

# La Macchina di Turing

# Sommario

- Codifica dei dati
- Macchina Astratta
- Definizioni
- Esempi

# Codifica dei dati

- È possibile introdurre la teoria della computabilità facendo riferimento ad algoritmi che elaborano **numeri naturali**
- **Non è una limitazione**, nonostante l'informatica tratti algoritmi relativi a dati di varia natura (testi, immagini, suoni, video, ...)
- Non si ha perdita di generalità in quanto è possibile codificare algebricamente qualsiasi tipo di dato in un numero naturale

# Macchina Astratta (1)

- La Macchina di Turing (TM) è una **macchina astratta**: non corrisponde ad alcuna macchina reale
- Non si considerano quei vincoli che intervengono nella realizzazione di macchine da calcolo reali
  - Dimensione memoria
  - Tempo di calcolo
  - ...

## Macchina Astratta (2)

- È definita indipendentemente dalla sua realizzazione fisica
- Definisce esclusivamente relazioni funzionali tra le sue parti

## Macchina Astratta (3)

- Storicamente introdotta da Alan Turing per indagare il **problema della decisione** (**Entscheidungsproblem**) posto da David Hilbert:
  - Esiste una procedura **formale** per decidere se una data **affermazione matematica** è vera?
- La TM nasce quindi uno **strumento matematico concettuale**, per studiare problemi di natura matematica
  - e con elevato impatto sulla filosofia del XX e XXI secolo

## Verso la Definizione (1)

- Un calcolo svolto da un umano consiste nell'operare su un insieme di simboli scritti su un **opportuno supporto**
  - È **inessenziale** che il supporto sia bidimensionale, ma deve essere **sufficientemente grande**, e tale che in una posizione si possa scrivere un solo simbolo
- I simboli devono appartenere a un prefissato alfabeto

## Verso la Definizione (2)

- Un umano, per svolgere un calcolo, effettua una successione di **operazioni elementari**
  - Ogni operazione svolta per il calcolo dipende dalle **operazioni precedenti** e dai **simboli letti**
  - Ogni operazione svolta determina un **cambiamento dello stato mentale** di chi la svolge
- Un umano, mentre svolge un calcolo, ricorda solo un numero finito di simboli letti dal supporto
  - Quando necessario **cerca** le informazioni sul supporto, focalizzando l'attenzione sui simboli a **sinistra o a destra** della posizione corrente

## Verso la Definizione (3)

- Il calcolo ha fine quando giunti a un certo punto **non è possibile svolgere più alcuna operazione**
  - Il risultato potrebbe anche non essere ottenuto
- Un calcolo è svolto utilizzando un **insieme di dati di input**, che l'umano riceve dall'esterno
  - per semplicità supponiamo noto l'input **all'inizio del calcolo**

## Macchina di Turing: Struttura (1)

- È un apparato costituito da:
  - un **nastro** monodimensionale, di lunghezza infinita, suddiviso in celle, ognuna delle quali può essere vuota oppure contenere un solo simbolo
  - una **testina di lettura/scrittura** dei simboli dalle/sulle celle
    - La testina oltre a leggere/scrivere, si può **spostare** di una cella a sinistra, a destra, oppure può restare ferma

## Macchina di Turing: Struttura (2)

- Sul nastro sono scritti i **simboli manipolati** dalla TM
  - Appartengono ad un alfabeto  $\Sigma$

## Macchina di Turing: Funzionamento (1)

- In ogni fase del calcolo, la testina è posizionata su una cella del nastro, contenente un simbolo  $s_i \in \Sigma$
- La TM può svolgere una **operazione atomica**:
  - leggere il simbolo contenuto della cella
  - scrivere un simbolo (eventualmente vuoto) nella cella
  - spostare la testina di un passo (a sinistra o a destra)
  - lasciare la testina ferma

## Macchina di Turing: Funzionamento (2)

- La successione degli eventi precedenti determina in ogni istante **uno e un solo** (TM **deterministica**) **stato** della TM
  - siano  $q_1, q_2, \dots, q_n$  i possibili stati (**finiti**) di una TM
- La **configurazione** di una TM in un dato istante è la coppia ordinata definita dallo stato corrente  $q_i$  e dal simbolo  $s_j$  puntato dalla testina  $C = \langle q_i, s_j \rangle$

## Macchina di Turing: Funzionamento (3)

- Ogni TM è **programmata** per eseguire uno specifico calcolo
  - cioè dispone delle **istruzioni** per eseguire quell'unico compito
- Le istruzioni hanno la forma:  
 **$\langle \text{configurazione} \rangle \rightarrow \langle \text{operazione atomica} \rangle$**
- Ogni istruzione specifica l'istruzione atomica che deve essere svolta quando la TM si trova in una data configurazione

## Macchina di Turing: Funzionamento (4)

- Le istruzioni di una TM sono registrate in una **matrice funzionale** che associa a ogni configurazione **una e una sola** operazione atomica
- Affinché il calcolo **termini** è necessario che ad almeno una configurazione possibile non corrisponda alcuna istruzione
  - Queste configurazioni sono dette **finali**

## Macchina di Turing: I/O

- I dati su cui opera una TM sono forniti in input scrivendoli sul nastro dall'esterno prima dell'inizio del calcolo
- L'output è ciò che è scritto sul nastro alla fine del calcolo

## Definizione Formale

- Una TM è una 7-pla  $TM = \langle Q, \Sigma, \Delta, \delta, q_0, B, F \rangle$ 
  - $Q$  insieme finito e non vuoto di **stati**
  - $\Sigma$  **alfabeto della macchina**
  - $\Delta$  **alfabeto di input**
  - $\delta$  **funzione di transizione**

$$\delta : (Q \times \Sigma) \rightarrow (Q \times \Sigma \times \{L, R, S\})$$
 (con L spostamento a sinistra, R a destra, S stop)
  - $q_0 \in Q$  stato iniziale
  - $B$  spazio vuoto (blank) =  $\Sigma \setminus \Delta$
  - $F \subseteq Q$  insieme degli stati finali

## Esempio successore (1)

- Calcolo del successore di un numero
- Prima di definire la opportuna TM è necessario analizzare il problema, e “**idearne**” una soluzione
  - Esaminare l’ultima cifra della stringa di input
    - Se è < 9, allora sommare 1 e stop
    - Altrimenti sostituire l’ultima cifra con 0 ed esaminare la precedente
    - Ripetere il passo precedente

## Esempio successore (2)

- Sia  $\Delta = \{0, 1, 2, \dots, 9\}$ ;  $\Sigma = \Delta \cup B$
- Ci sono due stati:
  - $q_0$ : deve essere aggiunto 1 – stato iniziale
  - $q_1$ : è stato aggiunto 1 – stato finale
- La matrice funzionale è:

|       | 0         | 1         | 2         | 3         | 4         | 5         | 6         | 7         | 8         | 9         | B         |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| $q_0$ | 1 $q_1$ D | 2 $q_1$ D | 3 $q_1$ D | 4 $q_1$ D | 5 $q_1$ D | 6 $q_1$ D | 7 $q_1$ D | 8 $q_1$ D | 9 $q_1$ D | 0 $q_0$ S | 1 $q_1$ D |
| $q_1$ |           |           |           |           |           |           |           |           |           |           |           |

## Tesi di Church (versione preliminare)

- Ogni algoritmo può essere espresso da un’opportuna MdT
- Tutto ciò che è calcolabile, è calcolabile attraverso una MdT