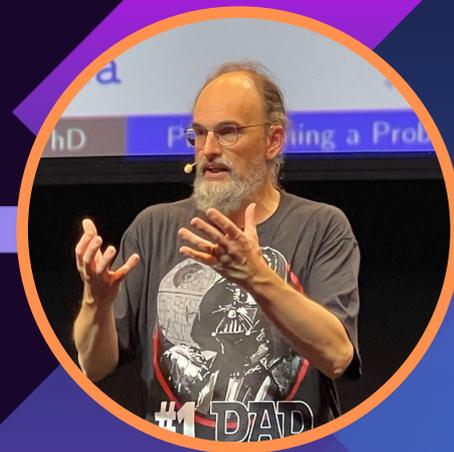


# Numeri e simboli: fenomenologia dell'AI di ieri, oggi e domani

Paolo Caressa

IT Expert @ GSE

External Researcher @ M&MoCS



>> AI CONF 2025

# Agenda

---

- Breve storia dell'AI
- Numeri
- Simboli
- Fenomenologia dell'AI contemporanea

# Breve storia dell'AI

---

If it's machine learning, it's probably written in Python. If it's AI, it's probably written in PowerPoint

AI simbolica

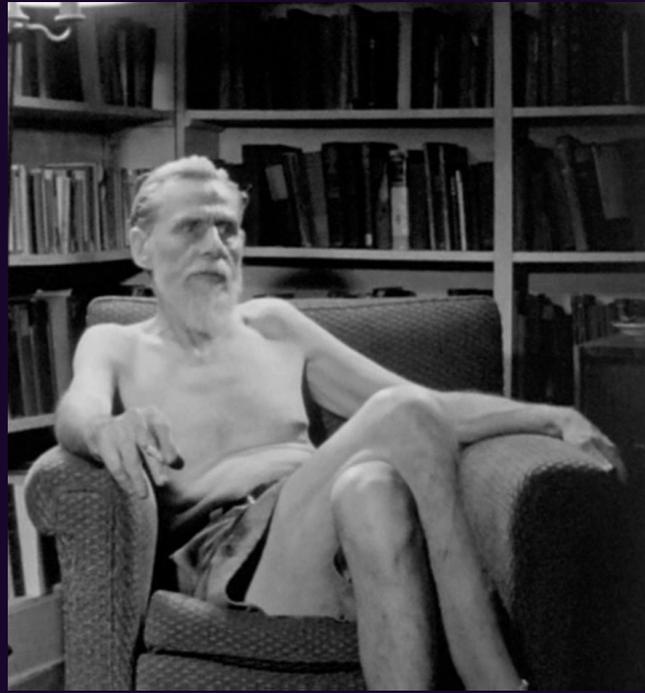
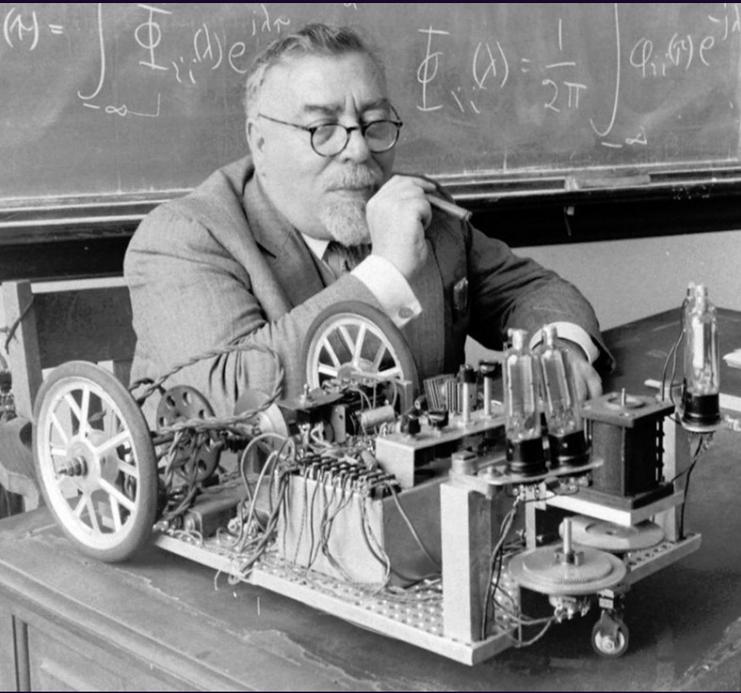
AI numerica

AI predittiva

AI generativa

AI auto!

# Precursori dell'AI



# L'inventore dell'AI



# L'infanzia del ML (1950)

Invece che scrivere un programma per computer per simulare una mente adulta, perché piuttosto non provare a produrne uno che simuli la mente di un bambino? Se questo fosse soggetto a un'appropriata educazione si otterrebbe la mente di un adulto. [...] La nostra speranza è che il meccanismo nel cervello del bambino sia così piccolo da poter essere programmato. La mole di lavoro nell'educazione la possiamo assumere, in prima approssimazione, pari a quella del bambino umano.

# Applicazioni dell'AI (1948)

Ci dobbiamo confrontare col problema di trovare opportune branche del pensiero in cui la macchina possa esercitare le proprie capacità. I seguenti campi mi sembrano avere dei vantaggi:

- Giochi vari, come scacchi, tris, bridge, poker
- L'apprendimento del linguaggio
- La traduzione del linguaggio
- La crittografia
- La matematica

# Reinforcement Learning (1950)

L'uso di premi e punizioni è nel migliore dei casi solo una parte del processo di apprendimento. Detto brutalmente, **se l'insegnante non ha altri mezzi di comunicazione con l'allievo, la quantità di informazioni che può raggiungerlo non eccede il numero totale di premi e punizioni applicati. [...]**

È quindi necessario avere altri canali "non emozionali" di comunicazione [...] **sarebbe possibile insegnare a una macchina con premi e punizioni a obbedire a ordini espressi in un linguaggio simbolico.**

# Il ruolo del caso (1951)

È probabilmente giusto inserire elementi casuali in una macchina che apprende. Un elemento di casualità è utile quando cerchiamo la soluzione a un problema. Supponiamo per esempio di voler trovare un numero fra 50 e 200 pari al quadrato della somma delle sue cifre, potremmo partire con 51, e poi provare 52 e così via fino a trovare un numero che vada bene. Altrimenti, potremmo scegliere un numero a caso fra 50 e 200 fino a che non ne estraiamo uno che vada bene. [...] Il metodo sistematico ha lo svantaggio che potrebbe esserci un enorme blocco senza soluzioni nella regione che per prima viene analizzata. [...] Poiché c'è probabilmente un gran numero di soluzioni soddisfacenti, il criterio casuale sembra migliore di quello sistematico.

# La fase simbolista dell'AI



# Un'estate lunga (almeno) 70 anni

Proponiamo che si porti avanti uno studio dell'intelligenza artificiale per due mesi, dieci persone, nell'estate del 1956 [...] Lo studio deve procedere sulla base della **congettura che ogni aspetto dell'apprendimento o di altre caratteristiche dell'intelligenza possa in principio essere descritto in modo così preciso che una macchina possa simularlo**. Sarà fatto un tentativo di capire come far usare il **linguaggio** alle macchine, formare **astrazioni e concetti**, risolvere il tipo di problemi ora appannaggio degli esseri umani, **migliorare se stesse**. Pensiamo che un progresso significativo possa esser fatto in uno o più di questi problemi se un gruppo di scienziati ben scelto ci lavori assieme per una estate.

# Vita da simbolista

Studi logica matematica, in particolare il calcolo dei predicati.

Conosci la teoria della calcolabilità e le regole di risoluzione.

Scrivi in un dialetto del Lisp o in Prolog.

Navighi dello spazio degli stati come un lupo di mare.

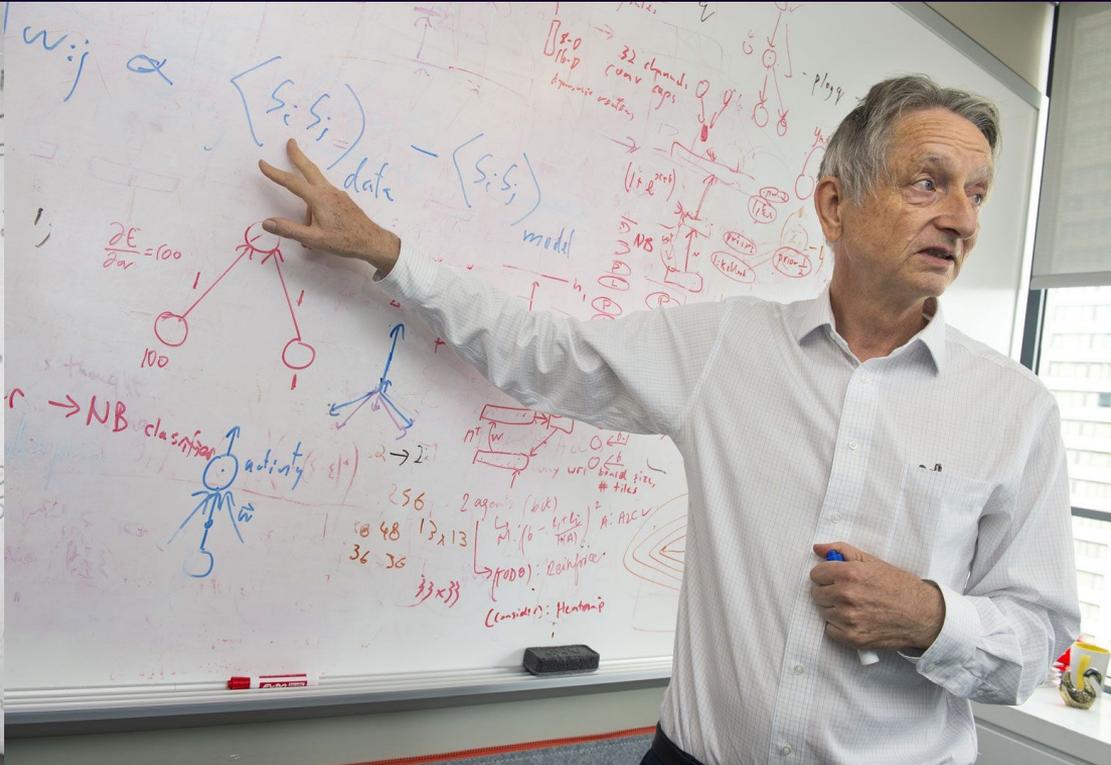
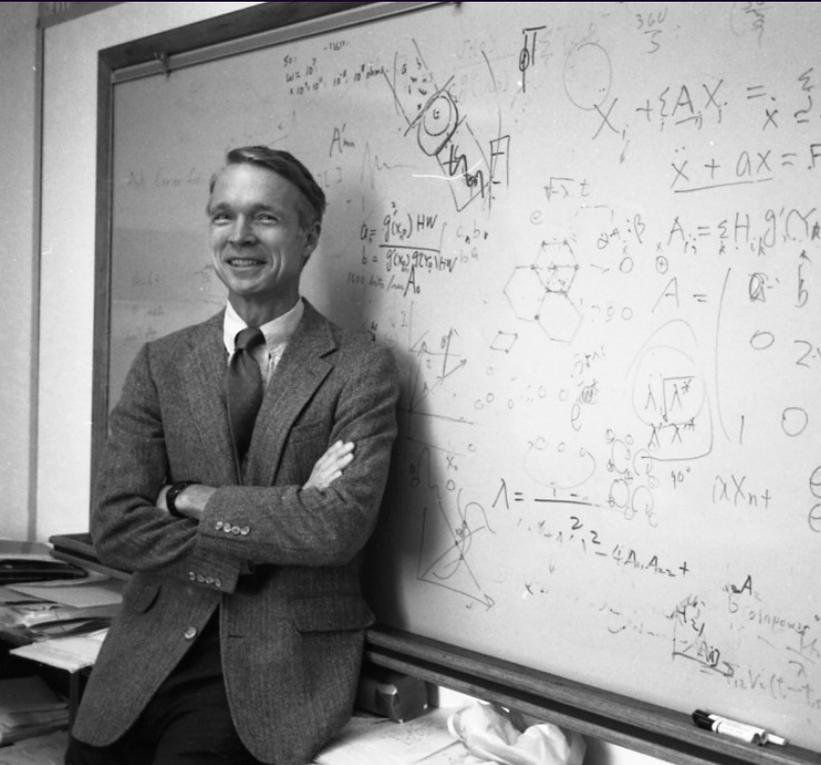
Sai pronunciare e magari anche implementare «euristica».

Conosci Bayes, ma quando lo incontri cambi marciapiede.

Per te i numeri sono solo interi.

Per te il linguaggio è grammatica allo stato puro.

# La fase connettivista dell'AI



# Hinton sulla riva del fiume (1985)

Le nostre attuali simulazioni sono lente per tre ragioni: è inefficiente simulare reti parallele con macchine seriali, ci vogliono molte decisioni da ciascuna unità prima che **una rete grande raggiunga l'equilibrio** e ci vuole un numero sterminato di coppie I/O prima che **una rete possa capire cosa rappresentare nelle sue unità interne**. Un hardware migliore potrebbe risolvere il primo problema, ma per gli altri due sono necessari ulteriori progressi teorici. Solo allora saremo in grado di applicare questo tipo di reti che apprendono a problemi più realistici.

# Vita da connettivista

Hai studiato algebra lineare, calcolo differenziale e ottimizzazione (in pratica un biennio STEM).

Ti interessi di qualsiasi cosa che abbia la sottostringa «neuro» nel nome.

Una volta nella vita hai implementato a mano la back propagation per le reti neurali.

Potresti scrivere in C++, ma è già tutto pronto in Python.

Conosci Bayes, ma gli preferisci Markov.

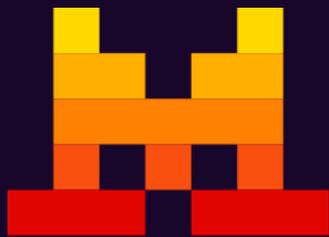
Per te i numeri sono float IEEE, al limite quantizzati.

Per te il linguaggio è solo una questione di parole.

# La fase generativa dell'AI

 Gemini

 Claude



 deepseek

# Vita da generativista

Hai studiato tutti i paper (attention etc.) di sistemi che non riuscirai mai a implementare.

Leggi compulsivamente a caso ogni paper pubblicato.

Ti attacchi a qualsiasi cosa tu possa effettivamente implementare: RAG, MCP, agenti etc.

Scrivi in Python perché è consuetudine, ma di nascosto programmi in Rust.

Non ti interessa conoscere né Bayes né Markov.

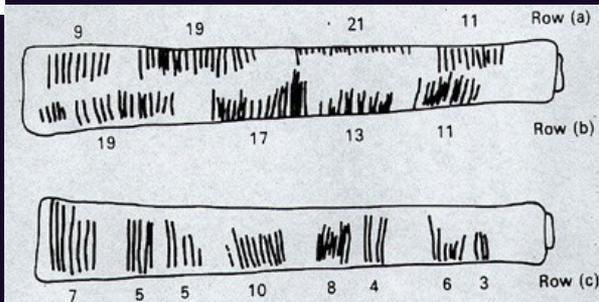
Per te i numeri sono pesi dentro gli LLM.

Per te il linguaggio è quello prodotto da un LLM.

# Numeri



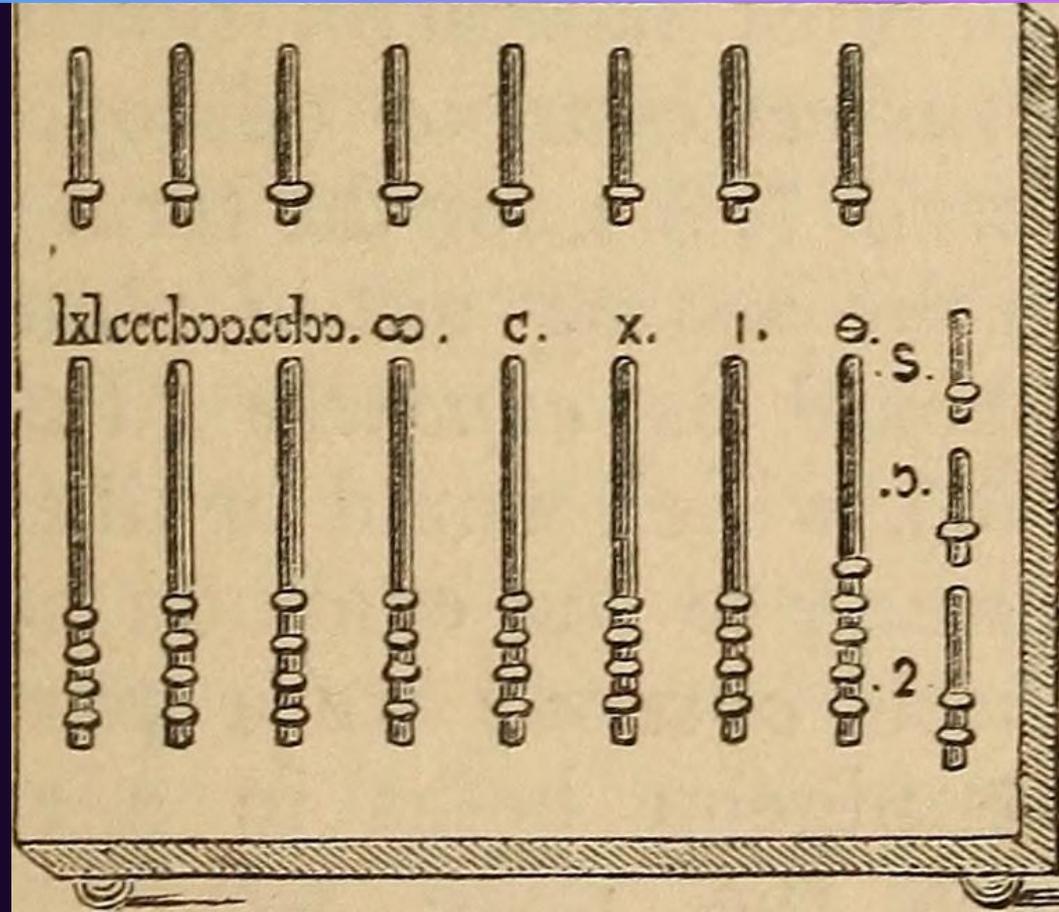
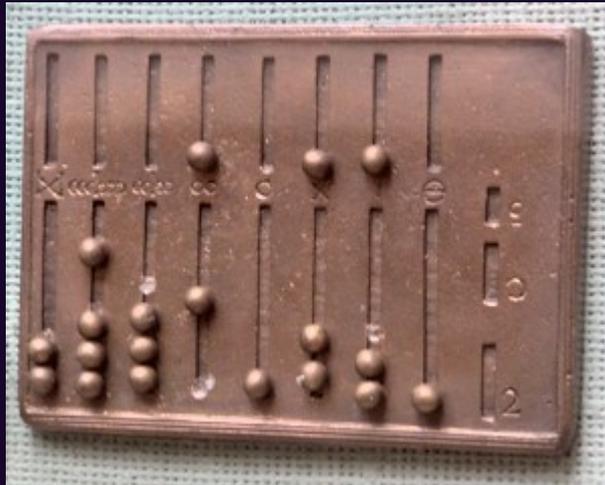
# La preistoria dei numeri



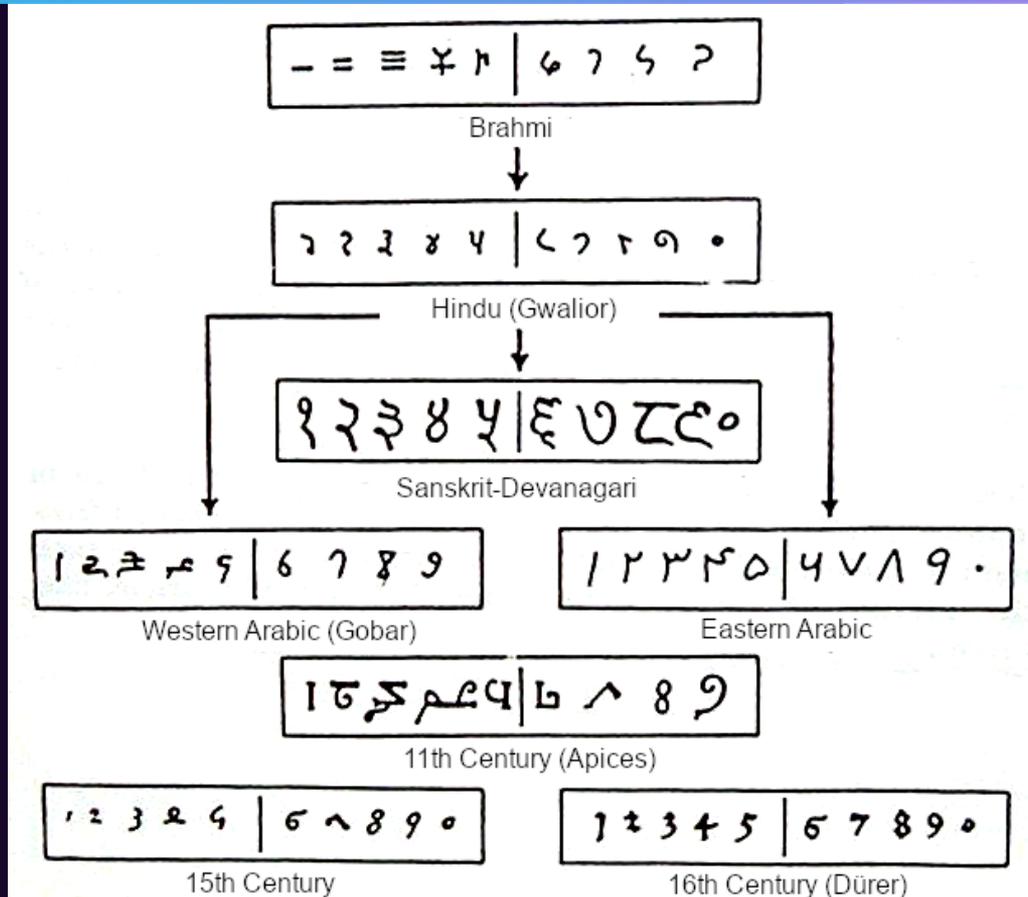
# Il minimo per amministrare un impero



# Altro che cifre romane!



# Genealogia delle cifre moderne



# I numeri sono stringhe di cifre?!?

```
def bin_sum(x: str, y: str) -> str:
    while len(x) < len(y): x = "0" + x
    while len(y) < len(x): y = "0" + y
    s, r = "", "0"
    for c in range(len(x) - 1, -1, -1):
        a, r = bin_dig_sum(x[c], y[c], r)
        s = a + s
    return s
```

```
def bin_dig_sum(x: str, y: str, z: str) -> tuple[str, str]:
    if x+y+z == "000": return "0", "0"
    if x+y+z in ["001", "010", "100"]: return "1", "0"
    if x+y+z in ["011", "101", "110"]: return "0", "1"
    return "1", "1"
```

```
print(bin_sum("1001", "11"))
```

Cioè, finora abbiamo parlato di simboli?



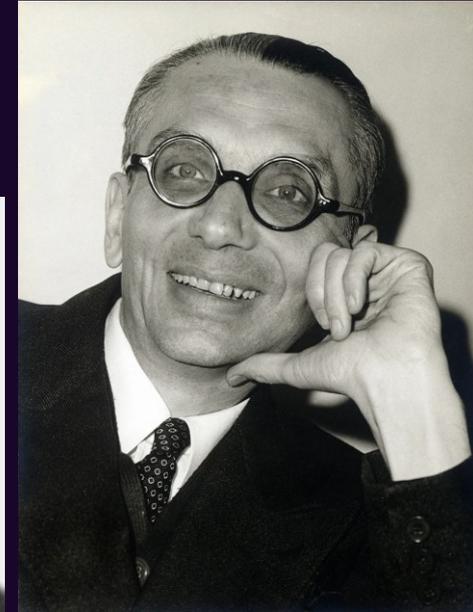
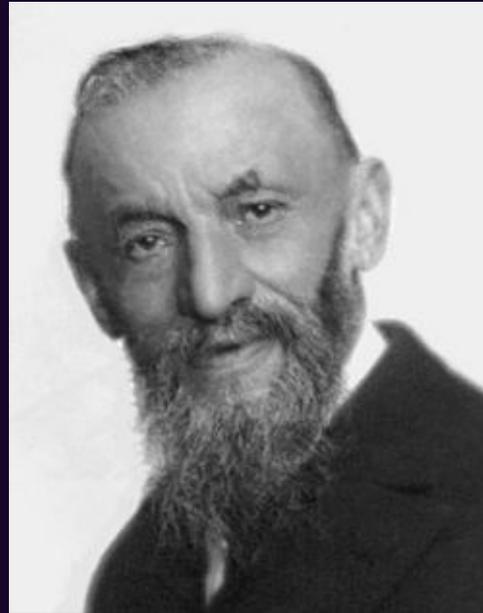
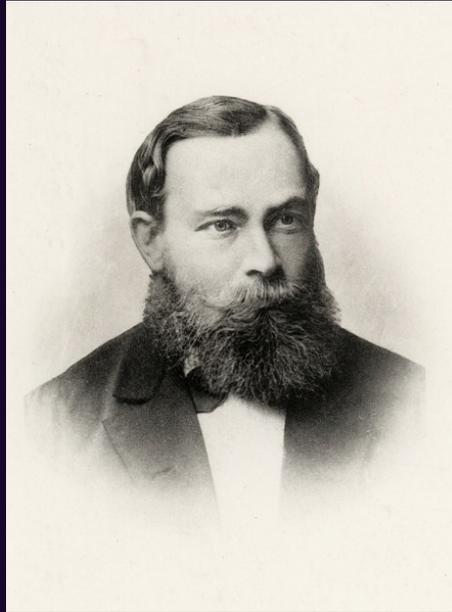
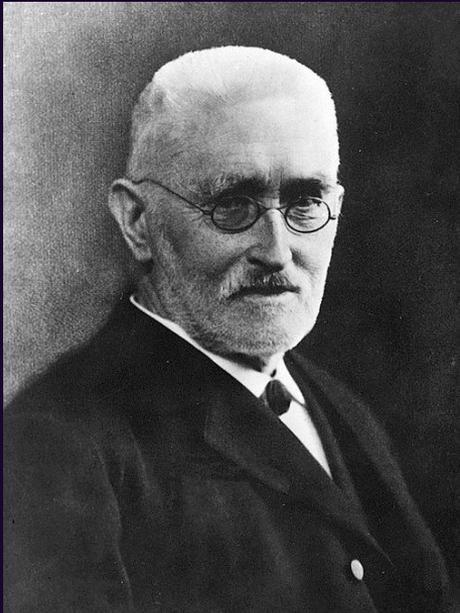
# Una città può avere quattro lettere?



Un numero si chiama **immane** se in lingua italiana non si riesce a scrivere a parole con meno di 20 sillabe.

Sia ora  $n$  il più piccolo numero immane...

# Numeri... che roba sono?

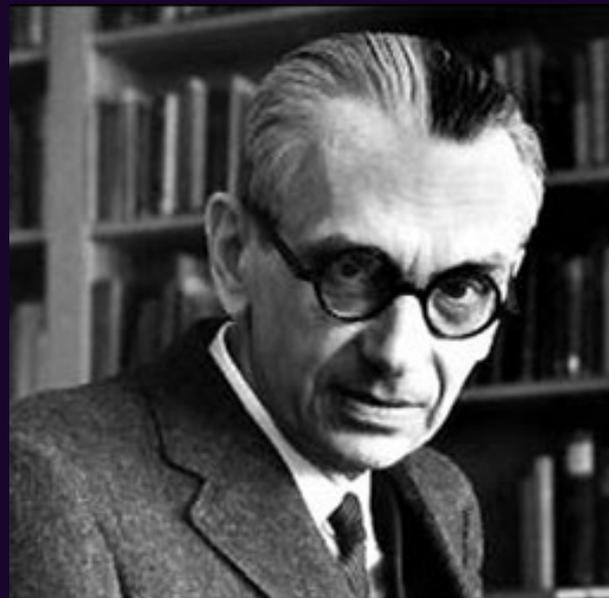


# Numeri che codificano simboli

Sia  $B(n, m)$  vera sse  $n = |P|$   
codifica una dimostrazione  $P$  per  
una formula  $F$  di numero  $m = |F|$ .

Sia  $D(n)$  vera sse  $\exists n B(n, m)$ .

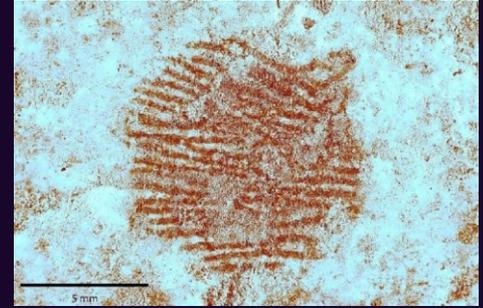
È dimostrabile  $D(|\neg D|)$ ? Essa  
afferma “non sono dimostrabile”  
e non può esserlo, altrimenti  
potremmo dimostrare tanto  $D$   
quanto  $\neg D$ . Quindi è vera!



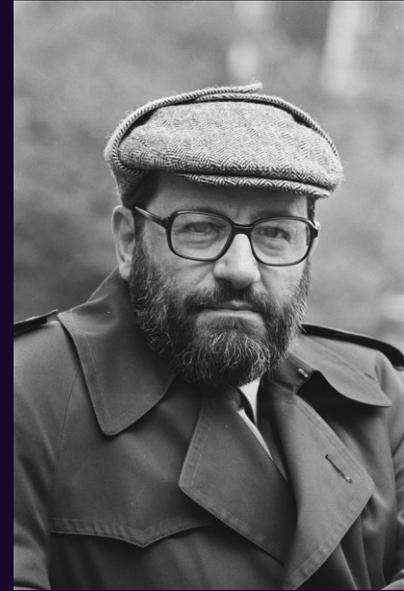
# Simboli



# Homo symbolicum



# Simboli... che roba sono?



# Animal symbolicum

Il simbolo non è il rivestimento meramente accidentale del pensiero, ma il suo organo necessario ed essenziale. Esso non serve soltanto allo scopo di comunicare un contenuto concettuale già bello e pronto ma è lo strumento in virtù del quale questo stesso contenuto si costituisce ed acquista la sua compiuta determinatezza. **L'atto della determinazione concettuale di un contenuto procede di pari passo con l'atto del suo fissarsi in qualche simbolo caratteristico.**

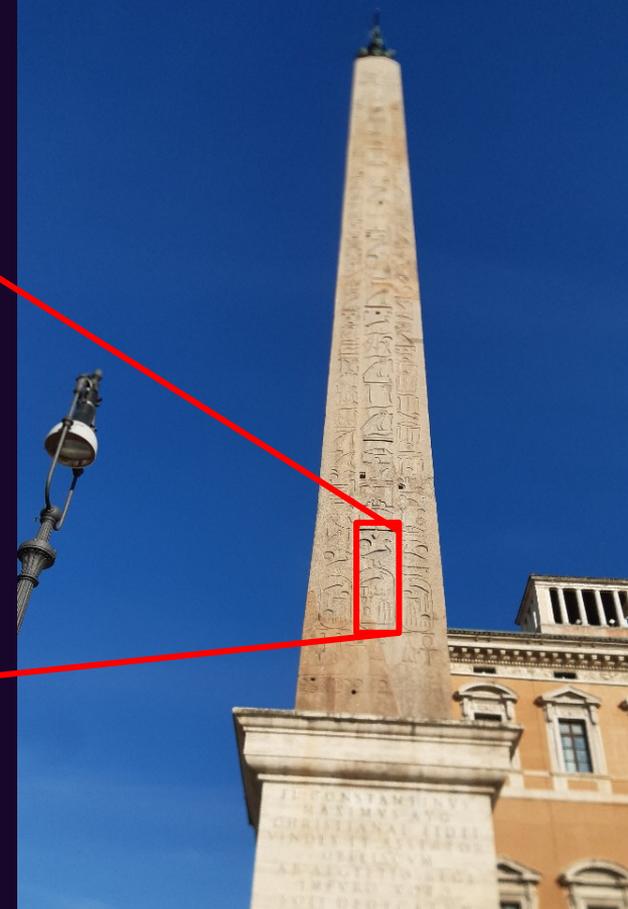


# Simboli che rappresentano numeri

```
00010011111100000010001110011110000
0011111101100000000000000011110000
00011010010000011110011111000100001
11110011110010111110011010000001101
00010011111100010110011110011100000
00010011000000000010011110001100000
00010011001100010010011110011100000
00011010010000011110011111000100001
00110100000000100111000000011001000
00010011000000000010011110001100000
00010011111100000010001110011110000
```

# Simboli vs. codici

$z^c$ -re dhwti-msi(w)



# Simboli vs numeri vs cifre

782330 209 765130,  
45505 218 13290 522,  
831486' 218 724170,  
14614218 12700698!  
61632 209 12651712,  
(47834 16 51920),  
14' 987328 765120,  
61946 209 56016.

12'14' 3344? 193 53706,  
0 193 3354 65246...

BEFFA D1 BACCA,  
B1C1 DA CACC1A,  
CAFFE' DA BOCCA,  
DEFECA C1CC1A!  
FOCO D1 C10CC0,  
(BADA 10 CAD0),  
E' F10C0 BACCO,  
F1FA D1 DAD0.

C'E' D10? C1 D1CA,  
0 C1 D1A FEDE...

# Fenomenologia dell'AI



# I computer non trattano numeri

Per i «numeri» nel computer non valgono né la legge di cancellazione

$$a + c = b + c \Rightarrow a = c$$

né la legge associativa:

$$(a + b) + c = a + (b + c)$$

Un computer, infatti, non tratta numeri razionali ma rappresentazioni posizionali binarie con un numero finito di cifre in formato esponenziale...

# Provare per credere...

```
Python 3.12 (64-bit) x + v - □ ×
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb 6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> 1.000000000000000002 - 1.000000000000000001 == 0.000000000000000001
False
>>>
>>> (1 + 1e-16) + 1e-16 == 1 + (1e-16 + 1e-16)
False
>>>
>>> x = 1 + 2**-29
>>> y = 1 + 2**-30
>>> x**2 - y**2 == (x + y)*(x - y)
False
>>> |
```

E... no, il problema non è Windows...

# I computer non trattano simboli

---

Ogni cosa all'interno di un computer è una sequenza binaria, interpretata come rappresentazione posizionale di un numero binario.

Una immagine, un glifo o una lettera sono rappresentati in forma binaria secondo una opportuna codifica.

# I computer trattano codici

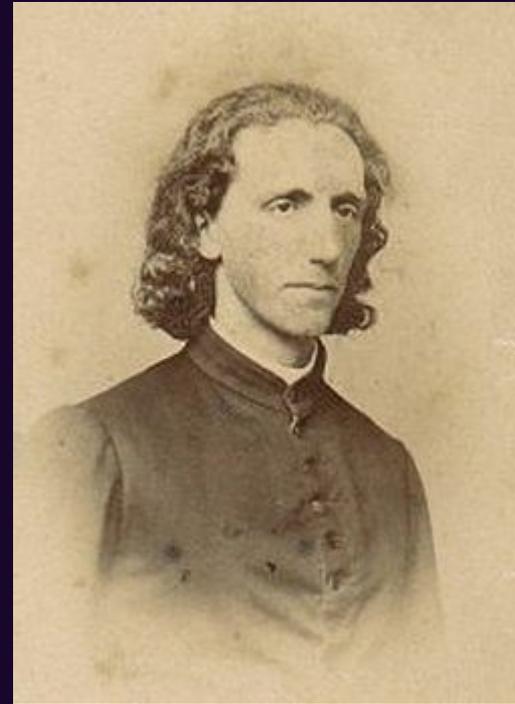
I computer non trattano né numeri né simboli, ma solo codici, precisamente codici binari.

"ABC" -> 01000001.01000010.01000011.00000000  
1000 -> 11101000.00000011.00000000.00000000  
3.14159 -> 11010000.00001111.01001001.01000000

«Subsimbolico» non allude ai numeri, la cui ontologia è fuori dalla macchina, ma ai codici.

# Il mondo come intelligenza e rappresentazione

Ogni fenomeno mentale è caratterizzato da ciò che gli Scolastici medievali chiamavano in-esistenza intenzionale (o mentale) di un oggetto e che noi potremmo chiamare, sebbene in modo parzialmente ambiguo, referentesi a un contenuto, diretto verso un oggetto (che non è qui inteso come una cosa) od oggettività immanente. Ogni fenomeno mentale include qualcosa come oggetto in se stesso, sebbene non tutti lo facciano allo stesso modo. [...] Questa in-esistenza intenzionale è caratteristica esclusivamente dei fenomeni mentali. Nessun fenomeno fisico esibisce nulla di simile. **Potremmo, quindi, definire i fenomeni mentali dicendo che sono quei fenomeni che contengono un oggetto intenzionalmente in se stessi.**



# Rappresentazione senza intelligenza

THE LOGIC THEORY MACHINE  
A COMPLEX INFORMATION PROCESSING SYSTEM

by

Allen Newell and Herbert A. Simon

P-868

June 15, 1956

# Intelligenza senza rappresentazione

## Intelligence without representation\*

Rodney A. Brooks

*MIT Artificial Intelligence Laboratory, 545 Technology Square, Rm. 836, Cambridge, MA 02139, USA*

Received September 1987

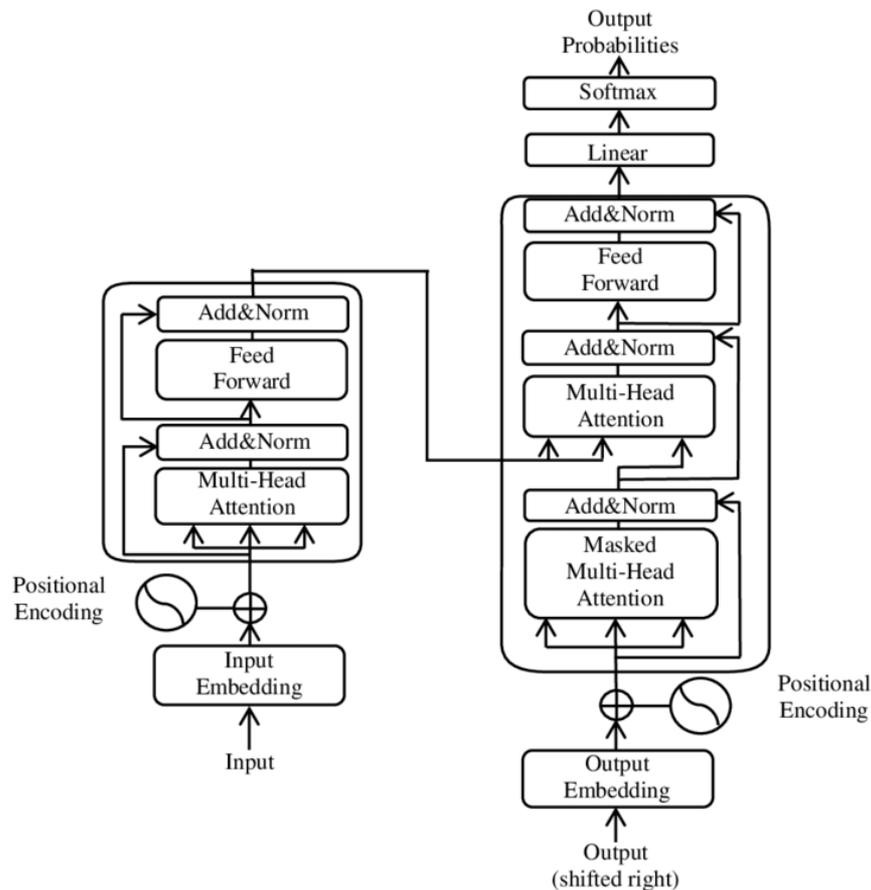
Brooks, R.A., Intelligence without representation, *Artificial Intelligence* 47 (1991), 139–159.

\* This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the research is provided in part by an IBM Faculty 9 Development Award, in part by a grant from the Systems Development Foundation, in part by the University Research Initiative under Office of Naval Research contract N00014-86-K-0685 and in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-85-K-0124.

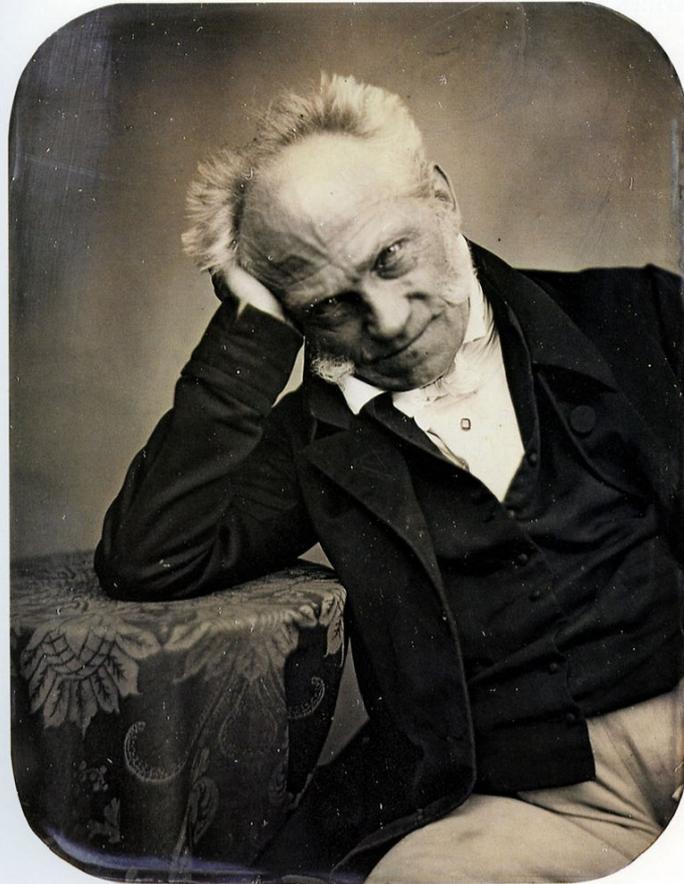
### *Abstract*

Artificial intelligence research has foundered on the issue of representation. When intelligence is approached in an incremental manner, with strict reliance on interfacing to the real world through perception and action, reliance on representation disappears. In this paper we outline our approach to incrementally building complete intelligent Creatures. The fundamental decomposition of the intelligent system is not into independent information processing units which must interface with each other via representations. Instead, the intelligent system is decomposed into independent and parallel activity producers which all interface directly to the world through perception and action, rather than interface to each other particularly much. The notions of central and peripheral systems evaporate everything is both central and peripheral. Based on these principles we have built a very successful series of mobile robots which operate without supervision as Creatures in standard office environments.

# Intelligenza senza chiedersi il perché



Ma che state a di'...



# Separazione fra sintassi e semantica

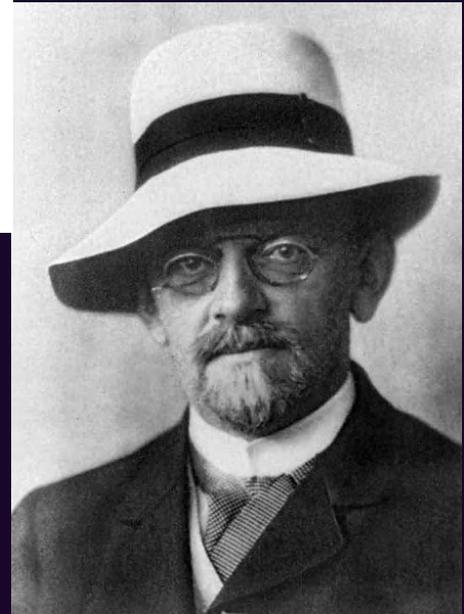
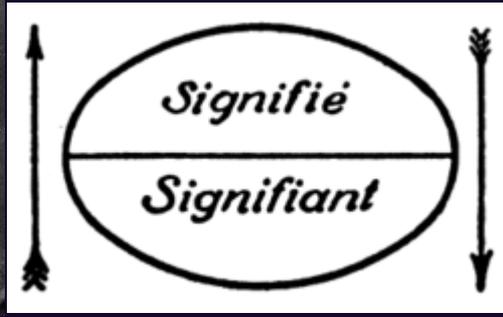
Wir geben nun das zugehörige Axiomensystem an. Als logische Grundformeln haben wir zunächst die Axiome des Aussagenkalküls, die wir der Einfachheit halber in derselben Form wie früher geben.

- a)  $X \vee X \rightarrow X$ .
- b)  $X \rightarrow X \vee Y$ .
- c)  $X \vee Y \rightarrow Y \vee X$ .
- d)  $(X \rightarrow Y) \rightarrow [Z \vee X \rightarrow Z \vee Y]$ .

( $\mathfrak{A} \rightarrow \mathfrak{B}$  ist hier wieder wie früher als eine Abkürzung für  $\mathfrak{A} \vee \mathfrak{B}$  aufzufassen.)

Dazu kommen jetzt als zweite Gruppe zwei Axiome für „alle“, und „es gibt“ hinzu.

- e)  $(x) F(x) \rightarrow F(y)$ .
- f)  $F(y) \rightarrow (E x) F(x)$ .



# Produttori di significanti



# Produttori di significati

ELOGIVM XXVII

## AEGYPTI PRISCA SAPIENTIA.

FERDINANDVS



Elogium hieroglyphicū  
FERD. III. CAESARIS  
temporalitatis  
huius erectionis obelisci  
aeternum consecrauit  
A. K. S. I.

III. CAESAR

tiz instrumentum,  
uersi oculus,  
Austriacus,  
cuncta perlustrat;  
natrice omnia constituat;  
nium benefactor,  
Sapientia conspicuus,

tium promotor,  
inventionum author,  
rialis Legislator,  
curz intentus uoice;  
fortitudine & robore

salciens,  
lectui conformer  
sua dirigens;  
diuus Legislator  
vitali influxu  
ris terrenis;  
rum populus

cutitor, sollicita  
Regnorum,  
commodis providens;  
que Ecclesiasticz  
lantissimus;  
dæmon politici  
seruationi incumbens,  
excisor, profligator,  
pulsor.



ATHANASII KIRCHERI E SOCIETATE IESV  
OEDIPVS ÆGYPTIACVS  
AD FERDINANDVM III CAESAREM SEMPER AVGVSTVM.



# Cibernetica e fantasmi



La macchina letteraria può effettuare tutte le permutazioni possibili in un dato materiale; ma **il risultato poetico** sarà l'effetto particolare d'una di queste permutazioni sull'uomo dotato d'una coscienza e d'un inconscio, cioè sull'uomo empirico e storico, **sarà lo shock che si verifica solo in quanto attorno alla macchina scrivente esistono i fantasmi nascosti dell'individuo e della società.** [...] Dirò che a questo punto è l'atteggiamento della lettura che diventa decisivo; è al lettore che spetta di far sì che la letteratura espliciti la sua forza critica, e ciò può avvenire indipendentemente dalla intenzione dell'autore.

# Li aspettavamo da 70.000 anni

Nelle ultime decine di migliaia di anni, ci siamo specializzati nel proiettare significati su qualsiasi cosa: il moto dei pianeti, le linee della mano, il fegato delle pecore etc.



# Ora sono qui e ci rimarranno

Ora finalmente, con gli LLM, abbiamo degli infaticabili produttori di significanti fatti apposta per noi e non frutto di un improbabile e ateleologico dono del caso.

**Come posso aiutarti?**

Invia un messaggio a Copilot



## SPONSOR



## PARTNER



Diamond  
Gold  
Bronze

# Thank you!

👉 slides & videos: <https://www.improve.tech/videos>

>> AI CONF 2025