Macchine Teoriche e Architettura di base

INDICE

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

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.
La definizione data ricollega quindi la nozione di computer a quella di algoritmo

Computer ⟺ **Algoritmo**

che possiamo definire come un insieme di istruzioni, scritte, date, …, codificate in un certo linguaggio, che stabilisce come si deve effettuare l'esecuzione di un certo lavoro.
Vediamo qualche esempio.
Supponiamo di voler cucinare una pietanza. Descriviamo i passi necessari per compiere questo "lavoro":

I 1 Prendere una pentola
I 2 Mettere mezzo bicchiere di olio di oliva nella pentola
I 3 Mettere due agli spezzettati nella pentola
I 4 Mettere la pentola sul fuoco fino a che l'aglio è dorato
I 5 Aggiungere 500 gr di pomodori nella pentola
I 6 Cuocere per 15 minuti
I 7 Aggiungere 8 foglie di basilico nella pentola
I 8 Cuocere per 15 minuti
I 9 Prendere una pentola più grande
I 10 Riempirla d'acqua
I 11 Metterci un pugno di sale
I 12 Mettere la pentola sul fuoco fino a che l'acqua non bolle
I 13 Buttare 500 gr di spaghetti nella pentola
I 14 Cuocere per 9 minuti
I 15 Scolare gli spaghetti
I 16 Mescolare gli spaghetti al sugo della prima pentola

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).

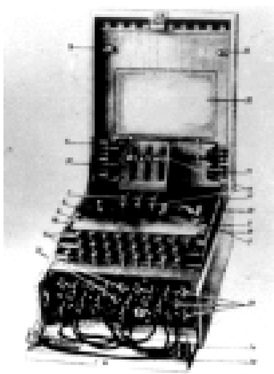
La macchina di Turing e la macchina di von Neumann sono due modelli di calcolo (ovvero; modi di definire e/o eseguire algoritmi) specifici. In breve, stiamo definendo il concetto di algoritmo come tutto ciò che può essere eseguito dai due tipi di macchine su menzionate.
Tra tutti i modelli di calcolo 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. Ne parleremo nella sezione I..
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 computers sono strutturati (ancora) come una macchina di Von Neumann. Definiremo la macchina di von Neumann nella sezione IV..

I. LA 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.


Alan Turing

Alan Turing nacque a Londra nel 1912 e morì suicida nel 1954 all'età di quarantadue anni, dopo aver mangiato una mela intinta nel cianuro. La biografia di Turing è appassionante quanto le sue ricerche perché egli era un eccentrico, un ateo anticonformista che mal si adeguava alle regole dell'Inghilterra conservatrice della prima metà del secolo ed era omosessuale, in tempi in cui l'omosessualità era un reato punibile con la prigione. Fu lui, negli anni '30, a coniare l'espressione "computer elettronico", e a porre le basi del concetto di intelligenza artificiale. E le conseguenze pratiche dei suoi studi hanno influito addirittura sull'esito del conflitto mondiale. Nel 1939 andò infatti a dirigere Bletchley Park, il quartier generale britannico per la comunicazione cifrata, in cui lavoravano matematici, linguisti, esperti di gioco d'azzardo ed enigmisti. L'obiettivo era la decodifica di "Enigma", il codice di comunicazione della Marina tedesca, che aveva resistito a ogni tentativo di decifrazione. Egli riuscì a decodificare il codice di trasmissione tedesco permettendo così alla marina inglese di decifrare tutte le comunicazioni nemiche. Dopo la guerra tentò inutilmente di dare all'Inghilterra un altro vantaggio sostanziale, con la costruzione di un cervello elettronico universale (un computer moderno), basato sugli studi teorici che egli aveva effettuato da studente, a ventitré anni, inventando


La macchina "Enigma"

quella che oggi si chiama "macchina di Turing". A titolo di ringraziamento per i suoi servizi l'Inghilterra dapprima lo decorò con l'Ordine dell'Impero Britannico, poi lo fece membro della Royal Society, ed infine lo processò nel 1952 per atti osceni in quanto omosessuale, condannandolo ad una cura ormonale che lo rese impotente.
Il nome di Turing è oggi famoso soprattutto per la sua analisi del processo di computazione in termini di macchine astratte, dette appunto "macchine 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.

...																...

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)

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.
La macchina si ferma quando raggiunge uno stato finale.
Il risultato calcolato dalla macchina di Turing è la sequenza dei simboli segnati sul nastro che sono conseguenza di tutti i cambiamenti di stato.

È abbastanza sorprendente che un dispositivo semplice come la macchina di Turing rappresenti il più potente strumento di calcolo conosciuto, nel senso che per ogni problema per cui è nota una procedura di soluzione è possibile formulare un algoritmo eseguibile da una macchina di Turing. Malgrado l'estrema semplicità di questa macchina, essa risulta capace (con un adatto assegnamento di istruzioni) di risolvere problemi di grande complessità, ma la sua importanza non sta in tale capacità ma nell'essere uno strumento concettuale che permette di definire rigorosamente gli algoritmi e di ottenere risultati di grande generalità. Le ricerche finora condotte fanno pensare che qualsiasi algoritmo possa essere realizzato mediante una macchina di Turing. L'accettazione di questa ipotesi ci porta a considerare la teoria degli algoritmi come coincidente con la teoria della macchina di Turing.

La domanda che si pone il matematico britannico appartiene ad un classico dominio di indagine del pensiero occidentale: che cosa significa calcolare?
L'indagine compiuta da Turing, dunque, prende le mosse dall'analisi dell'attività umana del calcolare, con particolare riguardo al processo di calcolo, da cui nasce l'idea stessa di 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.

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.

L'analisi di Turing riguarda i processi di calcolo eseguibili da un essere umano (U) idealizzato, cioè le azioni che U compie mentre computa, e si basa sull'individuazione di alcune condizioni di finitezza, giustificate dal fatto che memoria e percezione, coinvolte nel processo di calcolo, hanno dei limiti, e di condizioni di determinatezza, giustificate dalla richiesta che l'attività di calcolo sia determinata da una procedura (cioè non permetta scelte arbitrarie).
Ipotizziamo un calcolo su un nastro (avviene quindi in uno spazio unidimensionale) finito, ma estendibile a piacere.

Diamo le **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:
- **a)** Cambiare il contenuto di alcune caselle osservate
- **b)** Cambiare le caselle osservate (cioè spostare l'attenzione da una all'altra)
- **c)** Cambiare il proprio stato mentale (cioè quello che ricorda del calcolo)
- **d)** Osservare nuove caselle che si trovano al massimo ad una distanza prefissata L da una qualsiasi delle caselle osservate.

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.

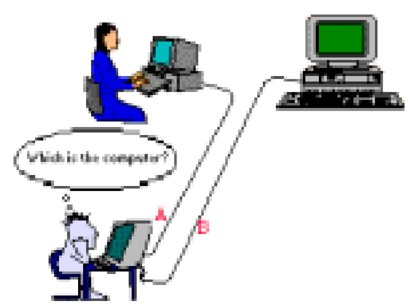
Ad una macchina di Turing associamo un procedimento di calcolo idealizzato, cioè supponiamo di non avere limiti di spazio (il nastro su cui il calcolo avviene è potenzialmente infinito) e di tempo, e che la macchina non commetta errori. Chiamiamo infine UNIVERSALE una macchina di Turing in grado di calcolare 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.

Resta da vedere se le condizioni date si applicano a qualsiasi processo di calcolo effettivamente eseguibile; La **tesi di Turing** afferma che "*ogni funzione parziale calcolabile con un algoritmo è una funzione parziale calcolabile da una macchina di Turing*."

Il lavoro svolto da una particolare macchina di Turing in funzione può quindi essere descritto come la trasformazione secondo regole delle informazioni depositate nella sua memoria.
La possibilità di identificare ogni macchina di Turing con il suo programma, cioè con quelle particolari informazioni che stabiliscono l'elenco delle operazioni da compiere, garantisce l'esistenza della macchina calcolatrice generica. Tuttavia, solo quando la tecnologia mise a disposizione una sufficiente quantità di memoria fu però possibile realizzare concretamente una macchina calcolatrice programmabile. La macchina di Turing è quindi una macchina che elabora informazione in senso generico, e la specifica elaborazione compiuta in un caso particolare è completamente determinata dal programma che in quel caso ne detta le regole. Ciò significa che una macchina di Turing può fare tutto ciò che può essere descritto in un programma, presentandosi così come una entità dal comportamento estremamente flessibile.

II. 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?” 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. 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.



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.

Tanto gli stati interni che i segnali di ingresso e di uscita sono insiemi di concatenazioni di simboli; gli stati interni sono gli insiemi di proposizioni che esprimono le conoscenze possedute dall'essere umano, i segnali di ingresso sono le domande rivolte dall'interlocutore e i segnali di uscita sono le eventuali risposte date dall'essere umano in base alle sue conoscenze. Se è noto il variare dello stato interno in dipendenza dal segnale di ingresso, e per ogni segnale di ingresso è possibile stabilire l'eventuale segnale di uscita, a patto di inserire questi dati in maniera opportuna in due tabelle analoghe a quelle raffigurate da Turing, è anche possibile programmare una macchina di Turing affinché simuli un essere umano. Un esempio tipico di

macchina simulatrice potrebbe una macchina che gioca esclusivamente a scacchi…
Finora 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.

Turing pensò una macchina astratta per fare questa elaborazione, una macchina che non è mai stata effettivamente costruita, sarebbe troppo scomoda da usare, che però è servita a dimostrare le qualità ed i limiti dei calcolatori reali di più di qualsiasi calcolatore reale.
Il fatto che tutte macchine sono casi particolari della macchina di Turing non è possibile dimostrarlo, ma è un principio, noto come **Tesi di Church**, che si è verificato vero in tutti i casi di macchine finora costruiti o anche solo pensati. Questa tesi, formulata nel 1936, all'indomani dell'ideazione della macchina stessa, afferma: "*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.

Le implicazioni di questa asserzione sono di notevole portata, infatti ne consegue che:

- le basi teoriche dell'informatica si riducono ad un solo argomento (con i vari punti di vista);
- è inutile tentare di costruire macchine diverse, hanno tutte gli stessi limiti
- l'importanza dell'informatica non sta nella teoria ma nelle applicazioni

Queste conclusioni hanno creato molte delusioni: per i meccanicisti (per chi pensa al mondo come una macchina) è difficile capire come spiegare le cose che la macchina di Turing non può fare, per chi non è meccanicista è difficile restringere il non meccanico a ciò che la macchina di Turing non può fare.

Certamente la MdT non è stata creata per risolvere problemi del mondo reale ma per essere sufficientemente semplice per dimostrare con essa le proprietà del calcolo automatico.
La prima di queste proprietà è che la macchina di Turing è **equivalente** come possibilità a quella di qualsiasi calcolatore con memoria illimitata. La dimostrazione è stata fatta per tutti i casi di macchine note e per tutte le procedure generali di calcolo inventate.
La seconda di queste proprietà è che una macchina di Turing rappresenta il **riconoscitore** del tipo di linguaggio più generale.
La terza proprietà è che la MdT rappresenta una **definizione formale** del concetto di Algoritmo. Le Tavole di una MdT rappresentano in effetti la forma più generale di algoritmo possibile. Quando analizziamo un algoritmo, cioè una procedura generale di soluzione, lo dobbiamo sempre fare facendo riferimento ad una architettura di calcolo concreta. Dobbiamo usare, in altre parole, passi dell'algoritmo che siano effettivamente realizzabili.
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. Infatti, vediamo il famoso **problema della fermata**.
Abbiamo una macchina di Turing qualsiasi M e le diamo un ingresso. 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.

III. JOHN VON NEUMANN


John Von Neumann

Von Neumann nacque ebreo ed ungherese a Budapest il 28 dicembre 1903 come Janos Neumann, e morì cattolico e statunitense a Washington l'8 febbraio 1957 come John Von Neumann. La sua morte precoce fu l'effetto di un contrappasso, dovuto ad un cancro alle ossa contratto per l'esposizione alle radiazioni dei test atomici di Bikini nel 1946, la cui sicurezza per gli osservatori egli aveva tenacemente difeso.
Anche lui fu un personaggio bizzarro e controverso, sebbene geniale, criticato e malvisto da molti.
Fu accanito guerrafondaio e lo scienziato con il maggiore potere politico negli Stati Uniti.
Si occupò di matematica, informatica, logica, fisica, economia, politica.
Fu un bambino prodigio: a sei anni conversava con il padre in greco antico; a otto conosceva l'analisi; a dieci aveva letto un'intera enciclopedia storica.
Da studente, frequentò contemporaneamente le università di Budapest e Berlino, e l'ETH di Zurigo.
Gli piaceva dare feste sfavillanti, guidare pericolosamente, bere e mangiare molto.
Nel 1937, appena ricevuta la cittadinanza statunitense, iniziò ad interessarsi di problemi matematici "applicati". Divenne rapidamente uno dei maggiori esperti di esplosivi, e si impegnò in un gran numero di consulenze militari, soprattutto per la marina.
Il suo risultato più famoso nel campo fu la scoperta che le bombe di grandi dimensioni sono più devastanti se scoppiano prima di toccare il suolo, a causa dell'effetto addizionale delle onde di detonazione. L'applicazione più infame del risultato si ebbe il 6 e 9 agosto del 1945, quando le più potenti bombe della storia detonarono sopra il suolo di Hiroshima e Nagasaki, all'altezza calcolata da Von Neumann affinché esse producessero il maggior danno aggiuntivo.
Questo non fu comunque l'unico contributo di Von Neumann alla guerra atomica. Egli proseguì imperterrito e divenne, assieme a Teller, il convinto padrino del successivo progetto di costruzione della bomba all'idrogeno (che fu approvato da Truman nonostante la raccomandazione contraria dell'apposito comitato presieduto da Oppenheimer, il quale pensava che gli scienziati avessero già fatto abbastanza male all'umanità).
Nell'agosto 1944, venuto a conoscenza dell'esistenza del calcolatore elettronico ENIAC (utilizzato per complessi calcoli balistici e per le tavole di tiro di armamenti sempre più sofisticati) Von Neumann si buttò a capofitto nel progetto e nel giro di quindici giorni dalla sua entrata in scena, il progetto del calcolatore veniva modificato in modo da permettere la memorizzazione interna del programma. La programmazione, che fino ad allora richiedeva una manipolazione diretta ed esterna dei collegamenti, era così ridotta ad un'operazione dello stesso tipo dell'inserimento dei dati, e l'ENIAC diveniva la prima realizzazione della macchina universale inventata da Alan Turing nel 1936: in altre parole, un computer programmabile nel senso moderno del termine.
Presso l'Istituto di Princeton egli si dedicò alla progettazione di un nuovo calcolatore, producendo una serie di lavori che portarono alla definizione di quella che oggi è nota come architettura Von

Neumann: in particolare, la distinzione tra memoria primaria (ROM) e secondaria (RAM), e lo stile di programmazione mediante diagrammi di flusso.

IV. LA 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:

- **ZERO**: $N \to N$

$n \to \text{ZERO}(n) = 0$

- **INC**: $N \to N$

$n \to \text{INC}(n) = n + 1$

- **SOM**: $N \times N \to N$

$(n, m) \to \text{SOM}(n, m) = n + m$
(il risultato della somma)

- **SOT**: $N \times N \to N$

$(n, m) \to \text{SOT}(n, m) = \begin{cases} n - m & \text{se } n \geq m \\ 0 & \text{se } n < m \end{cases}$

- **MOL**: $N \times N \to N$

$(n, m) \to \text{MOL}(n, m) = n \times m$

- **DIV**: $N \times N \to N$

$(n, m) \to \text{DIV}(n, m) = [n / m]$
(prende il quoziente intero della divisione)

- **UGUALE**: $N \times N \to N$

$(n, m) \to \text{UGUALE}(n, m) = \begin{cases} 1 & \text{se } n = m \\ 0 & \text{se } n \neq m \end{cases}$

- **MINORE**: $N \times N \to N$

$(n, m) \to \text{MINORE}(n, m) = \begin{cases} 1 & \text{se } n < m \\ 0 & \text{se } n \geq m \end{cases}$

- **SALCOND**: $N \times J \to$ Azione

$(n, j) \to \text{SALCOND}(n, j) = \begin{cases} \text{salta a } j & \text{se } n \neq 0 \\ \text{non salta a } j & \text{se } n = 0 \end{cases}$

essendo $J = \{j \in N : 0 \leq j < |P| - 1\}$ un insieme finito di indici.

- **ALT**: ferma la computazione 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.

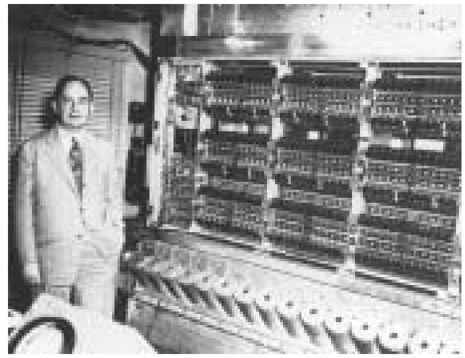
V. IL FUNZIONAMENTO DELLA MACCHINA DI VON NEUMANN

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)

A fianco di Von Neumann l'elaboratore IAS di Princeton o ***macchina di Von Neumann****, ultimata nel 1952. Nella parte bassa dell'elaboratore si può vedere una fila di tubi catodici (CRT) utilizzati come memoria già nel 1948 da F.C.Williams nell'elaboratore inglese Mark 1 a Manchester. Come con i tubi catodici si può definire un punto in uno schermo, con lo stesso principio anche il bit veniva definito con un impulso. Nel 1955 tali memorie vengono sostituite da quelle a nuclei di ferrite.*



Il processore è costituito da 4 parti fondamentali:

a) il contatore di programma, una locazione di memoria contenente l'indirizzo dell'istruzione da eseguire
b) il registro delle istruzioni, una locazione di memoria contenente l'istruzione da eseguire
c) l'unità aritmetica e logica, un sistema che esegue l'istruzione
d) il controllo, un sistema che, attraverso una sequenza di cambiamenti di stato, fa avvenire l'esecuzione dell'istruzione, dunque ha il comando del processo.

La computazione della macchina avviene eseguendo le istruzioni del programma nell'ordine definito dal programma, a meno di sati 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(M_1, M_2) M_1 e M_2 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(M_1, M_2) il risultato viene registrato nella locazione di memoria il cui indrizzo è M_1. 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;
ALT) Il ciclo si ripete fino a che non si incontra l'istruzione ALT.

<u>Esempio 1.</u> Realizzare il programma per la macchina di Von Neumann che calcola se un numero è divisibile per 3 (il valore è registrato in M_0; se è divisibile pre tre scrivere in M_1 il valore 1 altrimenti scrivere 0).

0	ZERO(M_1)	Scrive 3 in M_1
1	INC(M_1)	
2	INC(M_1)	
3	INC(M_1)	
4	ZERO(M_2)	Copia il contenuto di M_0 in M_2
5	SOM(M_2,M_0)	
6	DIV(M_2,M_1)	Calcola r = n–3(n/3) e mette il risultato in M_0
7	MOL(M_2,M_1)	

8	SOT(M_0,M_2)	
9	SALCOND(M_0, 13)	Se r≠0 salta a 13
10	ZERO(M_1)	Scrive 1 in M_1 e si ferma
11	INC(M_1)	
12	ALT	
13	ZERO(M_1)	Scrive 0 in M_1 e si ferma
14	ALT	

Esempio 2. Realizzare il programma per la macchina di Von Neumann che calcola se la somma di due numeri è maggiore di 5 (i valori sono registrati in M_0, M_1; se la somma è maggiore di 5 scrivere in M_2 il valore 1 altrimenti scrivere 0).

0	ZERO(M_2)	Scrive 5 in M_2
1	INC(M_2)	
2	INC(M_2)	
3	INC(M_2)	
4	INC(M_2)	
5	INC(M_2)	
6	ZERO(M_3)	Somma il contenuto di M_0 ed M_1 in M_3
7	SOM(M_3,M_0)	
8	SOM(M_3,M_1)	
9	MINORE(M_2,M_3)	Se $M_2 < M_3$ scrive 1 in M_2, altrimenti scrive 0
10	ALT	

Esercizi. Realizzare i seguenti programmi per la macchina di Von Neumann:

a) calcolare la media aritmetica (approssimata al numero intero) di tre numeri (i valori sono registrati in M_0, M_1, M_2; la media risultante va registrata in M_3);

b) dati due numeri, dividere il più grande per il più piccolo e prendere il resto intero della divisione (i valori sono registrati in M_0, M_1; il resto della divisione va registrato in M_3).

Per approfondimenti si possono consultare i seguenti siti Internet.

Turing:

Sito personale http://www.turing.org.uk/turing/

Biografia http://ei.cs.vt.edu/~history/Turing.html

Macchina di Turing http://riemann.unica.it/attivita/colloquium/greco/node11.html

Test di Turing http://www.oz-inc.com/ov/agents/main_index.html

La macchina Enigma http://www.riksoft.com/indexok.asp?Goto=crienig.htm

Von Neumann:

Biografia http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Von_Neumann.html

Architettura di von Neumann http://www.unibg.it/ctd/elab/hardware/archite.htm