Modello di calcolo di Von Neumann, macchine di Turing

Architettura degli elaboratori AA 2023/24

Corso di Laurea Triennale in Informatica

Massimo Orazio Spata

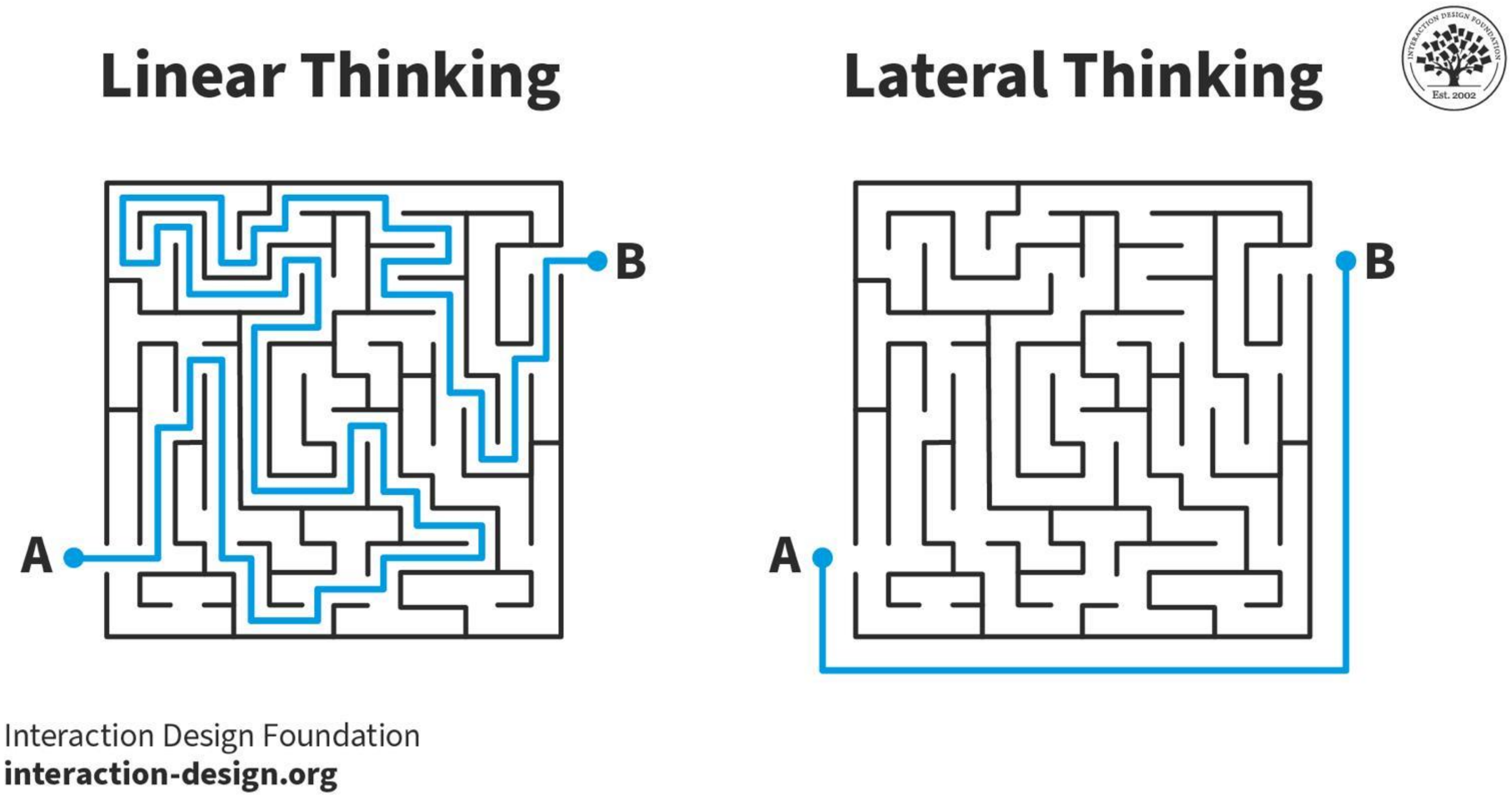
massimo.spata@unict.it

Dipartimento di Matematica e Informatica

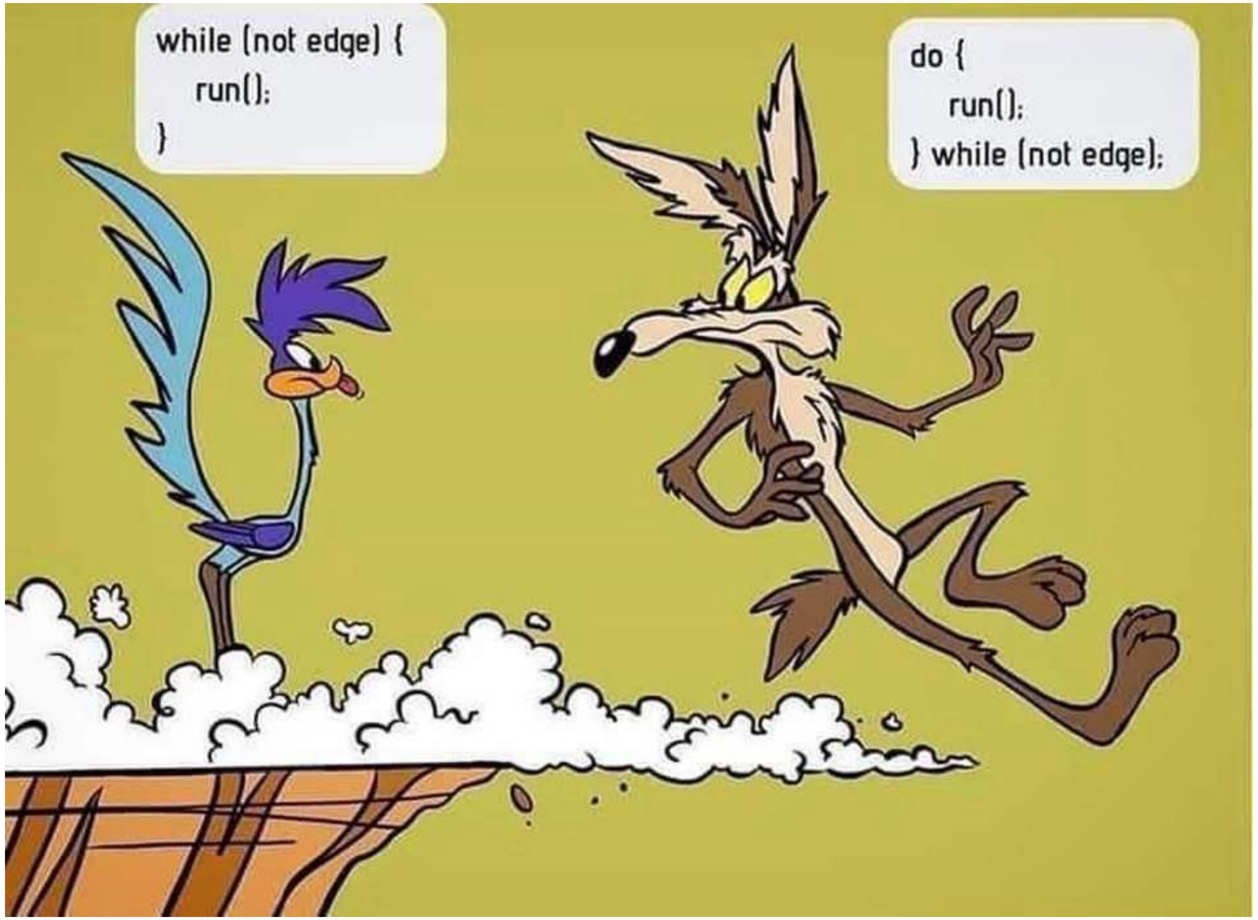
Algoritmo

- Possiamo definire un algoritmo come un insieme di istruzioni, scritte, date, ..., codificate in un certo linguaggio, che stabilisce come si deve effettuare l’esecuzione di un certo lavoro.

Lateral thinking



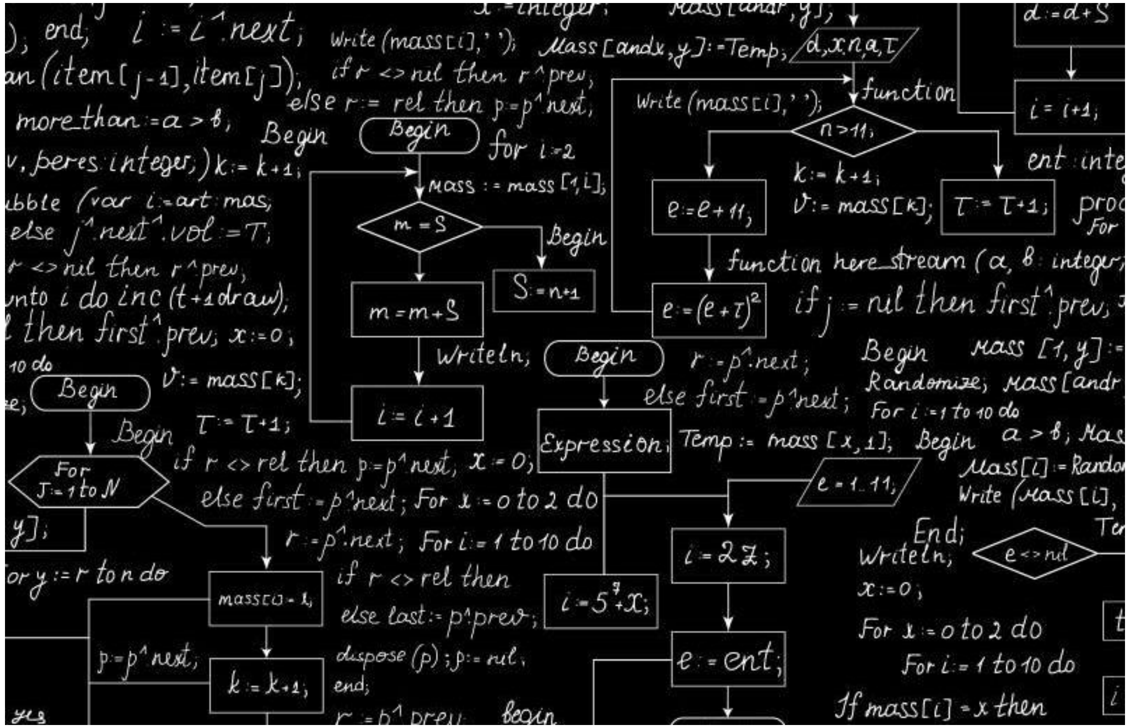
Ma con attenzione



Esempio di algoritmo

Supponiamo di voler cucinare una pietanza. Descriviamo i passi necessari per compiere questo “lavoro”:

- Prendere una pentola
- Mettere mezzo bicchiere di olio di oliva nella pentola
- Mettere due agli spezzettati nella pentola
- Mettere la pentola sul fuoco fino a che l’aglio è dorato
- Aggiungere 500 gr di pomodori nella pentola
- Cuocere per 15 minuti
- Aggiungere 8 foglie di basilico nella pentola
- Cuocere per 15 minuti
- Prendere una pentola più grande
- Riempirla d’acqua
- Metterci un pugno di sale
- Mettere la pentola sul fuoco fino a che l’acqua non bolle
- Buttare 500 gr di spaghetti nella pentola
- Cuocere per 9 minuti
- Scolare gli spaghetti
- Mescolare gli spaghetti al sugo della prima pentola

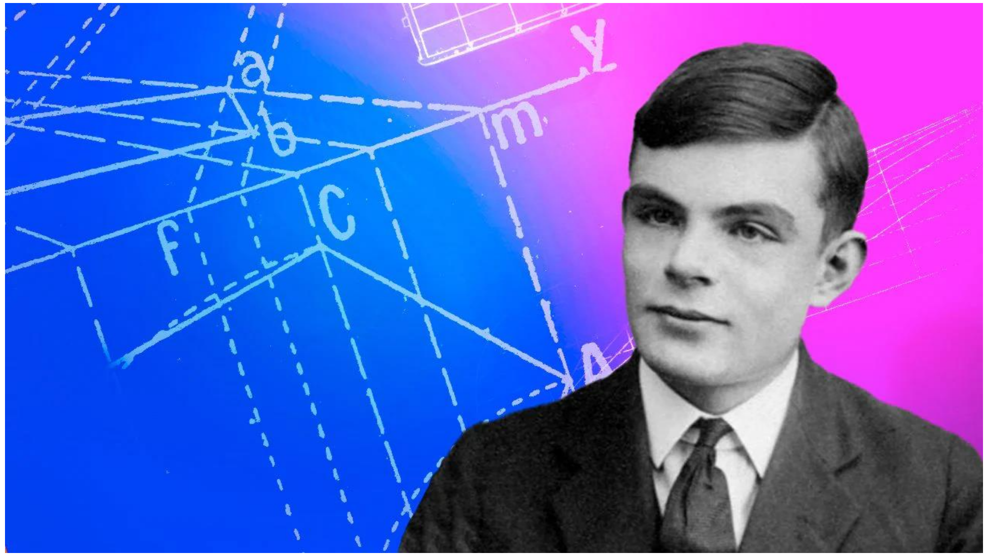


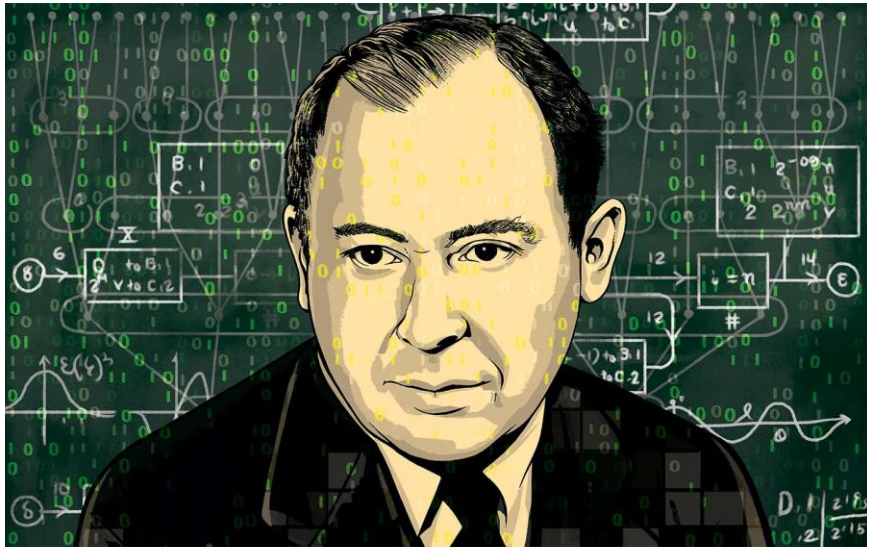
Un altro esempio

- Un altro esempio potrebbe essere quello di effettuare un calcolo matematico a partire da un dato noto, ad esempio quanto vale una certa funzione in corrispondenza di un dato valore.
- Ma possiamo descrivere anche un’attività artistica, ad esempio dipingere quadri, con una serie di istruzioni fissate? Su questo i dubbi sono molteplici; la componente creativa è riassumibile in una sequenza logica di operazioni...?
- Va allora capito con maggior chiarezza cosa si intende per algoritmo.

Definizione di Algoritmo

- Un Algoritmo è una particolare macchina di Turing (A.M. Turing 1912- 1954, matematico britannico) oppure un programma della macchina di Von Neumann (J. von Neumann 1903-1957, matematico statunitense di origine ungherese).





Macchine Teoriche e Architettura di base

- Per architettura di un calcolatore elettronico si intende l'insieme delle principali unità funzionali di un calcolatore ed il modo in cui queste interagiscono.
- Oggi le funzioni di base di un calcolatore si potrebbero così riassumere:
  - memorizzazione dei dati
  - elaborazione dei dati
  - trasferimento dei dati
  - controllo.

Il personal computer

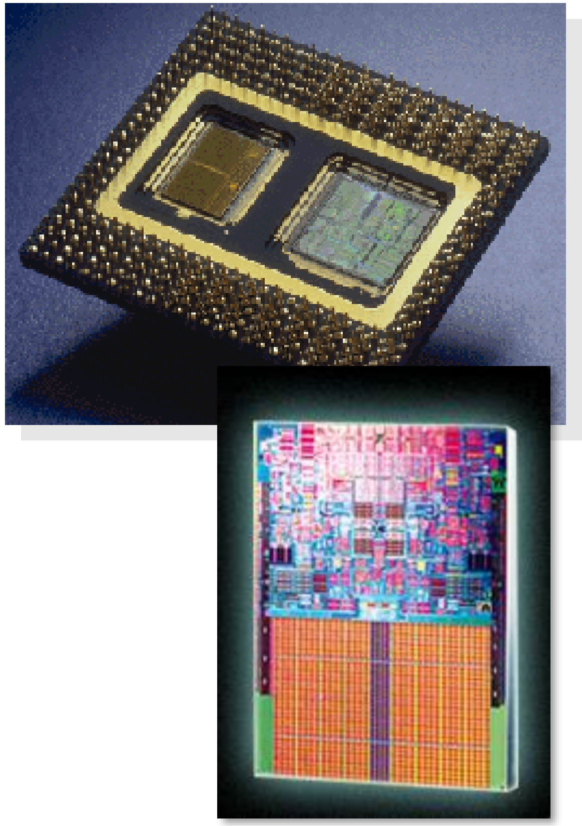
Tipi di computer

- Mainframe
- Server
- Desktop
- Laptop (portatile) o notebook
- Tablet PC
- Palmari (PDA)
- Telefoni cellulari / smartphone
- Lettori multimediali

Cos’è un computer?

- Ma che cos’è un computer?
- Ad una prima analisi possiamo dire che è una macchina che computa, ovvero che esegue un tipo di lavoro in maniera automatica, ovvero che esegue un algoritmo specificato in un linguaggio che può essere capito dalla macchina.

Il processore

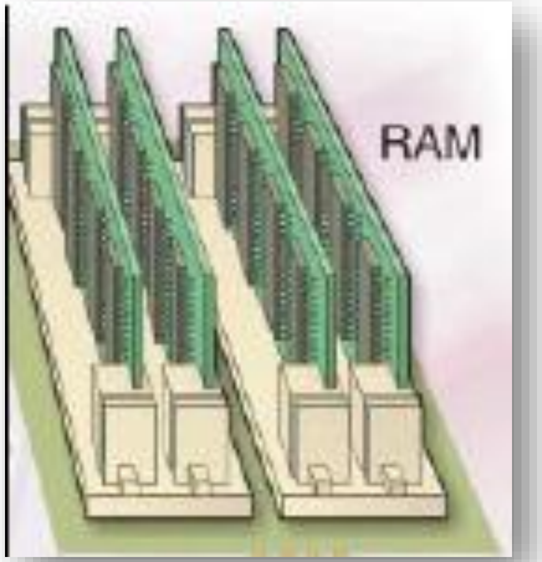


- microprocessore:

Central Processing Unit (**CPU**)

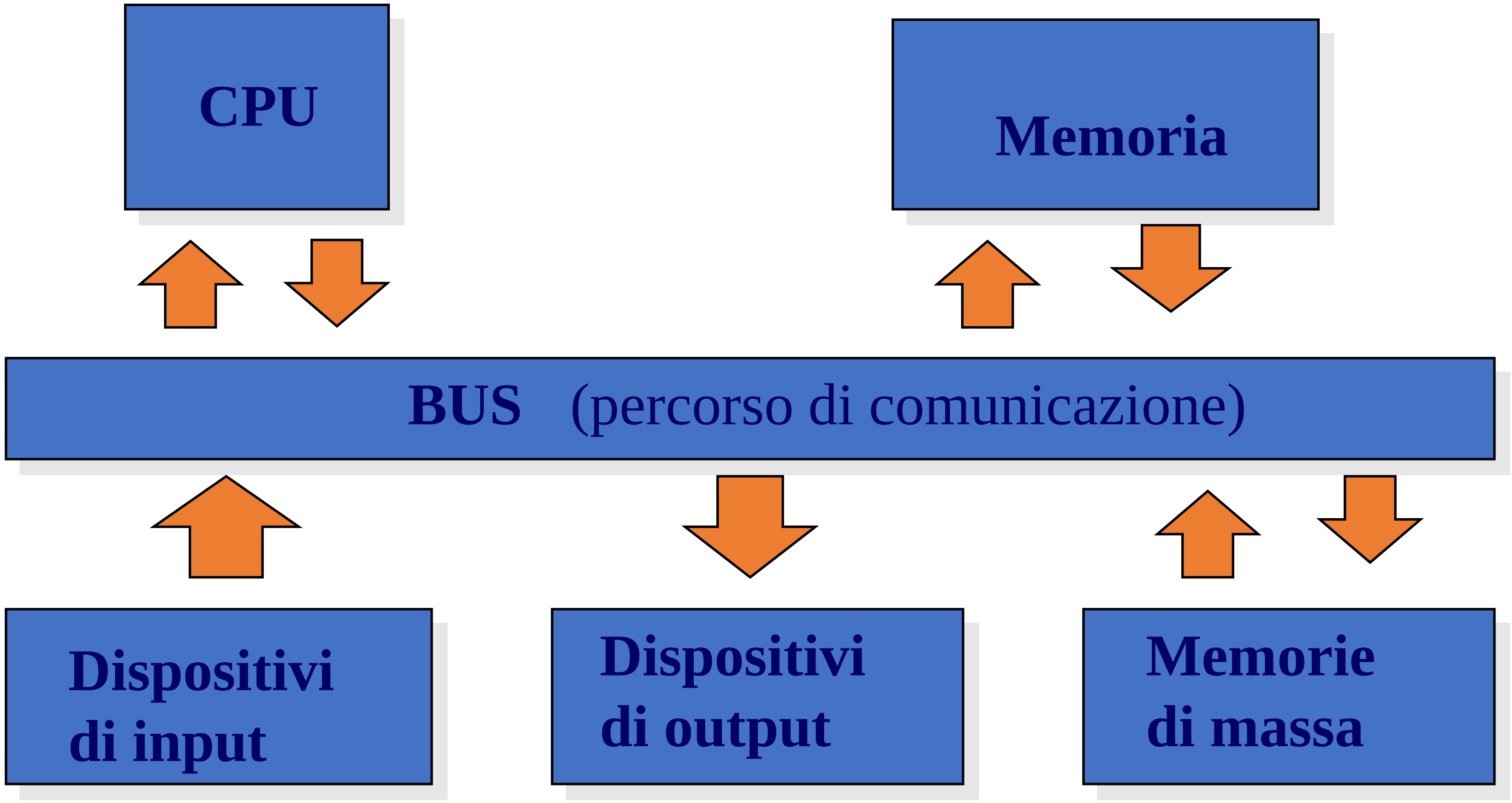
- memorie di lavoro proprie = **registri**
- velocità del processore = numero di cicli al secondo (**Mhz** o **Ghz**)

La memoria centrale



- deposito di dati e di istruzioni da eseguire
- **ROM** (*Read Only Memory*)
- **RAM** (*Random Access Memory*)
- Memorie **cache**

Macchina di Von Neumann



Macchina di Von Neumann

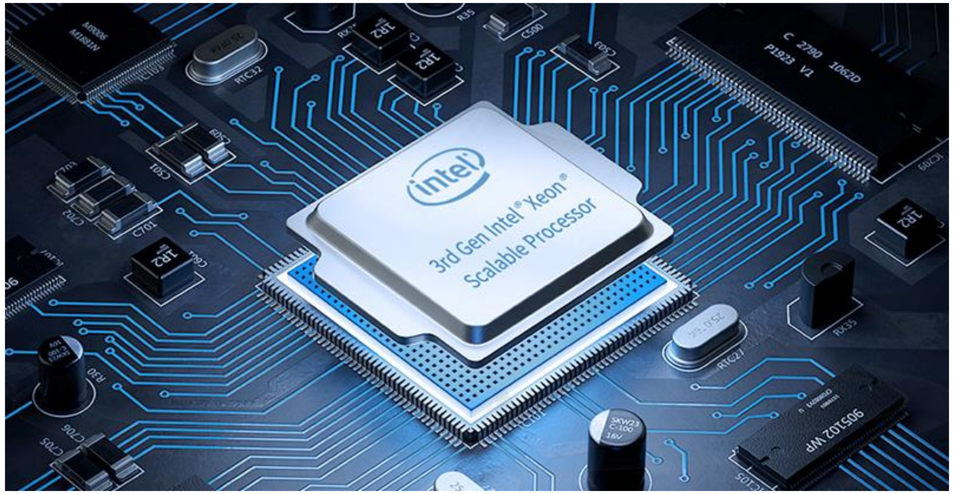
- Una macchina di Von Neumann è una terna (N, IS, P) dove:
- N = {0, 1, 2, 3, ...} è l’insieme dei numeri naturali (rappresenta l’alfabeto della macchina);
- IS = {ZERO, INC, SOM, SOT, MOL, DIV, UGUALE, MINORE, SALCOND, ALT} è l’Instruction Set ovvero l’insieme delle istruzioni generiche della macchina
- P = {I_0, I_1, I_2, I_3, ..., $I_{|P|}$ – 1} è una sequenza finita e non vuota di istruzioni specifiche prese dall’insieme IS in cui siano specificati particolari valori delle variabili. Questa sequenza è detta il programma della macchina.

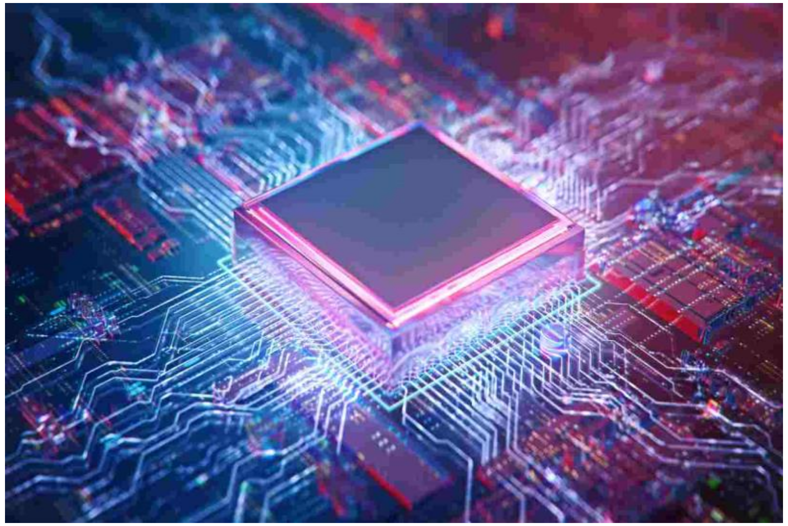
Funzionamento della macchina di Von Neumann

- Un programma eseguibile dalla macchina di Von Neumann consiste in una lista di istruzioni registrate in memoria centrale, che devono essere eseguite una alla volta secondo l'ordine specificato nel programma fino a quando non si incontra un’istruzione di arresto.
- La geniale soluzione che venne trovata da Von Neumann è quella di usare la memoria per conservare sia le istruzioni che i dati dei calcoli.
- L’Hardware della macchina di Von Neumann è costituito dalla memoria, ovvero dall’insieme delle locazioni di memoria, e dal processore.
- Inoltre:
  - ad ogni locazione è associato l’indice della locazione nella sequenza detto indirizzo della locazione di memoria
  - nella memoria sono registrati il programma ed i dati del programma
  - il programma è registrato nelle locazioni di memoria i cui indirizzi vanno da 0 a $|P| - 1$ (i-esima istruzione del programma registrata nella locazione i)

Processore

- Il processore è costituito da 4 parti fondamentali:
    1. il contatore di programma, una locazione di memoria contenente l’indirizzo dell’istruzione da eseguire
    2. il registro delle istruzioni, una locazione di memoria contenente l’istruzione da eseguire
    3. l’unità aritmetica e logica, un sistema che esegue l’istruzione
    4. il controllo, un sistema che, attraverso una sequenza di cambiamenti di stato, fa avvenire l’esecuzione dell’istruzione, dunque ha il comando del processo.





Computazione

- La computazione della macchina avviene eseguendo le istruzioni del programma nell’ordine definito dal programma, a meno di salti condizionati, nel seguente modo:

1. Legge il contenuto del contatore, ovvero l’indirizzo dell’istruzione da eseguire;
2. fa pervenire nel registro istruzioni (fetch) l’istruzione da eseguire;
3. decodifica l’istruzione, ovvero “capisce” di quale istruzione si tratta (fra quelle possibili in IS);
4. invia segnali all’unità logico-aritmetica per far eseguire l’istruzione;
5. acquisisce dalla memoria i dati necessari (attraverso gli indirizzi delle locazioni di memoria). Se ad esempio l’istruzione è SOM(M1, M2) M1 e M2 sono indirizzi di locazioni di memoria;
6. aspetta che l’unità logico-aritmetica calcoli il risultato;
7. registra il risultato nella locazione di memoria specificata dall’operando più a sinistra dell’istruzione. Se l’istruzione è SOM(M1, M2) il risultato viene registrato nella locazione di memoria il cui indirizzo è M1. Se l’istruzione è un salto condizionato, il controllo registra nel contatore l’indirizzo della prossima istruzione;
8. Incrementa il contatore a meno che l’istruzione non sia un salto condizionato o un ALT;
9. Il ciclo si ripete fino a che non si incontra l’istruzione ALT.

Esempio

- Realizzare il programma per la macchina di Von Neumann che calcola se un numero è divisibile per 3 (il valore è registrato in M0; se è divisibile per tre scrivere in M1 il valore 1 altrimenti scrivere 0)

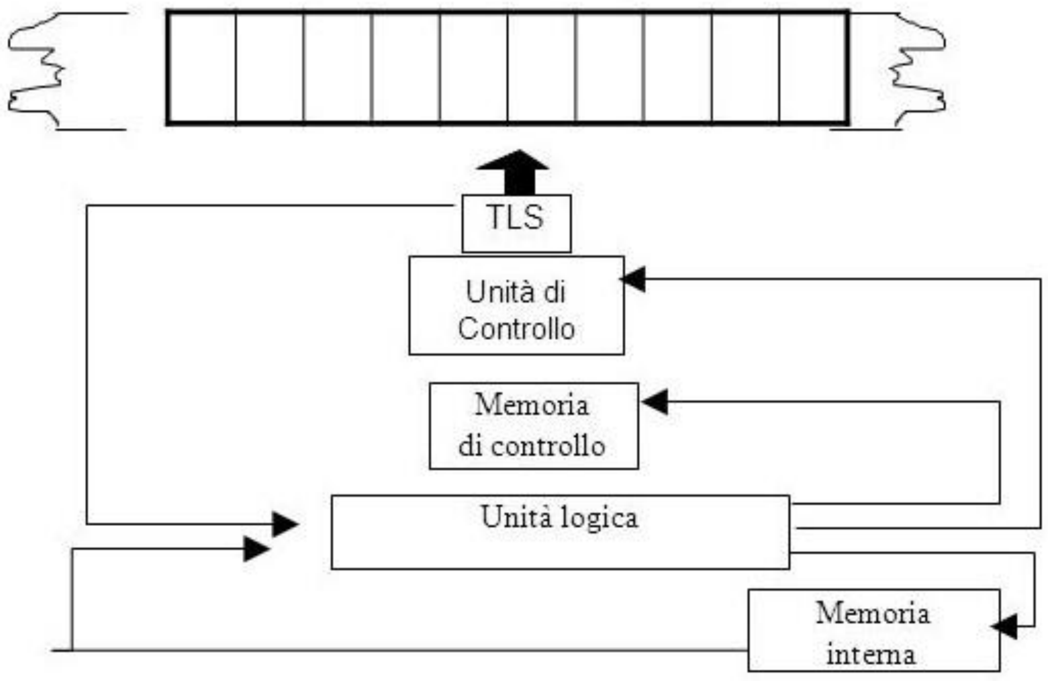
0	ZERO(M1)	Scrive 0 in M1
1	INC(M1)	Scrive 1 in M1
2	INC(M1)	Scrive 2 in M1
3	INC(M1)	Scrive 3 in M1
4	ZERO(M2)	Assegna a 0 in M2
5	SOM(M0,M1)	Copia il contenuto di M1 in M0
6	DIV(M2,M1)	Calcola r = n–3(n/3) e mette il risultato in M0
7	MOL(M2,M1)	Calcola r = n–3(n/3) e mette il risultato in M0
8	SOT(M0,M2)	
9	SALCOND(M0, 13)	Se r≠0 salta a 13
10	ZERO(M1)	Assegna 0 in M1
11	INC(M1)	Incremento 0 a 1 in M1
12	ALT	Scrive 1 in M1 e si ferma
13	ZERO(M1)	Assegna 0 in M1
14	ALT	Scrive 0 in M1 e si ferma

Macchina di Turing e di Von Neumann

- La macchina di Turing e la macchina di Von Neumann sono due modelli di calcolo (ovvero; modi di definire e/o eseguire algoritmi) specifici. In breve, stiamo definendo il concetto di algoritmo come tutto ciò che può essere eseguito dai due tipi di macchine su menzionate.
- Tra tutti i modelli di calcolo esistenti, questi due giocano un ruolo importante. Il modello di calcolo descritto dalla MdT (introdotto nel 1936) è importante perché è il primo modello di calcolo che sia stato definito ed è tuttora usato come definizione teorica di algoritmo.
- Il modello di calcolo descritto dalla macchina di Von Neumann (introdotto nel 1946) è importante perché traduce in pratica il concetto di algoritmo.
- In altre parole, tutti i più moderni computer sono strutturati (ancora) come una macchina di Von Neumann. Definiremo la macchina di Von Neumann più avanti.

I componenti di una macchina di Turing

- Da un punto di vista informale una MdT è costituita da:
  - Un nastro
  - Una testina di lettura/scrittura
  - Una unità di memoria interna
  - Una unità di calcolo
  - Una unità di controllo
  - Una unità di logica



Macchina di Turing

- Nel 1854, il matematico britannico George Boole (1815 - 1864), elaborò una matematica algebrica - di fondamentale importanza nella progettazione degli odierni computer - che da lui prese il nome. Nell'algebra booleana le procedure di calcolo si possono effettuare grazie a operatori matematici (AND, OR, NOT, ecc.) corrispondenti alle leggi della logica.
- L'algebra di Boole entrò prepotentemente alla ribalta nel 1936, quando il matematico britannico Alan Mathison Turing (1912-1954), immaginò una "macchina" o "automa" (che oggi sembra banale), esistente unicamente a livello teorico, con la quale dimostrò formalmente la possibilità di realizzare una macchina in grado di eseguire qualsiasi algoritmo: una procedura di calcolo o, più in generale, l'elenco delle operazioni necessarie per risolvere un problema in un numero finito di operazioni.

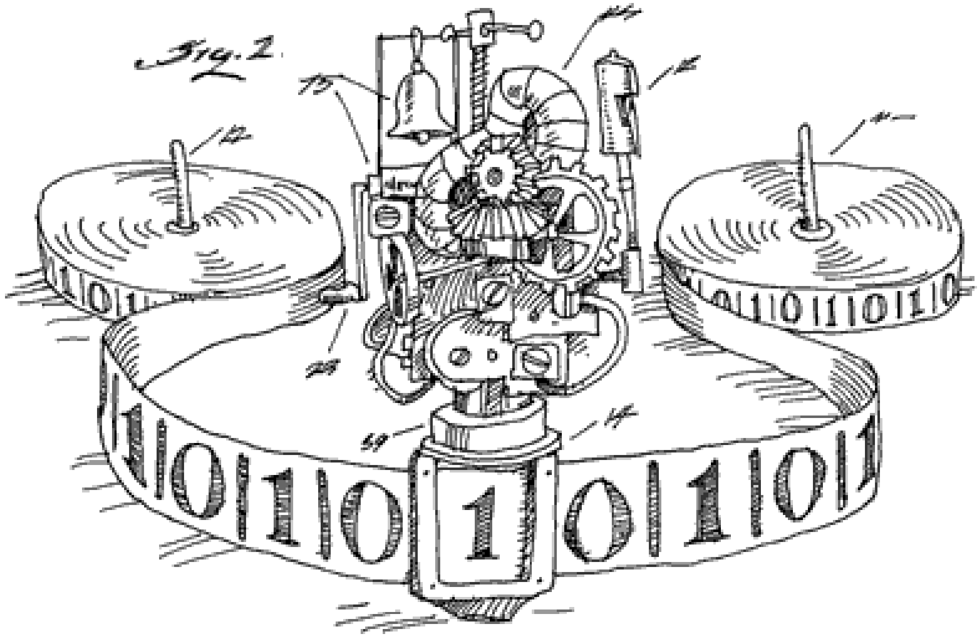
Macchina di Turing

- Una macchina di Turing (MdT) è definita da un insieme di regole che definiscono il comportamento della macchina su un nastro di input-output (lettura e scrittura).
- Il nastro può essere immaginato come un nastro di carta di lunghezza infinita, diviso in quadratini dette celle che dunque formano una sequenza lineare di celle.
- Ogni cella contiene un simbolo oppure è vuota. Una MdT ha una testina che si sposta lungo il nastro leggendo, scrivendo oppure cancellando simboli nelle celle del nastro.
- La macchina analizza il nastro, una cella alla volta, iniziando dalla cella che contiene il simbolo più a sinistra nel nastro. La macchina di Turing contiene un insieme finito di stati, un alfabeto finito (comprendente un simbolo nullo) e un insieme finito di istruzioni.

Macchina di Turing

Ad ogni passo, la macchina in accordo al suo stato interno corrente:

1. legge un simbolo sul nastro
2. decide il suo prossimo stato interno
3. scrive un simbolo sul nastro
4. decide se spostare la testina (di una posizione)



Macchina di Turing

- Come per uno stato della mente di un essere umano, lo stato interno di una MdT definisce l'ambiente in cui una decisione viene presa. Una MdT può avere solo un numero finito di stati.
- Il comportamento di una MdT può essere programmato definendo un insieme di regole, o quintuple, del tipo:
- (stato-interno-corrente, simbolo-letto, prossimo-stato-interno, simbolo-scritto, direzione).
- Per esempio la quintupla (0, A, 1, B, -) indica che se la macchina si trova nello stato interno 0 e legge sul nastro il simbolo A, allora passa nello stato interno 1, scrive B sul nastro e non sposta la testina di lettura.
- La quintupla (1, B, 0, A, >) indica invece che se la macchina si trova nello stato interno 1 e legge sul nastro il simbolo B, allora passa nello stato interno 0, scrive A sul nastro e si sposta di una posizione a destra.

Macchina di Turing (M.d.T.)

- un dispositivo di calcolo in grado di operare, mediante una successione (finita) di passi, secondo determinate regole (programma), su di un numero finito di simboli, facendo astrazione da limiti di spazio (memoria), di tempo (lunghezza della computazione) e da possibili errori di calcolo.

M.d.T.

- È importante sottolineare come l'attenzione di Turing sia rivolta al processo di calcolo, indipendentemente da come esso avviene fisicamente. In modo rigoroso, infatti, una M.d.T è un dispositivo ideale, cioè indipendente da ogni sua possibile realizzazione fisica.
- Sulla base della nozione di M.d.T. possiamo definire il concetto 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.

Funzione parziale

- Nella definizione di una macchina di Turing universale, si usano le funzioni parziali per una ragione fondamentale legata alla natura della computabilità e alla capacità delle macchine di Turing di eseguire calcoli.
- Cos'è una funzione parziale?Una funzione parziale è una funzione che non è necessariamente definita per ogni possibile input. Questo significa che per alcuni input, la funzione potrebbe non restituire alcun output (ovvero, potrebbe "non terminare" o "non essere definita").

Perché le funzioni parziali sono importanti nelle macchine di Turing universali?

- Rappresentazione della computabilità reale: Non tutte le funzioni che possiamo descrivere sono calcolabili in un tempo finito.
- Alcune computazioni potrebbero andare avanti indefinitamente (per esempio, se cercassimo di risolvere il problema della fermata di un'altra macchina di Turing).
- Le funzioni parziali catturano questa idea, rappresentando non solo le computazioni che terminano con successo (come le funzioni totali), ma anche quelle che potrebbero non terminare mai.

Generalità della computazione

- Una macchina di Turing universale deve essere in grado di simulare qualsiasi altra macchina di Turing.
- Poiché alcune macchine di Turing potrebbero non terminare per certi input, la funzione che descrive il comportamento di queste macchine è parziale. Usare funzioni parziali permette alla macchina universale di essere abbastanza potente da simulare qualsiasi macchina di Turing, incluse quelle che non terminano.

Accuratezza nella modellazione dei problemi di decisione

- Molti problemi di decisione (come il problema della fermata) sono tali che non esiste un algoritmo che fornisca una risposta in tutti i casi. Le funzioni parziali modellano accuratamente questi problemi, riconoscendo che per alcuni input, non ci sarà mai una risposta.

Funzioni parziali e MdT

- Le funzioni parziali vengono utilizzate nella definizione di una macchina di Turing universale per riflettere la realtà della computabilità, dove non tutte le computazioni terminano con successo.
- Questo rende il modello più realistico e potente, in grado di rappresentare tutte le possibili computazioni, inclusi quei processi che potrebbero continuare indefinitamente senza produrre un risultato.

Condizioni di finitezza

1. 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.
2. Il numero di caselle del nastro osservabili in una volta è finito (questa richiesta è implicata dai limiti delle nostre capacità di percezione).
3. 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.
4. Le operazioni che si possono compiere sono:
    1. Cambiare il contenuto di alcune caselle osservate
    2. Cambiare le caselle osservate (cioè spostare l'attenzione da una all'altra)
    3. Cambiare il proprio stato mentale (cioè quello che ricorda del calcolo)
    4. Osservare nuove caselle che si trovano al massimo ad una distanza prefissata L da una qualsiasi delle caselle osservate.

Macchina di Turing Universale

- Ad una macchina di Turing associamo un procedimento di calcolo idealizzato, cioè supponiamo di non avere limiti di spazio (il nastro su cui il calcolo avviene è potenzialmente infinito) e di tempo, e che la macchina non commetta errori. Chiamiamo infine ***UNIVERSALE*** una macchina di Turing in grado di calcolare tutte le funzioni calcolabili da ogni singola macchina di Turing.
- Allora possiamo dire, se l'analisi di Turing è corretta, che ogni funzione parziale a valori interi che può essere computata da un essere umano che soddisfa le condizioni di finitezza e determinatezza date sopra è Turing-computabile, cioè può essere computata da una macchina di Turing.

Condizione di determinatezza

- Le azioni di U, ad ogni istante, dipendono solo dai simboli contenuti nella casella osservata in quell'istante e dallo "stato mentale" corrente (cioè da quello che U ricorda in quel momento). Un procedimento così caratterizzato è simulabile da una macchina di Turing.

Tesi di Turing

- *"ogni funzione parziale calcolabile con un algoritmo è una funzione parziale calcolabile da una macchina di Turing."*

Il Test di Turing

- **Can machines think?**
- Turing riformulò questa domanda, per certi versi classica, nei termini di un gioco, che chiamò gioco dell'imitazione.
- Questo viene giocato da tre persone, un uomo (A), una donna (B) e un interrogante.
- L'interrogante viene chiuso in una stanza, separato dagli altri due, i quali sono a lui noti con le etichette X e Y.
- Scopo del gioco per l'interrogante è determinare quale sia l'uomo e quale la donna, facendo delle domande del tipo "Vuol dirmi X la sua altezza?"

Test di Turing

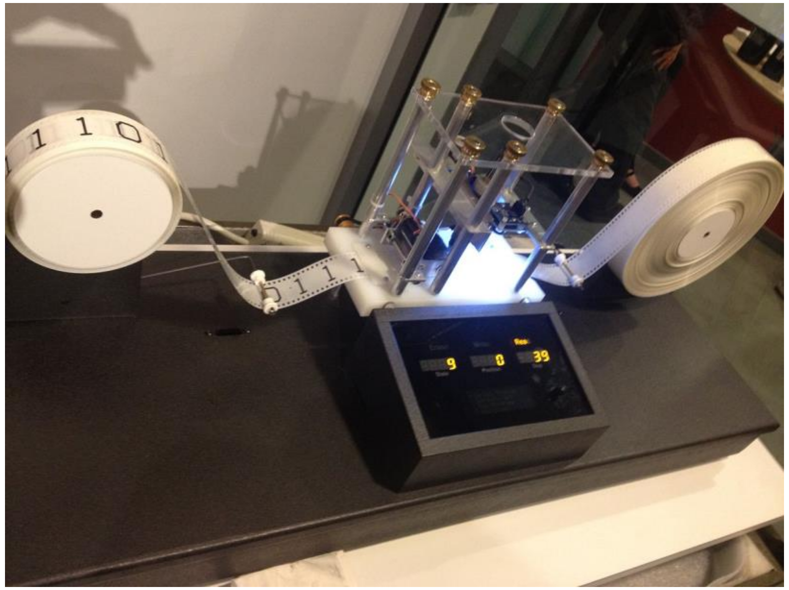
- Affinché né il tono della voce né la calligrafia possano aiutare l'interrogante, le risposte sono dattiloscritte.
- Lo scopo di A nel gioco è quello di ingannare l'interrogante e far si che fornisca una identificazione errata.
- Lo scopo di B è invece quello di aiutarlo.
- Turing si chiese che cosa sarebbe accaduto se una macchina avesse preso il posto dell'uomo nel gioco.

Test di Turing

- Più precisamente è vero che, modificando il calcolatore in modo da avere a disposizione una memoria adeguata, incrementando adeguatamente la sua velocità di azione e fornendogli una programmazione adeguata, C può prendere soddisfacentemente la parte di A nel gioco dell'imitazione, se la parte di B viene assunta da un essere umano?
- Per una macchina, il test di Turing consiste quindi nell'ingannare un essere umano giocando al gioco dell'imitazione, inducendolo a credere di conversare con un altro essere umano e non, appunto, con una macchina.

Criterio di similitudine

- Il criterio di similitudine incorporato nel test, per valutare la somiglianza tra la macchina e l'essere umano, è la capacità di interagire linguisticamente, ovvero la capacità di usare un insieme di simboli (le lettere dell'alfabeto) e un insieme di regole (lessicali, grammaticali e logiche) per combinarli a formare parole e frasi.



Criterio di similitudine

- Fino a qualche anno fa, nessun programma ha superato il test di Turing. Il più noto è Eliza, un programma scritto nel 1966 da Joseph Weizenbaum. Eliza è una psicoterapeuta che simula una conversazione tra lei (il medico), e voi (il paziente). Il programma non era molto convincente; tuttavia, ai primordi dei computer domestici molte persone erano convinte che un computer fosse un "cervello" elettronico e quindi non facevano molto caso alla piega bizzara che ben presto delineava la "seduta". D'altra parte, il test di Turing non prevedeva l'ingenuità della persona incaricata di saggiare la macchina: doveva essere un operatore esperto.
- Dopo Eliza sono stati realizzati molti programmi per simulare l'intelligenza; sebbene alcuni siano progettati per argomenti ben definiti (per es. teatro di Shakespeare), nessuno è stato in grado di ingannare un giudice esperto.
- Ma qualche mese fa ChatGPT-4 pare abbia superato questo test, fonte: Mei, Q. et al., "A Turing test of whether AI chatbots are behaviorally similar to humans", PNAS (2024)Stanford University

In conclusione

- Concludiamo menzionando che la Macchina di Turing “non può capire se stessa”. Si può dimostrare, infatti, che una macchina di Turing (MdT) non può calcolare fino in fondo il comportamento di un’altra MdT o di se stessa.
- In altre parole, se trasformiamo anche la MdT in una serie di simboli da dare in pasto ad una MdT, non esiste una MdT in grado di assicurarci che la MdT in ingresso faccia bene il proprio lavoro.

Il problema della fermata

- Abbiamo una macchina di Turing qualsiasi M e le diamo un input.
- Vogliamo sapere se essa si ferma oppure procede all’infinito.
- Supponiamo ora di costruire una macchina di Turing MF che, dati in ingresso i dati e le istruzioni della generica M, calcola se essa si ferma oppure no.
- Vedremo che questa MdT non esiste.
- Costruiamo ora una seconda macchina MFL uguale a MF eccetto che in un particolare: dopo aver scoperto che la M si ferma questa esegue un loop infinito.
- Se esiste MF allora anche MFL si può facilmente costruire.
- Supponiamo di prendere coma macchina generica M = MFL cioè chiediamo alla MFL di calcolare se essa stessa si ferma oppure no.
- Allora i casi sono due:
    - a) se MFL si ferma vuol dire che MFL ha scoperto che MFL non si ferma
    - b) se MFL non si ferma allora vuol dire MFL ha scoperto che MFL si ferma
- in entrambe i casi siamo arrivati a una conclusione assurda.