

# Lezione di Informatica Teorica

## Macchine di Turing Multinastro

Appunti da Trascrizione Automatica

30 giugno 2025

### Indice

<b>1</b>	<b>Introduzione alle Macchine di Turing Multinastro</b>	<b>2</b>
1.1	Definizione e Equivalenza . . . . .	2
1.2	Convenzioni di Notazione nelle Transizioni . . . . .	2
<b>2</b>	<b>Esercizi di Riconoscimento di Linguaggi</b>	<b>3</b>
2.1	Linguaggio $L_1 = \{A\#B\#AB \mid A, B \in \{0, 1\}^+\}$ . . . . .	3
2.1.1	Strategia . . . . .	3
2.1.2	Implementazione (Stati e Transizioni) . . . . .	3
2.2	Linguaggio $L_2 = \{A\#B\#C \mid A, B, C \in \{0, 1\}^+,  A  >  B  >  C ,  C  =  A  -  B \}$ . . .	4
2.2.1	Strategia . . . . .	4
2.2.2	Implementazione (Stati e Transizioni) . . . . .	5
2.3	Linguaggio $L_3 = \{W\#W\#W \mid W \in \{0, 1\}^*\}$ . . . . .	5
2.3.1	Strategia . . . . .	6
2.3.2	Implementazione (Stati e Transizioni) . . . . .	6
2.4	Linguaggio $L_4 = \{A\#B \mid A, B \in \{a, b\}^*, A \subseteq B\}$ . . . . .	7
2.4.1	Strategia . . . . .	7
2.4.2	Implementazione (Stati e Transizioni) . . . . .	7
2.5	Linguaggio $L_5 = \{WW \mid W \in \{0, 1\}^+\}$ . . . . .	8
2.5.1	Strategia . . . . .	8
2.5.2	Implementazione (Stati e Transizioni) . . . . .	9
2.6	Esercizio Aggiuntivo per Casa . . . . .	10

# 1 Introduzione alle Macchine di Turing Multinastro

## 1.1 Definizione e Equivalenza

Le macchine di Turing (MT) multinastro sono estensioni del modello standard che possiedono più nastri, ciascuno con la propria testina indipendente. Questo permette di accedere a diverse parti della memoria contemporaneamente.

**Definizione 1** (Macchina di Turing Multinastro). *Una macchina di Turing multinastro è definita come una tupla  $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ , dove:*

- $Q$  è un insieme finito di stati.
- $\Sigma$  è l'alfabeto di input (non contiene il simbolo di blank  $\square$ ).
- $\Gamma$  è l'alfabeto del nastro (contiene  $\Sigma$  e  $\square$ ).
- $\delta : Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{L, R, S\})^k$  è la funzione di transizione, dove  $k$  è il numero di nastri. Per ogni stato e per ogni tupla di simboli letti dalle  $k$  testine, la funzione specifica il prossimo stato, i simboli da scrivere su ogni nastro e le direzioni in cui muovere ogni testina (Left, Right, Stay).
- $q_0 \in Q$  è lo stato iniziale.
- $q_{acc} \in Q$  è lo stato di accettazione.
- $q_{rej} \in Q$  è lo stato di rifiuto.

**Proposizione 1** (Equivalenza). *Le macchine di Turing multinastro sono equivalenti alle macchine di Turing standard (mononastro). Ciò significa che ogni linguaggio riconoscibile/decidibile da una MT multinastro è riconoscibile/decidibile anche da una MT standard, e viceversa.*

Sebbene non aumentino il potere computazionale (cioè, non possono risolvere più problemi), le MT multinastro possono essere più efficienti in termini di tempo o più facili da programmare, grazie alla capacità di gestire più flussi di dati in parallelo.

## 1.2 Convenzioni di Notazione nelle Transizioni

Nei diagrammi di stato (e nella loro descrizione testuale), le transizioni sono etichettate con una notazione compatta per rappresentare le operazioni su più nastri. Una transizione da uno stato  $q_i$  a  $q_j$  etichettata con, ad esempio:

nastro 1: (leggi<sub>1</sub>/scrivi<sub>1</sub>, muovi<sub>1</sub>),    nastro 2: (leggi<sub>2</sub>/scrivi<sub>2</sub>, muovi<sub>2</sub>)

significa che per eseguire la transizione, la MT deve leggere leggi<sub>1</sub> dal nastro 1 e leggi<sub>2</sub> dal nastro 2. Se le condizioni di lettura sono soddisfatte, la MT scrive scrivi<sub>1</sub> su nastro 1 e scrivi<sub>2</sub> su nastro 2, e muove le testine come indicato.

- **Simboli Variabili:** Spesso si usano simboli come  $\alpha, \beta, \gamma$  per indicare caratteri generici dall'alfabeto del nastro ( $\Gamma$ ).

- Se lo **stesso simbolo variabile** (es.  $\alpha$ ) compare nella lettura di **più nastri** nella stessa etichetta di transizione, significa che su quei nastri deve essere letto lo **stesso identico carattere**. Es. nastro 1:  $(\alpha/\alpha, R)$ , nastro 2:  $(\alpha/\alpha, R)$  implica che se si legge '0' su nastro 1, si deve leggere '0' anche su nastro 2. Se si legge '1' su nastro 1, si deve leggere '1' anche su nastro 2. Se si legge '0' su nastro 1 e '1' su nastro 2, questa transizione non è valida.
- Se **simboli variabili diversi** (es.  $\alpha, \beta$ ) compaiono nella lettura di **più nastri** nella stessa etichetta di transizione, significa che i caratteri letti possono essere **diversi o uguali**. Es. nastro 1:  $(\alpha/\alpha, R)$ , nastro 2:  $(\beta/\beta, R)$  implica che  $\alpha$  può essere '0' e  $\beta$  '1' (o viceversa, o uguali). Non viene imposto alcun confronto.
- **Omissione di operazioni:** Se per un nastro non viene specificata alcuna operazione di scrittura o movimento, si intende che la testina rimane ferma (S) e il simbolo non viene modificato.
- **Simbolo di blank:** Il simbolo  $\square$  indica una cella vuota.

## 2 Esercizi di Riconoscimento di Linguaggi

Illustriamo il funzionamento delle MT multinastro attraverso esempi pratici di riconoscimento di linguaggi.

### 2.1 Linguaggio $L_1 = \{A\#B\#AB \mid A, B \in \{0, 1\}^+\}$

**Definizione 2.** Il linguaggio  $L_1$  è l'insieme di tutte le stringhe composte da tre parti  $A$ ,  $B$  e  $AB$  (concatenazione di  $A$  e  $B$ ), separate da simboli  $\#$ . Le stringhe  $A$  e  $B$  sono composte da caratteri '0' o '1' e devono avere lunghezza di almeno 1.

**Esempio 1.** Esempi di stringhe in  $L_1$ :  $0\#1\#01$ ,  $01\#0\#010$ . Esempi di stringhe non in  $L_1$ :  $0\#1\#10$  ( $AB$  sbagliato),  $0\#\#0$  ( $B$  vuoto),  $\#1\#1$  ( $A$  vuoto).

#### 2.1.1 Strategia

Utilizziamo una MT con 2 nastri.

- **Nastro 1 (Input):** Contiene la stringa di input  $A\#B\#AB$ .
- **Nastro 2 (Lavoro):** Utilizzato per copiare la stringa  $A$  e poi  $B$  sequenzialmente, ottenendo  $AB$ . Successivamente, il contenuto di Nastro 2 (che è  $AB$ ) verrà confrontato con l'ultima parte della stringa sul Nastro 1 (che dovrebbe essere  $AB$ ).

#### 2.1.2 Implementazione (Stati e Transizioni)

$q_0$  (**Copia A**): Legge i caratteri di  $A$  dal Nastro 1 e li copia sul Nastro 2.

- **Condizione:** Leggi  $\alpha \in \{0, 1\}$  su Nastro 1,  $\square$  su Nastro 2.
- **Azione:** Scrivi  $\alpha$  su Nastro 2. Sposta entrambe le testine a destra. Rimani in  $q_0$ .
- **Uscita:** Quando Nastro 1 legge  $\#$ .
- **Transizione:** Da  $q_0$  a  $q_1$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(\square/\square, S)$ .

$q_1$  (**Copia B**): Legge i caratteri di  $B$  dal Nastro 1 e li copia sul Nastro 2 (dopo la copia di  $A$ ).

- **Condizione:** Leggi  $\alpha \in \{0, 1\}$  su Nastro 1,  $\square$  su Nastro 2.
- **Azione:** Scrivi  $\alpha$  su Nastro 2. Sposta entrambe le testine a destra. Rimani in  $q_1$ .
- **Uscita:** Quando Nastro 1 legge  $\#$ .
- **Transizione:** Da  $q_1$  a  $q_2$ : Nastro 1: ( $\#/\#, R$ ), Nastro 2: ( $\square/\square, S$ ).

$q_2$  (**Riavvolgi Nastro 2**): Riporta la testina del Nastro 2 all'inizio della stringa  $AB$  (cioè all'inizio del Nastro 2).

- **Condizione:** Leggi  $\alpha \in \{0, 1\}$  su Nastro 2.
- **Azione:** Scrivi  $\alpha$  su Nastro 2. Sposta testina Nastro 2 a sinistra. Rimani in  $q_2$ .
- **Uscita:** Quando Nastro 2 legge  $\square$ .
- **Transizione:** Da  $q_2$  a  $q_3$ : Nastro 2: ( $\square/\square, R$ ). (Nastro 1 rimane fermo).

$q_3$  (**Confronto AB**): Confronta la stringa  $AB$  sul Nastro 2 con l'ultima parte della stringa sul Nastro 1.

- **Condizione:** Leggi  $\alpha \in \{0, 1\}$  su Nastro 1,  $\alpha$  su Nastro 2. (Stesso  $\alpha$  indica che devono essere uguali).
- **Azione:** Scrivi  $\alpha$  su Nastro 1 e  $\alpha$  su Nastro 2. Sposta entrambe le testine a destra. Rimani in  $q_3$ .
- **Uscita:** Quando entrambe le testine leggono  $\square$ .
- **Transizione:** Da  $q_3$  a  $q_{acc}$ : Nastro 1: ( $\square/\square, S$ ), Nastro 2: ( $\square/\square, S$ ).

**Rifiuto:** Se in qualsiasi stato la MT tenta una transizione non definita (es. simboli non corrispondenti durante il confronto,  $\#$  al posto di un carattere, stringa non conforme al formato  $A\#B\#AB$ ), la macchina si blocca e rifiuta.

## 2.2 Linguaggio $L_2 = \{A\#B\#C \mid A, B, C \in \{0, 1\}^+, |A| > |B| > |C|, |C| = |A| - |B|\}$

**Definizione 3.** Il linguaggio  $L_2$  consiste in stringhe  $A\#B\#C$  dove  $A, B, C$  sono stringhe non vuote di '0' o '1'. Devono valere le condizioni sulle lunghezze:

- $|A| > |B| > |C|$  (lunghezze strettamente decrescenti)
- $|C| = |A| - |B|$

### 2.2.1 Strategia

Utilizziamo 3 nastri.

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Lavoro):** Usato per marcare la lunghezza di  $A$  (con 'X') e poi sottrarre la lunghezza di  $B$ . Al termine, conterrà  $|A| - |B|$ .
- **Nastro 3 (Lavoro):** Usato per marcare la lunghezza di  $B$  (con 'X').

### 2.2.2 Implementazione (Stati e Transizioni)

$q_0$  (**Copia  $|A|$  su Nastro 2**): Legge i caratteri di  $A$  dal Nastro 1 e marca 'X' sul Nastro 2 per ogni carattere letto.

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\square/X, R)$ . Rimani in  $q_0$ .
- **Uscita:** Quando Nastro 1 legge #.
- **Transizione:** Da  $q_0$  a  $q_1$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(X/X, L)$  (sposta indietro per iniziare la **sottrazione**).

$q_1$  (**Copia  $|B|$  su Nastro 3 e calcola  $|A| - |B|$** ): Legge i caratteri di  $B$  dal Nastro 1, marca 'X' sul Nastro 3 per ogni carattere letto, e cancella una 'X' dal Nastro 2 per ogni carattere letto (simulando  $|A| - |B|$ ).

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $X$ , Nastro 3:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(X/\square, L)$ , Nastro 3:  $(\square/X, R)$ . Rimani in  $q_1$ .
- **Controllo  $|A| > |B|$ :** Se Nastro 2 diventa  $\square$  prima che Nastro 1 trovi #, significa che  $|A| \leq |B|$ , quindi la condizione  $|A| > |B|$  non è rispettata. La MT si blocca e rifiuta. Altrimenti, se Nastro 1 trova # e Nastro 2 ha ancora X, la condizione è rispettata.
- **Uscita:** Quando Nastro 1 legge #.
- **Transizione:** Da  $q_1$  a  $q_2$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(X/X, S)$ , Nastro 3:  $(\square/\square, L)$ . (Nastro 2 rimane fermo sulla X residua, Nastro 3 si riavvolge per il confronto con C).

$q_2$  (**Confronto  $|C|$  con  $|A| - |B|$  e  $|B| > |C|$** ): Legge i caratteri di  $C$  dal Nastro 1, cancella una 'X' dal Nastro 2 (verificando  $|C| = |A| - |B|$ ), e cancella una 'X' dal Nastro 3 (verificando  $|B| > |C|$ ).

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $X$ , Nastro 3:  $X$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(X/\square, L)$ , Nastro 3:  $(X/\square, L)$ . Rimani in  $q_2$ .
- **Controllo  $|C| = |A| - |B|$ :** Se Nastro 2 diventa  $\square$  e Nastro 1 non è finito, significa  $|C| > |A| - |B|$ . Se Nastro 1 finisce e Nastro 2 ha ancora X, significa  $|C| < |A| - |B|$ . In entrambi i casi, rifiuta. Se Nastro 1 e Nastro 2 finiscono assieme (Nastro 2 legge  $\square$  quando Nastro 1 legge  $\square$ ), allora  $|C| = |A| - |B|$ .
- **Controllo  $|B| > |C|$ :** Durante il processo, Nastro 3 deve ancora avere 'X' quando Nastro 1 finisce (cioè Nastro 3 deve diventare  $\square$  \*dopo\* Nastro 1). Se Nastro 3 diventa  $\square$  prima o insieme a Nastro 1, significa  $|B| \leq |C|$ , quindi rifiuta.
- **Uscita:** Quando Nastro 1 legge  $\square$ .
- **Transizione:** Da  $q_2$  a  $q_{acc}$ : Nastro 1:  $(\square/\square, S)$ , Nastro 2:  $(\square/\square, S)$ , Nastro 3:  $(X/X, S)$  (Nastro 3 deve ancora avere X per  $|B| > |C|$ ).

**Rifiuto:** Tutte le condizioni non soddisfatte portano al rifiuto (es. testina Nastro 2 su  $\square$  troppo presto, testina Nastro 3 su  $\square$  troppo presto, etc.).

### 2.3 Linguaggio $L_3 = \{W\#W\#W \mid W \in \{0, 1\}^*\}$

**Definizione 4.** Il linguaggio  $L_3$  consiste in stringhe  $W\#W\#W$ , dove  $W$  è una stringa (anche vuota) di '0' o '1'. Le tre occorrenze di  $W$  devono essere identiche.

**Esempio 2.** Esempi: 01#01#01, ## (per  $W = \epsilon$ ).

### 2.3.1 Strategia

Utilizziamo 3 nastri.

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Lavoro):** Copia la prima  $W$ .
- **Nastro 3 (Lavoro):** Copia la prima  $W$ .

Dopo aver copiato la prima  $W$ , le testine di Nastro 2 e 3 vengono riavvolte. Successivamente, la seconda  $W$  sul Nastro 1 viene confrontata con la copia su Nastro 2. Infine, la terza  $W$  sul Nastro 1 viene confrontata con la copia su Nastro 3.

### 2.3.2 Implementazione (Stati e Transizioni)

$q_0$  (**Copia  $W_1$  su Nastro 2 e 3**): Legge i caratteri della prima  $W$  dal Nastro 1 e li copia su Nastro 2 e Nastro 3.

- **Condizione:** Nastro 1:  $\alpha \in \{0,1\}$ , Nastro 2:  $\square$ , Nastro 3:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\square/\alpha, R)$ , Nastro 3:  $(\square/\alpha, R)$ . Rimani in  $q_0$ .
- **Uscita:** Quando Nastro 1 legge #.
- **Transizione:** Da  $q_0$  a  $q_1$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(\square/\square, L)$ , Nastro 3:  $(\square/\square, L)$ . (Per  $W = \epsilon$ , la MT va direttamente a  $q_1$  leggendo # sul Nastro 1 e  $\square$  su Nastro 2 e 3. Le testine 2 e 3 si muovono a sinistra sul  $\square$  e poi, nel successivo riavvolgimento, si riposizionano correttamente).

$q_1$  (**Riavvolgi Nastro 2 e 3**): Riporta le testine di Nastro 2 e 3 all'inizio delle loro rispettive stringhe  $W$ .

- **Condizione:** Nastro 2:  $\alpha \in \{0,1\}$ , Nastro 3:  $\alpha \in \{0,1\}$ .
- **Azione:** Nastro 2:  $(\alpha/\alpha, L)$ , Nastro 3:  $(\alpha/\alpha, L)$ . Rimani in  $q_1$ .
- **Uscita:** Quando Nastro 2 e Nastro 3 leggono  $\square$ .
- **Transizione:** Da  $q_1$  a  $q_2$ : Nastro 2:  $(\square/\square, R)$ , Nastro 3:  $(\square/\square, R)$ . (Nastro 1 rimane fermo).

$q_2$  (**Confronta  $W_2$  con Nastro 2**): Confronta la seconda  $W$  sul Nastro 1 con la copia su Nastro 2.

- **Condizione:** Nastro 1:  $\alpha \in \{0,1\}$ , Nastro 2:  $\alpha \in \{0,1\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\alpha/\alpha, R)$ . Rimani in  $q_2$ .
- **Uscita:** Quando Nastro 1 legge # e Nastro 2 legge  $\square$ .
- **Transizione:** Da  $q_2$  a  $q_3$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(\square/\square, S)$ .

$q_3$  (**Confronta  $W_3$  con Nastro 3**): Confronta la terza  $W$  sul Nastro 1 con la copia su Nastro 3.

- **Condizione:** Nastro 1:  $\alpha \in \{0,1\}$ , Nastro 3:  $\alpha \in \{0,1\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 3:  $(\alpha/\alpha, R)$ . Rimani in  $q_3$ .
- **Uscita:** Quando Nastro 1 e Nastro 3 leggono  $\square$ .
- **Transizione:** Da  $q_3$  a  $q_{acc}$ : Nastro 1:  $(\square/\square, S)$ , Nastro 3:  $(\square/\square, S)$ .

**Rifiuto:** Qualsiasi discrepanza tra i simboli durante i confronti o formato non corretto della stringa porta al rifiuto.

## 2.4 Linguaggio $L_4 = \{A\#B \mid A, B \in \{a, b\}^*, A \subseteq B\}$

**Definizione 5.** Il linguaggio  $L_4$  è l'insieme di stringhe  $A\#B$  dove  $A$  e  $B$  sono stringhe sull'alfabeto  $\{a, b\}$ , e  $A$  è una sottostringa di  $B$ . La stringa vuota ( $\epsilon$ ) è una sottostringa di ogni stringa.

**Esempio 3.** Esempi:  $a\#bab$ ,  $ab\#ababa$ ,  $\#ab$  ( $A$  è vuota). Esempi non in  $L_4$ :  $ab\#ba$ ,  $a\#b$ .

### 2.4.1 Strategia

Questa è più complessa a causa della natura della sottostringa, che può apparire in qualsiasi posizione all'interno di  $B$ .

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Lavoro):** Copia la stringa  $A$ .

La strategia consiste nel provare a far corrispondere la stringa  $A$  (copiata sul Nastro 2) con ogni possibile sottostringa di  $B$  sul Nastro 1. Se un tentativo fallisce, si sposta la posizione di partenza su  $B$  di un carattere e si riprova.

### 2.4.2 Implementazione (Stati e Transizioni)

$q_0$  (**Copia A**): Legge i caratteri di  $A$  dal Nastro 1 e li copia sul Nastro 2.

- **Condizione:** Nastro 1:  $\alpha \in \{a, b\}$ , Nastro 2:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\square/\alpha, R)$ . Rimani in  $q_0$ .
- **Uscita:** Quando Nastro 1 legge  $\#$ .
- **Transizione:** Da  $q_0$  a  $q_1$ : Nastro 1:  $(\#/ \#, R)$ , Nastro 2:  $(\square/\square, L)$ .

$q_1$  (**Riavvolgi Nastro 2**): Riporta la testina del Nastro 2 all'inizio della stringa  $A$ .

- **Condizione:** Nastro 2:  $\alpha \in \{a, b\}$ .
- **Azione:** Nastro 2:  $(\alpha/\alpha, L)$ . Rimani in  $q_1$ .
- **Uscita:** Quando Nastro 2 legge  $\square$ .
- **Transizione:** Da  $q_1$  a  $q_2$ : Nastro 2:  $(\square/\square, R)$ . (Nastro 1 rimane fermo).

$q_2$  (**Tentativo di Match Iniziale**): Questo stato è il punto di partenza per ogni tentativo di confronto di  $A$  con una sottostringa di  $B$ . Gestisce anche il caso di  $A = \epsilon$ .

- **Condizione ( $A$  è  $\epsilon$ ):** Nastro 2:  $\square$ .
- **Azione:** Nastro 2:  $(\square/\square, S)$ . Spostati a  $q_5$  e accetta. (Se  $A$  è vuota, è sottostringa di qualsiasi  $B$ ).
- **Condizione (Inizio Match):** Nastro 1:  $\alpha \in \{a, b\}$ , Nastro 2:  $\alpha \in \{a, b\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\alpha/\alpha, R)$ . Spostati a  $q_3$ . (Qui inizia il vero confronto).
- **Condizione (Nastro 1 finisce, A non vuota):** Nastro 1:  $\square$ , Nastro 2:  $\alpha \in \{a, b\}$ . (Se  $B$  finisce ma  $A$  non è stata trovata)
- **Azione:** Rifiuta. (Non c'è una transizione definita per questo caso, quindi la MT si blocca e rifiuta).

$q_3$  (**Match di A**): Continua il confronto tra  $A$  (Nastro 2) e la sottostringa di  $B$  (Nastro 1).

- **Condizione (Match):** Nastro 1:  $\alpha \in \{a, b\}$ , Nastro 2:  $\alpha \in \{a, b\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\alpha/\alpha, R)$ . Rimani in  $q_3$ .
- **Uscita (Match di A completato):** Nastro 2 legge  $\square$ .
- **Transizione:** Da  $q_3$  a  $q_4$ : Nastro 1:  $(\alpha/\alpha, S)$ , Nastro 2:  $(\square/\square, S)$ . (Nastro 1 è posizionato dopo la sottostringa, ora controlliamo il resto di B).
- **Uscita (Mismatch):** Nastro 1:  $\alpha \in \{a, b\}$ , Nastro 2:  $\beta \in \{a, b\}$ , con  $\alpha \neq \beta$ .
- **Transizione:** Da  $q_3$  a  $q_6$ : Nastro 1:  $(\alpha/\alpha, L)$ , Nastro 2:  $(\beta/\beta, L)$ . (La testina del nastro 1 torna indietro di 1, poi entrambe si riavvolgeranno al punto di inizio per il prossimo tentativo).

$q_4$  (**Verifica fine di B**): Dopo aver trovato  $A$  come sottostringa, si assicura che non ci siano simboli  $\#$  o altro sul resto di  $B$ .

- **Condizione:** Nastro 1:  $\alpha \in \{a, b\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ . Rimani in  $q_4$ .
- **Uscita:** Nastro 1 legge  $\square$ .
- **Transizione:** Da  $q_4$  a  $q_5$ : Nastro 1:  $(\square/\square, S)$ .

$q_5$  (**Accetta**): Stato di accettazione.

$q_6$  (**Riavvolgi Nastro 1 e 2 dopo Mismatch**): Riporta Nastro 1 e Nastro 2 all'inizio della stringa  $B$  e  $A$  rispettivamente.

- **Condizione:** Nastro 1:  $\gamma \in \{a, b, \#\}$ , Nastro 2:  $\gamma \in \{a, b, \square\}$ .
- **Azione:** Nastro 1:  $(\gamma/\gamma, L)$ , Nastro 2:  $(\gamma/\gamma, L)$ . Rimani in  $q_6$ . (Questo porta entrambe le testine indietro fino all'inizio di  $A$  e all'inizio di  $B$ ).
- **Uscita:** Nastro 1 legge  $\#$ .
- **Transizione:** Da  $q_6$  a  $q_2$ : Nastro 1:  $(\#/\#, R)$ , Nastro 2:  $(\square/\square, R)$ . (La testina del Nastro 1 si sposta a destra di un posto, la testina del Nastro 2 si posiziona all'inizio di  $A$ . Ora Nastro 1 è pronto per un nuovo tentativo di match in  $B$  un carattere più avanti).

## 2.5 Linguaggio $L_5 = \{WW \mid W \in \{0, 1\}^+\}$

**Definizione 6.** Il linguaggio  $L_5$  è l'insieme di stringhe che sono la concatenazione di una stringa  $W$  con sé stessa, dove  $W$  è non vuota e composta da '0' o '1'.

**Esempio 4.** Esempi: 00, 0101, 1111. Esempi non in  $L_5$ : 0110 (non  $WW$ ), 0 (lunghezza dispari),  $\epsilon$  ( $W$  non vuota).

### 2.5.1 Strategia

Il problema principale è identificare il punto centrale della stringa senza un delimitatore.

- **Nastro 1 (Input):** Contiene la stringa di input.
- **Nastro 2 (Lavoro):** Utilizzato per determinare la lunghezza di  $W$ . Si marca un carattere sì e uno no per trovare la metà. Se la lunghezza totale è dispari, rifiuta.
- **Nastro 3 (Lavoro):** Copia la prima metà (il primo  $W$ ).

## 2.5.2 Implementazione (Stati e Transizioni)

$q_0$  (**Determina metà lunghezza**): Scorre la stringa di input, scrivendo una 'X' sul Nastro 2 ogni due caratteri del Nastro 1. Questo identifica la metà della stringa.

- **Condizione (Pari):** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\square/X, R)$ . Spostati a  $q_1$ . (Marca il primo carattere di una coppia).
- **Uscita (Nastro 1 finito, dispari):** Se Nastro 1 legge  $\square$  mentre Nastro 2 legge  $\square$ , significa che la stringa era di lunghezza dispari. Rifiuta. (Non c'è transizione definita per Nastro 1:  $\square$ , Nastro 2:  $\square$ ).

$q_1$  (**Salta un carattere**): Salta un carattere sul Nastro 1 senza scrivere sul Nastro 2.

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(\square/\square, R)$ . Rimani in  $q_0$ . (Va a marcare il prossimo carattere della coppia, o si ferma se la stringa finisce).
- **Uscita (Nastro 1 finito, pari):** Se Nastro 1 legge  $\square$  e Nastro 2 legge  $X$ , significa che la stringa era di lunghezza pari.
- **Transizione:** Da  $q_1$  a  $q_2$ : Nastro 1:  $(\square/\square, S)$ , Nastro 2:  $(X/X, S)$ . (Nastro 1 si è fermato alla fine, Nastro 2 sulla sua ultima X).

$q_2$  (**Riavvolgi Nastro 1 e 2**): Riporta le testine del Nastro 1 e 2 all'inizio delle stringhe.

- **Condizione (Nastro 2):** Nastro 2:  $X$ .
- **Azione:** Nastro 2:  $(X/X, L)$ . Rimani in  $q_2$ .
- **Uscita (Nastro 2 finito):** Nastro 2 legge  $\square$ .
- **Transizione:** Da  $q_2$  a  $q_3$ : Nastro 2:  $(\square/\square, R)$ .
- **Condizione (Nastro 1):** Nastro 1:  $\alpha \in \{0, 1\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, L)$ . Rimani in  $q_3$ .
- **Uscita (Nastro 1 finito):** Nastro 1 legge  $\square$ .
- **Transizione:** Da  $q_3$  a  $q_4$ : Nastro 1:  $(\square/\square, R)$ .

$q_4$  (**Copia  $W_1$  su Nastro 3**): Copia la prima metà della stringa (i primi  $|W|$  caratteri) dal Nastro 1 al Nastro 3, cancellando le 'X' dal Nastro 2.

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 2:  $X$ , Nastro 3:  $\square$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 2:  $(X/\square, R)$ , Nastro 3:  $(\square/\alpha, R)$ . Rimani in  $q_4$ .
- **Uscita:** Nastro 2 legge  $\square$ . (Abbiamo copiato esattamente metà della stringa).
- **Transizione:** Da  $q_4$  a  $q_5$ : Nastro 2:  $(\square/\square, S)$ .

$q_5$  (**Riavvolgi Nastro 3**): Riporta la testina del Nastro 3 all'inizio della stringa  $W_1$ .

- **Condizione:** Nastro 3:  $\alpha \in \{0, 1\}$ .
- **Azione:** Nastro 3:  $(\alpha/\alpha, L)$ . Rimani in  $q_5$ .
- **Uscita:** Nastro 3 legge  $\square$ .

- **Transizione:** Da  $q_5$  a  $q_6$ : Nastro 3:  $(\square/\square, R)$ .

$q_6$  (**Confronta  $W_2$  con Nastro 3**): Confronta la seconda metà della stringa sul Nastro 1 con la copia della prima metà sul Nastro 3.

- **Condizione:** Nastro 1:  $\alpha \in \{0, 1\}$ , Nastro 3:  $\alpha \in \{0, 1\}$ .
- **Azione:** Nastro 1:  $(\alpha/\alpha, R)$ , Nastro 3:  $(\alpha/\alpha, R)$ . Rimani in  $q_6$ .
- **Uscita:** Nastro 1 e Nastro 3 leggono  $\square$ .
- **Transizione:** Da  $q_6$  a  $q_{acc}$ : Nastro 1:  $(\square/\square, S)$ , Nastro 3:  $(\square/\square, S)$ .

**Rifiuto:** Qualsiasi incongruenza (lunghezza dispari, simboli non corrispondenti durante il confronto) porta al rifiuto.

## 2.6 Esercizio Aggiuntivo per Casa

**Definizione 7.** Sia  $L_6 = \{W\#A \mid W, A \in \{0, 1\}^+, A = W \text{ OR } A = W^R\}$

Il linguaggio  $L_6$  include stringhe  $W\#A$  dove  $W$  e  $A$  sono stringhe non vuote di '0' o '1'. La stringa  $A$  deve essere identica a  $W$  oppure essere la sua inversa ( $W^R$ ).