



UNIVERSITÀ
degli STUDI
di CATANIA

Architettura di base dei calcolatori. Macchine teoriche

Corso zero F-N AA 2022/23

Corso di Laurea Triennale in Informatica

Fabrizio Messina

fabrizio.messina@unict.it

Dipartimento di Matematica e Informatica

1. Il calcolatore
2. Modelli teorici
3. Macchina di Turing
4. Architettura di Von Neumann

Il calcolatore

Un calcolatore elettronico è dotato di alcune componenti di base che permettono di :

- memorizzare dati
- elaborare dati
- trasferire dati dall'esterno / verso l'esterno

Un calcolatore elettronico:

- permette di eseguire del lavoro (la **computazione**) in modo automatico;
- il lavoro viene specificato in forma di **algoritmo**;
- **algoritmo** va specificato in un **linguaggio** “comprensibile” per il calcolatore;

Modelli teorici

La macchina di Turing rappresenta il primo modello di calcolo che sia stato definito e riveste ancora oggi enorme importanza.

La macchina di Turing rappresenta di fatto una formalizzazione del concetto di algoritmo.

È possibile definire un **algoritmo come una particolare macchina di Turing.**

La macchina di Von Neumann è importante perchè descrive il concetto di algoritmo in **modo pratico**.

L'architettura dei moderni calcolatori si basa sul modello di Von Neumann.

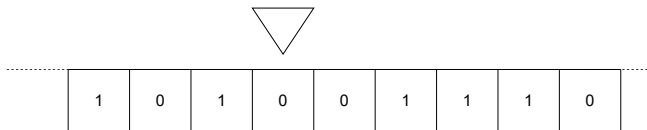
È possibile definire un algoritmo come un **programma della macchina di Von Neumann**.

Macchina di Turing

Alan Mathison Turing (1912–1954) sviluppò una macchina teorica (automa) per la quale dimostrò la possibilità di **eseguire qualsiasi algoritmo**.

La Macchina di Turing (MdT) permette di modellare il concetto di **Algoritmo**.

Macchina di Turing



Macchina di Turing (MdT) definita dai segg. elementi:

- un **nastro** di lunghezza infinita, suddiviso in celle
- il nastro contiene dei **simboli di un alfabeto finito**
- ogni cella del nastro contiene un simbolo oppure è vuota
- una **testina** si sposta lungo il nastro (dx o sx), la quale può leggere, scrivere o semplicemente cancellare il simbolo della singola cella

Comportamento della macchina di Turing.

Ad ogni singolo passo:

- la testina legge il simbolo sul nastro nella cella corrente (**input**)
- la macchina decide il suo prossimo **stato interno**
- la testina può modificare il contenuto della cella corrente sul nastro (**output**)
- la testina potrà muoversi a destra o sinistra sul nastro

In altre parole, il comportamento di una MdT può essere programmato come un insieme di regole o **quintuple** del tipo

$$(S, \alpha, S^*, \beta, \{<, >\})$$

S stato interno corrente, S^* Prossimo stato interno

α simbolo letto sulla cella del nastro, β simbolo da scrivere nella cella

'<' significa sposta la testina a sx, '>' significa sposta la testina a destra

Insieme degli stati interni deve includere *stato iniziale* e *stato finale*.

Il risultato della computazione sarà rappresentato *dal contenuto del nastro quando la MdT avrà raggiunto lo stato finale*.

Turing focalizzò la sua attenzione sul **processo di calcolo**, indipendentemente da come esso avviene fisicamente.

In particolare, il modello si basa sul processo di calcolo eseguibile da un *essere umano idealizzato* (U).

Una M.d.T è infatti un **dispositivo ideale**, ovvero indipendente da ogni sua possibile realizzazione fisica.

Condizioni di determinatezza

Le azioni della M.d.T (U), ad ogni istante, dipendono solo

- i) dai simboli contenuti nella casella osservata in quell'istante
- ii) dallo *stato mentale* corrente (cioè da quello che U ricorda in quel momento).

Tale procedimento è così simulabile da una macchina di Turing.

Condizioni di finitezza

Il numero di simboli è fissato e finito. Altrimenti, se il numero di simboli fosse infinito, per i limiti della nostra capacità di percezione, avremmo simboli così simili fra loro che non riusciremmo a distinguerli.

Il numero di caselle del nastro osservabili in una volta è finito (questa richiesta è implicata dai limiti delle nostre capacità di percezione).

È possibile ricordare distintamente solo un numero finito di stadi precedenti del processo di calcolo, altrimenti ci sarebbero stadi così simili uno all'altro che non sapremmo distinguerli.

Turing afferma che la memoria è uno “stato mentale”, ma intende con questo **l'influenza degli stadi precedenti del calcolo** su quello attuale.

Condizioni di finitezza (continua..)

Le operazioni che si possono compiere sono:

- **Cambiare** il contenuto di **alcune caselle** osservate.
- **Cambiare le caselle osservate** (spostare l'attenzione su altre casella).
- **Cambiare** il proprio **stato mentale** (cioè quello che ricorda del calcolo).
- **Osservare nuove caselle** che si trovano ad una **distanza massima** prefissata L da una qualsiasi delle caselle osservate.

Definizione di funzione (parziale) Turing-computabile:

una funzione parziale $f_i(a)$ si dice Turing-computabile se esiste una M.d.T., diciamo T_i , che è in grado di computare, con un numero finito di passi il suo valore (se esiste)

Ci sono però funzioni che una M.d.T. non può calcolare. (*)

Se l'analisi di Turing è corretta, *ogni funzione parziale a valori interi che può essere computata da un essere umano e che soddisfa le condizioni di finitezza e determinatezza date sopra è "Turing-computabile"*, cioè può essere computata da una macchina di Turing.

(*) richiede ulteriori approfondimenti teorici

Tesi di Turing.

Ogni funzione parziale calcolabile con un algoritmo è una funzione parziale calcolabile da una macchina di Turing.

Tesi di Church. *Ogni funzione intuitivamente computabile è computabile con la macchina di Turing* (intuitivamente significa che molte persone riconoscono l'algoritmo come computabile).

La tesi di Turing è equivalente a quella di Church, per questo si parla spesso di “tesi di Church-Turing”.

Osservazioni:

Definire un insieme di regole o quintuple per definire una specifica MdT è equivalente a definire un **algoritmo per la risoluzione di un problema** (ad esempio sommare due numeri).

Macchina di turing universale: una macchina di Turing in grado di calcolare tutte le funzioni calcolabili da ogni singola macchina di Turing.

In un articolo del 1950, Alan Turing immaginò un criterio per *determinare se una macchina sia in grado di “pensare”, in altre parole di mostrare un comportamento “intelligente”*.

Turing prende spunto dal “gioco dell’imitazione”, a tre partecipanti: un uomo (A), una donna (B), e una terza persona C.

- Quest’ultima è tenuta separata dagli altri due e tramite una serie di domande deve stabilire qual è l’uomo e quale la donna.
- A cerca di ingannare C e portarlo a fare un’identificazione errata.
- B cerca di aiutare C.
- Conversazioni dattiloscritte o trasmesse in qualche modo per non aiutare C (es: timbro della voce).

Turing immagina di **sostituire A con una macchina**, che quindi cerca di ingannare l'interlocutore C facendogli credere di essere un umano.

Se si conta quante volte C indovina chi sia l'uomo e chi la donna nei due scenari differenti, e se la percentuale di successi è la stessa, allora la macchina dovrebbe essere considerata "intelligente" perchè *indistinguibile da un essere umano*.

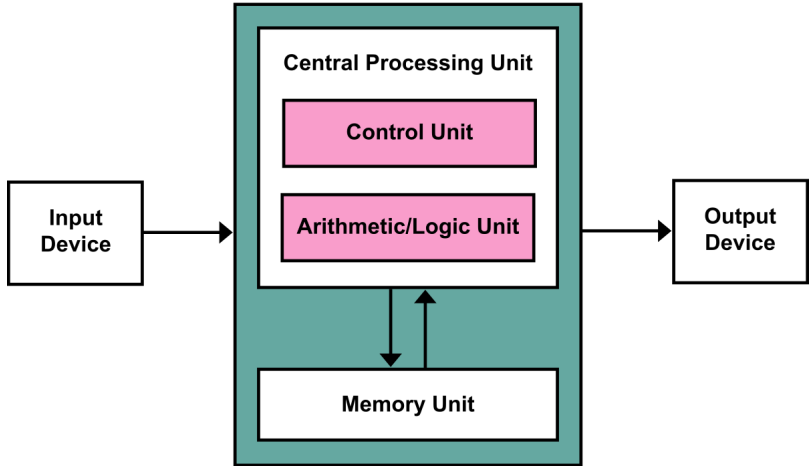
La disciplina **dell'intelligenza artificiale** si occupa di studiare e sviluppare modelli di calcolo che permettano di realizzare sistemi informatici "intelligenti" ovvero di simulare un comportamento del pensiero umano.

Architettura di Von Neumann

Von Neumann (1903 – 1957) contribuì, partendo dal progetto [ENIAC](#), alla realizzazione di un calcolatore con un'architettura adottata ancora oggi nei moderni calcolatori:

- Memoria centrale detta anche RAM (Random Access Memory)
- Processore

Architettura di Von Neumann



Una macchina di Von Neumann è definita formalmente come una terna $\{N, IS, P\}$:

- N è l'insieme dei numeri naturali (rappresenta l'alfabeto della macchina)
- IS è l'insieme delle istruzioni (Instruction Set) generiche della macchina
- P è una sequenza finita di istruzioni che operano su dati specifici (il programma).

Le istruzioni del programma P:

- vengono registrate in memoria centrale (RAM)
- eseguite secondo l'ordine specificato nel programma
- la loro esecuzione si arresta a seguito di una specifica istruzione di arresto.

Ad ogni cella in memoria è associato un indice detto anche indirizzo della locazione della memoria.

Il programma è registrato nella prima parte della memoria centrale, mentre i dati sono registrati nelle locazioni successive.

Processore:

- Program counter: una cella in memoria che contiene l'indirizzo della prossima istruzione da eseguire;
- registro delle istruzioni: locazione di memoria contenente l'istruzione da eseguire;
- unità logico-aritmetica;
- unità di controllo: sistema che si occupa dell'esecuzione dell'istruzione, controllando le altre componenti del processore;

Architettura di Von Neumann

La computazione si basa sull'esecuzione delle istruzioni del programma nel seg modo:

- legge il contenuto del contatore (indirizzo dell'istruzione da eseguire)
- copia nel registro istruzioni (fetch) l'istruzione da eseguire;
- decodifica l'istruzione
- invia segnali all'unità logico-aritmetica per l'esecuzione dell'istruzione;
- acquisisce dalla memoria eventuali operandi dell'istruzione (es: operazione somma)
- una volta che l'unità logico-aritmetica ha calcolato il risultato, lo memorizza in una opportuna locazione di memoria;

Questi argomenti saranno trattati in modo piú approfondito e formale durante il corso di Fondamenti di Informatica (Proff. F. Barbanera e M. Madonia)

[Dispense Pdf](#)