Calcolabilità

Mattia Patetta

14 novembre 2025

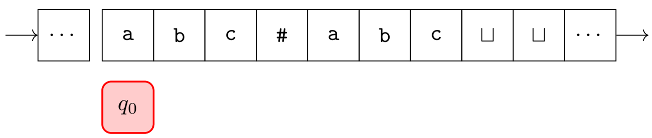
Indice

1 Macchine di Turing

Le Turing Machine corrisponde ad un modello astratto del computer. Utilizza un **nastro** di lunghezza infinita come memoria e una **testina** per poter leggere e scrivere simboli spostandola sia a destra che a sinistra. Inoltre è dotata di uno stato di **accettazione/rifiuto con effetto immediato**, cioè che lo stato *accetta/rifiuta* appena ci si entra.

Esempio 1.

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



Dove ⊔ rappresenta uno spazio vuoto.

1.1 Definizione formale di macchina di Turing

Definizione 1: Macchina di Turing

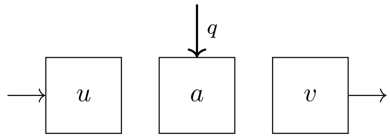
Una macchina di Turing è una 7-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{acc}}, q_{\text{rej}})$ dove:

- Q è l'insieme degli stati,
- Σ l'alfabeto di input,
- Γ l'alfabeto del nastro,
- δ la funzione di transizione definita come:

  $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$, con L ed R che indicano **left** e **right**

- $q_{start}, q_{acc}, q_{rej} \in Q$, sono rispettivamente gli stati: iniziali, accettanti e di rifiuto, con la caratteristica che $q_{start} \neq q_{acc}$.

Durante la computazione di una macchina di Turing, si verificano cambiamenti rispetto a: **stato**, **contenuto nastro** e **posizione testina**, ed l'impostazione di questi tre elementi è chiamata una **configuarazione** della macchina di Turing. Introduciamo in merito a queste configurazioni, la seguente notazione:



$uqav$, dove $u, v \in \Gamma^*, q \in Q, a \in \Gamma$.

In cui:

- $q \in Q$, rappresenta lo stato corrente,
- $u, v \in \Gamma^*$, sono stringhe nonché il contenuto corrente del nastro,
- $a \in \Gamma$, rappresenta la posizione attuale della testina.

Nota!

Dopo l'ultimo simbolo di v, il nastro contiene solo simboli blank.

Si dice che la configuarazione C_1 **produce** ($\vdash_M$) la configurazione C_2 se la macchina di Turing M può passare da C_1 a C_2 in un unico passo.

$$uaq_ibv \vdash_M uq_jacv \Leftrightarrow \delta(q_i, b) = (q_j, c, L),$$

Similmente:

$$uaq_ibv \vdash_M uacq_jv \Leftrightarrow \delta(q_i, b) = (q_j, c, R).$$

Abbiamo dei casi particolari nei casi in cui la testina si trova in una delle due estremità della configurazione:

- Per l'estremità di sinistra abbiamo:
  - $q_ibv \vdash_M q_jcv$ se la transizione è verso sinistra,
  - $q_ibv \vdash_M cq_jv$ se la transizione è verso destra.
- Per l'estremità di destra invece la configurazione è:

  $$uaq_i = uaq_i\sqcup,$$

  perché assumiamo che i simboli *blank* seguano la parte del nastro rappresentata dalla configurazione.

Avremo quindi come configurazione iniziale:

$$q_{start}w, \text{ dove } w \text{ è l'input,}$$

mentre come configurazione finale, usando lo stato $q \in \{q_{acc}, a_{rej}\}$:

$$uq_{acc}av, \qquad uq_{rej}av.$$

Condizione di accettazione

$$M \text{ accetta } w \in \Sigma^* \quad \Leftrightarrow \quad C_{start} = q_{start}w : C_{start} \vdash^*_M C_{acc},$$

dove C_{acc} è una qualsiasi configurazione di accettazione.

Definizione 2: Linguaggi Turing Riconoscibili

Un linguaggio è Turing riconoscibile se esiste una macchina di Turing M che lo riconosce, ovvero $\forall w \in L$ abbiamo che M accetta w.

La macchina di Turing quando riceve un input può produrre tre risultati:

- Accettato,
- Rifiutato,
- Loop, cioè che la macchina non arriverà mai ad uno stato di arresto.

Nota!

Se $w \notin L$, $M(w)$ può rifiutare o entrare in loop.

Poiché è difficile distinguere una macchina di Turing che è effettivamente in loop da una che sta semplicemente impiegando molto tempo, preferiamo considerare macchine di Turing che si fermano su ogni input. Una macchina del genere è chiamata **decisore** perché in ogni caso *decide* se accettare o rifiutare. Un decisore decide un linguaggio L se è in grado di distinguere correttamente tutte le stringhe che appartengono a L da quelle che non vi appartengono (quindi se lo riconosce).

Definizione 3: Linguaggio Turing Decidibile

Un linguaggio si dice Turing-decidibile (o decidibile) se esiste una macchina di Turing M che lo decide.

Nota!

Ogni descrizione formale di una particolare Turing Machine ha alcune scelte arbitrarie, scendendo in complicati livelli di dettaglio. Per questo motivo noi le eviteremo concentrandoci su precise e chiare descrizioni ad alto livello adatte ai nostri scopi.

1.2 Varianti di Turing Machines

Definizioni alternative di macchine di Turing prendono il nome di **varianti**. Queste varianti hanno lo stesso potere computazionale del modello originale (riconoscono lo stesso linguaggio).
Esempi di varianti:

- Nastro infinito in 2 direzioni,
- TM che si muove a R, L, S, ($S =$ STAY),
- TM con k-nastro,
- TM non deterministiche.

1.2.1 Macchina di Turing multinastro

Una macchina di Turing multinastro è come una normale macchina di Turing dotata però di vari nastri, ognuno con la propria testina per la lettura e la scrittura. La funzione di transizione viene modificata per permettere la lettura e la scrittura ma anche per spostare le testina contemporaneamente su alcuni o tutti i nastri:

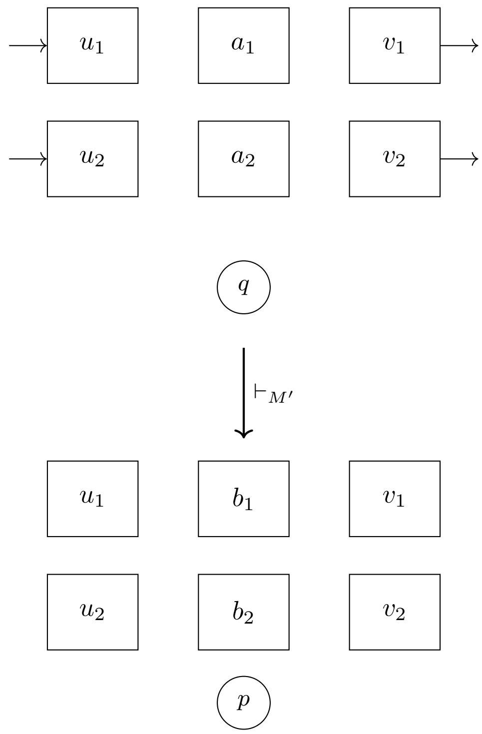
$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k,$$

Con k nastri. Semplicemente, diciamo che una TM standard può simulare una TM M' come sopra, muovendo prima a L e poi a R ogni volta che M' sta ferma. Più nel dettaglio:

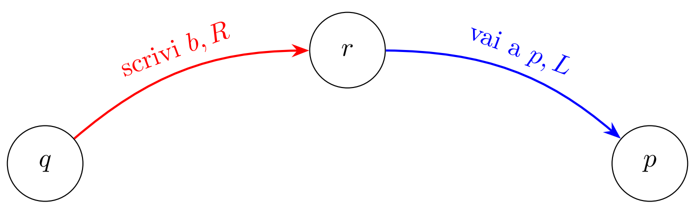
$$\delta'(q, a) = (p, b, S).$$

Introduciamo un nuovo stato r e aggiungo:

- $\delta(q, a) = (r, b, R)$
- $\delta(r, *) = (p, *, L)$



Macchina di Turing M'.



Macchina di Turing M che simula M' con stato intermedio r e due nastri.

Teorema 1.1. *Per ogni TM M' multinastro ne esiste una M a nastro singolo equivalente*

Idea: Occorre simulare sul nastro di M tutti i nastri di M'.

Dimostrazione. La TM M utilizzerà alcuni simboli speciali:

- # per separare i nastri di M'
- $\forall a \in \Gamma$, avremo $\hat{a}$ per tenere traccia delle posizioni delle k testine di M'

Vediamo ora il comportamento di M su un input w:

1. Per prima cosa M imposta il suo nastro nel formato aggiungendo i simboli appena introdotti:

$$\#\hat{w}_1 w_2 \cdots w_n \# \hat{\sqcup} \# \hat{\sqcup} \# \cdots \#.$$

2. Per simulare una mossa di M':

   - Scansiono il nastro per leggere i simboli in ognuno dei k nastri,
   - Torno indietro al primo #,
   - Scansiono nuovamente andando ad aggiornare i simboli mancanti in accordo a δ',
   - Se M' costringe M a muovere la testina emulata di uno dei nastri su #, M sposta tutto il nastro a R di una posizione e inserirà il simbolo $\hat{\sqcup}$.

□

Corollario 1.1. *Un linguaggio è Turing-decidibile se e solo se esiste una TM multinastro che lo decide.*