su Nastro 1 e B su Nastro 3 sono raggiunti, passa a  $Q_{loop\_check}$  (stato intermedio per decidere se continuare o accettare).
- **Stato  $Q_9$  (Cerca  $B$  in  $W_i$  - Lunghezza pari):**
  - Si muove non deterministicamente su Nastro 1 (leggendo  $\alpha$ , riscrivendo,  $R$ ) per trovare l'inizio della  $B$  in  $W_i$ .
  - Quando decide di iniziare il confronto: Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $R$ . Passa a  $Q_{10}$ . (Nastro 3 ha  $B$ , leggendolo  $R$  si legge  $B$ ).
- **Stato  $Q_{10}$  (Confronto  $B$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $R$ .
  - Loop in  $Q_{10}$ .

- Quando Nastro 3 legge B (fine di B): Nastro 1 continua a leggere  $\alpha$ , riscrive  $\alpha$ , R. Passa a  $Q_{11}$ .
- **Stato  $Q_{11}$  (Pulizia dopo B):**
  - Riavvolge Nastro 1 a destra fino al prossimo # o B. Riavvolge Nastro 3 a destra fino al B iniziale.
  - Quando # o B su Nastro 1 e B su Nastro 3 sono raggiunti, passa a  $Q_{loop\_check}$ .
- **Stato  $Q_{loop\_check}$  (Controllo N e Ciclo/Accetta):**
  - Se Nastro 2 ha ancora X (altre  $W_i$ ):
    - \* Nastro 1: legge # o B, riscrive, R.
    - \* Nastro 2: legge X, riscrive X, S. (Indica che ci sono ancora  $W_i$  da processare, la cancellazione è stata fatta in  $Q_3$ ).
    - \* Nastro 3: riavvolge.
    - \* Torna a  $Q_3$ .
  - Se Nastro 2 è B (tutte le  $W_i$  sono state processate):
    - \* Nastro 1: legge B, riscrive B, S.
    - \* Accetta ( $Q_{acc}$ ).

## 2.4 Esercizio 4: $A$ o $B^R$ in $W_i$ in base alla Parità dell'Indice $i$

**Definizione 4** (Linguaggio  $L_4$ ). Sia  $L_4$  il linguaggio definito come:  $L_4 = \{A\#B\#W_1\#W_2\#\dots\#W_N \mid A, B, W_i \in \{a, b, c, d\}^+, |A| > |B| > 0, N = |A| + |B|, \text{ (se } i \text{ è dispari, allora } A \subseteq W_i) \wedge \text{(se } i \text{ è pari, allora } B^R \subseteq W_i)\}$

### 2.4.1 Strategia della Macchina di Turing

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Conteggio  $N$  e Parità Indice):** Memorizza  $N = |A| + |B|$ . Per ogni  $W_i$  processata, un X viene cancellato, permettendo al nastro di fungere anche da contatore di indice  $i$  (implicitamente).
- **Nastro 3 (Copia A):** Memorizza la stringa A.
- **Nastro 4 (Copia B):** Memorizza la stringa B. Per  $B^R$ , Nastro 4 verrà letto all'indietro.

### 2.4.2 Descrizione degli Stati e delle Transizioni

Sia  $\Sigma_I = \{a, b, c, d, \#\}$  l'alfabeto di input e  $\Gamma_T = \Sigma_I \cup \{B, X\}$  l'alfabeto del nastro. Usiamo  $\alpha$  per un simbolo generico da  $\{a, b, c, d\}$ .

- **Stato  $Q_0$  (Copia A e  $|A|$  su Nastro 2):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ , R. Scrive X su Nastro 2, R. (Copia A come Xs su Nastro 2). Scrive  $\alpha$  su Nastro 3, R. (Copia A su Nastro 3).
  - Alla lettura di #: riscrive #, R. Riavvolge Nastro 3 (L). Passa a  $Q_1$ .

- **Stato  $Q_1$  (Copia  $B$  e Calcolo  $|A| + |B|$  su Nastro 2):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Scrive  $X$  su Nastro 2,  $R$ . (Aggiunge  $|B|$  a Nastro 2). Scrive  $\alpha$  su Nastro 4,  $R$ . (Copia  $B$  su Nastro 4).
  - Alla lettura di #: riscrive #,  $R$ . Riavvolge Nastro 4 ( $L$ ).
  - Controlla  $|A| > |B| > 0$ : questo avviene verificando che Nastro 2 ha almeno due  $X$ s dopo aver contato  $A$  e  $B$  e che Nastro 4 ha almeno un  $X$ . Implicitamente, si assicura che  $B$  non sia vuoto e che  $A$  sia più lungo di  $B$ .
  - Passa a  $Q_2$ .
- **Stato  $Q_2$  (Preparazione e Inizio Loop  $W_i$ ):**
  - Riavvolge Nastro 2 a sinistra per posizionarsi sul primo  $X$  da consumare. Riavvolge Nastro 3 all'inizio di  $A$ . Riavvolge Nastro 4 all'inizio di  $B$ .
  - Passa a  $Q_3$ .
- **Stato  $Q_3$  (Processa  $W_i$  - indice dispari):** Questo stato gestisce  $W_i$  con  $i$  dispari (es.  $W_1, W_3, \dots$ ).
  - Su Nastro 2: legge  $X$ , lo cancella ( $B$ ),  $L$ . (Decrementa  $N$  per questa  $W_i$ ).
  - Nondeterministicamente cerca l'inizio di  $A$  in  $W_i$  su Nastro 1 (legge  $\alpha$ , riscrive,  $R$ ).
  - Quando decide di iniziare il confronto: Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $R$ . Passa a  $Q_4$ .
- **Stato  $Q_4$  (Confronto  $A$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 3, riscrive  $\alpha$ ,  $R$ .
  - Loop in  $Q_4$ .
  - Quando Nastro 3 è  $B$  (fine di  $A$ ): Nastro 1 continua a leggere  $\alpha$ , riscrive  $\alpha$ ,  $R$ . Passa a  $Q_5$ .
- **Stato  $Q_5$  (Pulizia dopo  $A$  e Preparazione per  $W_{i+1}$ ):**
  - Riavvolge Nastro 3 a sinistra al  $B$  iniziale.
  - Riavvolge Nastro 1 a destra fino al prossimo # o  $B$ .
  - Quando # su Nastro 1 è raggiunto e Nastro 3 è riavvolto: Legge #, riscrive #,  $R$ . Passa a  $Q_6$ .
- **Stato  $Q_6$  (Processa  $W_i$  - indice pari):** Questo stato gestisce  $W_i$  con  $i$  pari (es.  $W_2, W_4, \dots$ ).
  - Su Nastro 2: legge  $X$ , lo cancella ( $B$ ),  $L$ . (Decrementa  $N$  per questa  $W_i$ ).
  - Nondeterministicamente cerca l'inizio di  $B^R$  in  $W_i$  su Nastro 1 (legge  $\alpha$ , riscrive,  $L$  - posizionandosi per il reverse check).
  - Quando decide di iniziare il confronto: Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 4, riscrive  $\alpha$ ,  $L$ . Passa a  $Q_7$ .
- **Stato  $Q_7$  (Confronto  $B^R$ ):**
  - Legge  $\alpha$  da Nastro 1, riscrive  $\alpha$ ,  $R$ . Legge  $\alpha$  da Nastro 4, riscrive  $\alpha$ ,  $L$ .
  - Loop in  $Q_7$ .

- Quando Nastro 4 è B (fine di  $B^R$ ): Nastro 1 continua a leggere  $\alpha$ , riscrive  $\alpha$ ,  $R$ . Passa a  $Q_8$ .
- **Stato  $Q_8$  (Pulizia dopo  $B^R$  e Preparazione per  $W_{i+1}$ ):**
  - Riavvolge Nastro 4 a sinistra al B iniziale.
  - Riavvolge Nastro 1 a destra fino al prossimo # o B.
  - Quando # su Nastro 1 è raggiunto e Nastro 4 è riavvolto: Legge #, riscrive #,  $R$ . Passa a  $Q_3$  (per la prossima  $W_i$  dispari).
- **Accettazione ( $Q_{acc}$ ):**
  - Dopo l'elaborazione dell'ultima  $W_N$ , se Nastro 2 legge B (tutti gli X sono stati consumati) e Nastro 1 legge B (fine input):
    - \*  $(B, B, any, any) \rightarrow (B, B, any, any), (S, S, S, S)$  e accetta.

### 3 Conclusioni

Le esercitazioni di oggi hanno permesso di approfondire la progettazione di Macchine di Turing Non Deterministiche, in particolare mostrando come la capacità di guess (indovinare) combinata con un robusto check (controllo) possa semplificare la logica di alcune verifiche complesse. È stato evidenziato come le NDTM possano scrivere in anticipo su nastri ausiliari stringhe che verranno poi validate. Nella prossima lezione, si inizierà a esplorare i concetti di calcolabilità, inclusi problemi indecidibili e il concetto fondamentale di riduzione, che sarà un pilastro per comprendere la complessità dei problemi.