



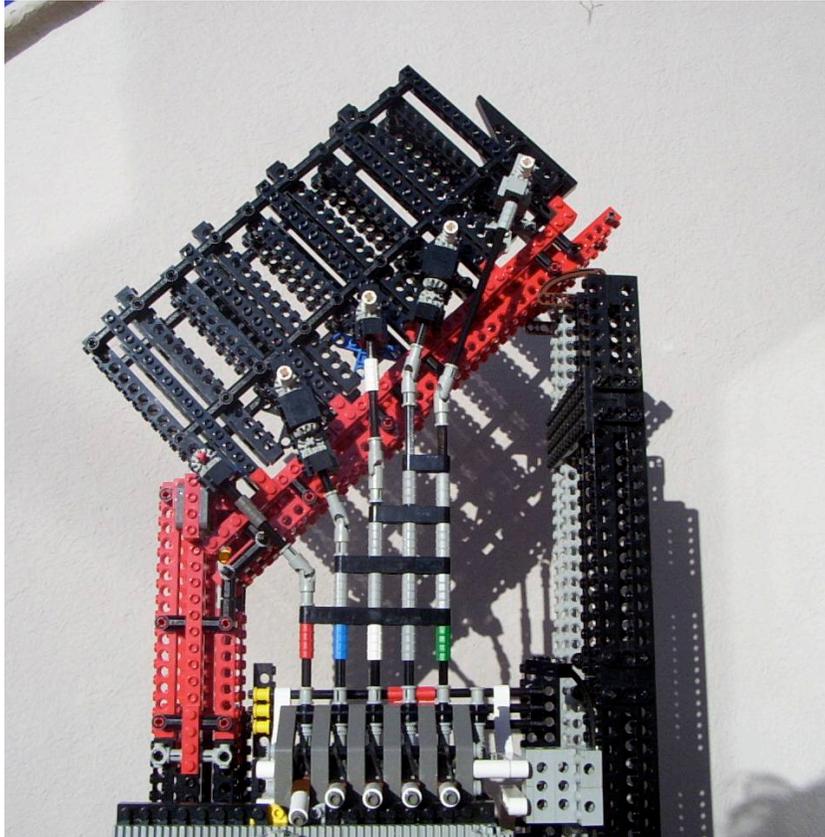
Turing

Daniele Mundici

Dipartimento di Matematica “Ulisse Dini”

Università di Firenze

mundici@math.unifi.it



la prima
macchina

a scuola

Dopo due tentativi
andati male è
ammesso al
King's College di
Cambridge



Il programma di Hilbert



David Hilbert 1862-1943

Wir müssen wissen,
wir werden wissen
Epitaffio

- dopo gli scossoni causati dalle geometrie non euclidee e soprattutto dai paradossi
- Hilbert pensò che il più solido fondamento per la Matematica
- sarebbe dato da una procedura per decidere ogni problema
- **in un numero finito di passi**
- *problema della decisione = Entscheidungsproblem*

Kurt Gödel

Nel 1931 Kurt Gödel dimostra il grande Teorema di Incompletezza:

Ogni sistema di assiomi sufficiente per trattare l'addizione e la moltiplicazione dei numeri naturali contiene un'affermazione indimostrabile

Rimaneva aperto il problema della decisione: *Entscheidungsproblem*:

decidere in un numero finito di passi se un'affermazione A sia o no conseguenza un'affermazione B



Kurt Gödel

ma come definire **numero finito di passi**?

- da tempo l'umanità sa costruire figure geometriche, ma solo nel 1801 Gauss trovò una condizione necessaria e sufficiente su n affinché l' n -gono regolare sia costruibile con riga e compasso.
- Per questo risultato non basta saper costruire poligoni regolari: occorre **definire la non costruibilità**
- analogamente, da millenni è chiaro che funzioni come l'addizione e il massimo comun divisore sono “effettivamente calcolabili”. Euclide descrive un meraviglioso algoritmo per calcolare il massimo comun divisore in pochi facili passi
- ma fu l' Entscheidungsproblem a imporre il problema di **definire il non calcolabile**
- perché chi congetturava che la Matematica non sia calcolabile, non sapeva da dove partire

all'università



nel 1935 segue un corso a Cambridge del topologo M. H. A. Newman e da lui apprende che il problema della decisione di Hilbert é ancora aperto: Hilbert chiede se esista una procedura per decidere ogni congettura matematica
in un numero finito di passi

Turing allora scrive un articolo...

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]



The “computable” numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π , e , etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel†. These results



† Gödel, “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I”, *Monatshefte Math. Phys.*, 38 (1931), 173–198.

I simboli sono stati paragonati agli angeli

- portano informazioni fondamentali (sui numeri, su punti e rette, o anche solo su tavoli e boccali di birra, direbbe Hilbert)
- non avendo massa, attraversano immense distanze con la velocità della luce, silenziosamente, a costi quasi nulli
- ma a differenza degli angeli, **i simboli si sposano con gli algoritmi**: e generano ziloni di bit di informazione
- le agenzie matrimoniali che combinano questi matrimoni sono i **sistemi notazionali** (romano, greco, decimale, binario,...)
- il sistema romano combina nozze coi fichi secchi, come si vede subito provando a moltiplicare XVIII per XLVII
- il sistema decimale si insegna a scuola: 18×47 è decente ora
- altri sistemi funzionano ancora meglio...

il miglior modo di scrivere XLVII

- 47 in notazione binaria è **1 0 1 1 1 1**
- 1 trentaduetto, neanche un sedicetto, 1 otetto, 1 quartetto, 1 coppia, 1 unità
- **ancor meglio** è scrivere 47 in **notazione binaria inversa**:
- **47 = 1 1 1 1 0 1**
- 1 unità, 1 coppia, 1 quartetto, 1 otetto, 0 sedicetti, 1 trentaduetto
- chi decide la bontà, e quindi il successo, di una notazione?
- la felicità dello sposalizio tra simboli e algoritmi in tale notazione
- **ora molti di voi saranno giustamente perplessi**: cosa si può ottenere di buono, dal punto di vista matematico, con questa attenzione allo sposalizio tra i simboli e gli algoritmi, dettata essenzialmente da ragioni utilitaristiche di efficienza?

chi è perplesso è in buona compagnia

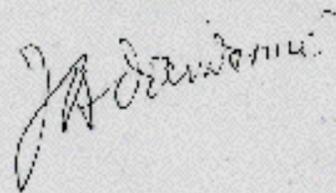
Paris , April 15 , 1989

Dear Professor Mundici ,

I have received your letter and your paper on logic and mathematics . I think it was well adapted to your audience , and that you presented my point of view very fairly. Your examples are certainly understandable by high school teachers , but I think you might add a word of warning that they represent an infinitesimal portion only of present day mathematics. Questions linked to computing , such as the complexity of computations or of proofs , may have practical interest , but are completely irrelevant for a specialist in number theory who wants to solve outstanding problems .And up to now the new results

to problems in C^* algebras or to problems in logic .

Sincerely yours



J. Dieudonné

i problemi computazionali, come la complessità dei calcoli o delle dimostrazioni, possono avere interesse pratico, ma sono totalmente irrilevanti per uno specialista di teoria dei numeri che voglia risolvere problemi di grande portata

per capire come Turing definì il “passo di calcolo” occorre allontanarsi dalle buone compagnie bourbakiste...

la funzione “successore” è indubbiamente l'insieme delle coppie $(i, i+1)$ secondo Dirichlet

ma dovendola insegnare agli scolari, o al computer, è meglio definirla in un altro modo

dando le istruzioni necessarie per calcolare il successore di un qualsiasi numero

la funzione “successore” è
questo misterioso trio di istruzioni
il nostro primo esempio di Macchina di Turing

se in stato 0 leggi 1, scrivi 0, vai a destra, e stai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h

vediamo che cosa ci dicono di fare queste istruzioni quando
diamo in input al computer il numero 47 in **notazione binaria
inversa** $47 = 111101$

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h

0

1	1	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

1	1	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	1	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	1	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	1	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	1	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	0	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	0	0	1				
---	---	---	---	---	---	--	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

in assenza di istruzioni, la macchina si ferma

h



avendo calcolato 48
in notazione binaria inversa

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h

Teorema *Questo terzetto di istruzioni, partendo da qualsiasi numero n scritto in notazione binaria, calcola il successore di n in un numero finito di passi*

questo terzetto è il nostro primo esempio di
macchina di Turing

da 73 anni, i "passi di calcolo" sono i passi
delle macchine di Turing

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h

DIMOSTRAZIONE

andrebbe fatta per induzione,
sai che pacco, oltre tutto poco istruttivo

noi la faremo per ogni $n = 127$

ossia vediamo se questo terzetto
su input 111111 produce in output 00000001

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	1	1	1	1	1			
---	---	---	---	---	---	---	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	1	1	1	1			
---	---	---	---	---	---	---	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	0	1	1	1			
---	---	---	---	---	---	---	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

0	0	0	0	0	1	1			
---	---	---	---	---	---	---	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0

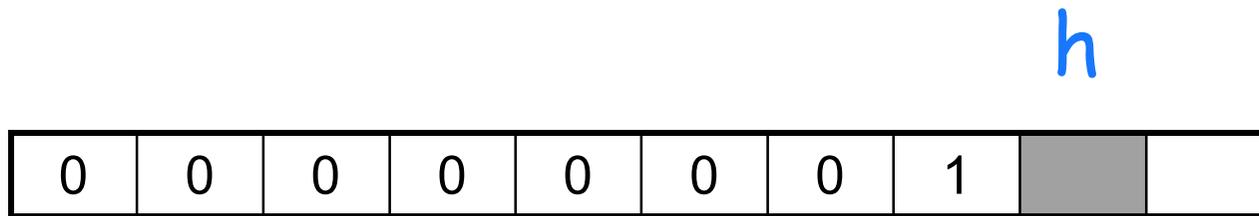
0	0	0	0	0	0	1			
---	---	---	---	---	---	---	--	--	--

se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi __, scrivi 1, vai a destra e vai in stato h

0



se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h



in assenza di istruzioni la macchina si
ferma : ha calcolato 128

QED

le tre istruzioni

- se in stato 0 leggi 1, scrivi 0, vai a destra, e vai in stato 0
se in stato 0 leggi 0, scrivi 1, vai a destra e vai in stato h
se in stato 0 leggi _, scrivi 1, vai a destra e vai in stato h
- sono (tutto quanto è necessario sapere per calcolare) la funzione successore, in notazione binaria inversa. In notazione romana occorrono molte più istruzioni. Anche in decimale ce ne vogliono almeno una dozzina...
- la MACCHINA che calcola il successore **in un numero finito di passi** consiste nelle tre linee di questa matrice:

0	1	0	>	0
0	0	1	>	h
0	_	1	>	h

la meravigliosa macchina di Turing U

- Ogni macchina di Turing è una lista di simboli
- noi ne abbiamo vista solo una

010>0

001>h

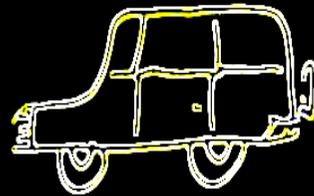
0_1>h

possiamo scriverla e darla in input a un'altra macchina

	0	1	0	>	0		0	0	1	>	h		0	_	1	>	h				
--	---	---	---	---	---	--	---	---	---	---	---	--	---	---	---	---	---	--	--	--	--

Turing scrisse le istruzioni di una macchina U che "simula"
non solo questa, ma ogni macchina di Turing...

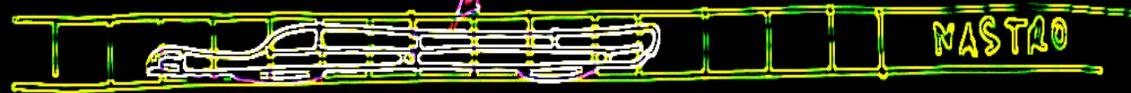
LA METAMORFOSI DELL' HARDWARE (= FERRAGLIA)



(MACCHINA)

IN SOFTWARE (= PURE SIMBOLO)
O DATO-INPUT = NUMERO

(MACCHINA FORMALIZZATA)



la macchina U lavora su questo tipo di input

nel 1936 U è solo un oggetto matematico, ma nel 1946 si materializzerà e nascerà una nuova scienza.

la metamorfosi fondamentale

- questa metamorfosi della macchina che si fa simbolo
- segnala che l'hardware si può trasformare in software
- ogni macchina, divenuta simbolo, può essere messa come input per la macchina universale U
- U ha un certo numero (finito!) di quintuple capaci di simulare le quintuple della macchina M , per qualsiasi M
- U calcola tutte le funzioni effettivamente calcolabili
- nessuno ha finora scoperto una funzione effettivamente calcolabile che non sia calcolabile da U

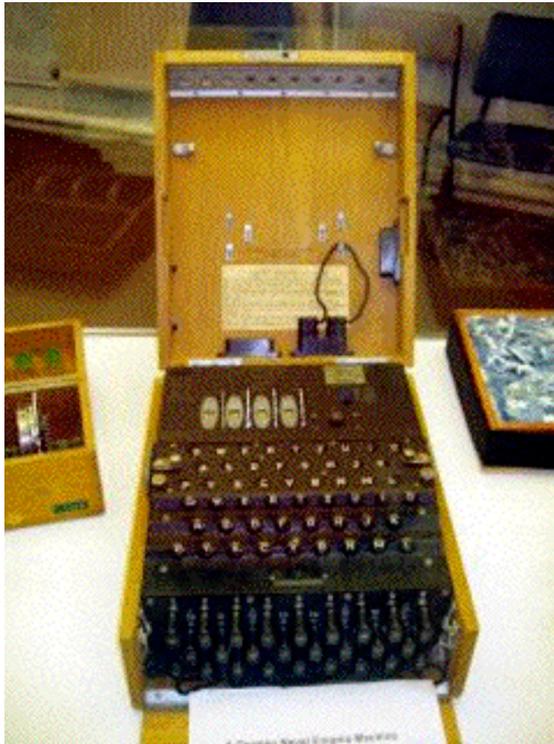
cosa contiene questo articolo del 1936

1. viene introdotta la *macchina di Turing*.
2. da allora in poi i passi di calcolo sono i passi delle macchine di Turing, e il *costo* di ogni algoritmo f è misurato dal *numero di passi* necessari per avere $f(x)$
3. viene introdotta la *macchina universale* U
4. il problema se U vada in loop su un arbitrario input x è dimostrato essere *incalcolabile* in un numero finito di passi
5. trascrivendo questo problema in formule viene data *risposta negativa al problema della decisione di Hilbert*



la seconda macchina

Enigma

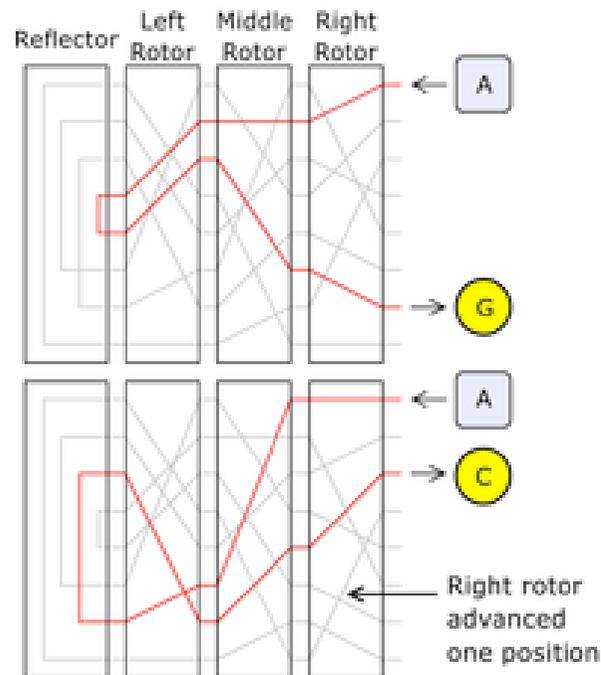


- inventata da Arthur Scherbius nel 1918
- commercializzata nel 1923
- e da allora usata per comunicazioni interbancarie protette
- usata poi dalla Wermacht
- prodotta in 50000 (cinquantamila) esemplari

la macchina Enigma

- era costituita da una serie di rotori che venivano fatti girare dalle lettere battute a macchina

Enigma è una macchina simmetrica: la stessa macchina serve per cifrare e decifrare



il Dipartimento Crittografico polacco

- 1932: il crittografo polacco Marian Rejewski analizza la macchina Enigma: applicando la **teoria dei gruppi** risolve un problema strutturale decisivo per decifrarle
- 25 luglio 1939 – la Polonia è minacciata da Hitler. Rejewski invita in Polonia alcuni crittografi francesi e inglesi, e rivela loro le scoperte dei crittografi polacchi
- passando inoltre all'Inghilterra una copia della macchina Enigma

una rincorsa contro il tempo

- Enigma viene potenziato nei giorni successivi alla visita di Chamberlain in Germania
- i crittografi polacchi non riescono più ad analizzare i cifrati



settembre 1939

- 1 Settembre 1939: la Germania invade la Polonia
- 3 Settembre 1939: l'Inghilterra dichiara guerra alla Germania
- inizia la seconda guerra mondiale
- molti crittografi polacchi furono catturati, ma nessuno rivelò i risultati segreti ottenuti dal gruppo di Rejewski, e trasmessi a francesi e inglesi

la seconda guerra mondiale

- 1940: la Wehrmacht invade la Francia
- la Gran Bretagna deve fronteggiare il più poderoso esercito di tutti i tempi
- Durante tutta la guerra, la sua sopravvivenza dipese dai rifornimenti che riceveva dalle colonie e dagli Stati Uniti attraverso l'Atlantico
- L'ammiraglio Dönitz era a capo della flotta tedesca dei sottomarini U-boat che colpivano le navi mercantili che trasportavano questi rifornimenti



Karl Dönitz

la battaglia dell'Atlantico

- la tattica degli U-boot era gestita da un sistema di trasmissioni via radio
- ogni U-boot doveva periodicamente affiorare per ricevere ordini da Berlino
- i messaggi tra gli U-boot e il centro strategico sotto il comando di Dönitz erano ovviamente in codice



- L'esercito tedesco trasmetteva ogni giorno migliaia di messaggi codificati con sistema Enigma.
- la minaccia degli U-boot era andata crescendo: da un solo sottomarino nel 1935, a 57 allo scoppio della guerra.
- nei primi nove mesi del conflitto gli U-boot affondarono 701 navi alleate, con 2.3 milioni di tonnellate di carico, per lo più proveniente dall'America.
- l'Inghilterra non avrebbe potuto sopravvivere a lungo con rotte così insicure.



**“The only thing that really
frightened me during the
war was the U-Boot peril.”**

Winston Churchill

Bletchley Park

- Quando l'Inghilterra dichiarò guerra alla Germania, Turing passò a tempo pieno nel dipartimento crittoanalitico di **Bletchley Park**.
- lavorando per l'esercito britannico, che cercava di scoprire i codici usati dai tedeschi, che giravano sulla macchina Enigma
- 10000 persone lavoravano a Bletchley Park per decrittare Enigma (100000 per il progetto Manhattan)





la terza
macchina

Bomba, la macchina anti-Enigma

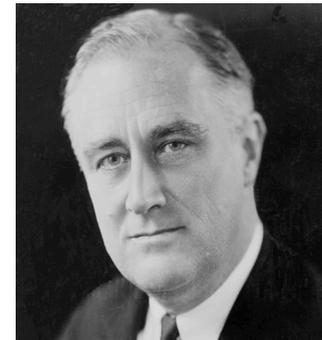
- Quello stesso Turing che aveva ideato la macchina universale U
- ora crea un'altra macchina, la Bomba.
- La Bomba di Turing era un potenziamento della bomba polacca.
- il nome proviene dalla “Bomba” di Rejewski (in polacco “bomba” è come in italiano)
- un altro matematico di Cambridge, W. G. Welchman, dette un contributo importante alla Bomba
- Primo esemplare “Victory”: Bletchley Park, Marzo 1940

Bomba, la macchina anti-Enigma

- prima di Turing nessuno aveva avuto molto successo con Enigma a Bletchley Park
- le macchine elettro-meccaniche sviluppate a Bletchley Park nel 1940 decodificavano appena 50 messaggi Enigma alla settimana
- era opinione generale che Enigma fosse inattaccabile
- Turing attaccò Enigma, utilizzando anche sofisticati processi statistici e materiale informativo intercettato
- a metà del 1941 la decrittazione di Enigma procedeva già sistematicamente
- nel 1943 si decodificavano 3000 messaggi al giorno

1941-1942

- alla fine del 1941, quando gli USA entrarono in guerra, la battaglia dell'Atlantico cominciava a pendere a favore degli Alleati
- Turing attraversò l'Atlantico nel novembre 1942 per curare la cifratura della comunicazione tra Roosevelt e Churchill.
- Hut 8 era la sezione di Bletchley Park incaricata di decrittare i messaggi navali di Enigma.
- Turing ne era il direttore.



la sconfitta di Enigma

- dipese da una combinazione di genialità matematica, spionaggio, errori degli operatori della Wehrmacht, e tanto lavoro di migliaia di persone
- ebbe un effetto notevole sull'esito della guerra:
- l'Inghilterra era in grado di conoscere in anticipo le dislocazioni degli U-boat
- e i piani di bombardamento dell'aviazione tedesca
- **Ma: come fanno tutti i crittoanalisti, tenere segreta una scoperta è più importante che usarla a breve termine, o celebrare gli scopritori: pertanto al lavoro di Turing venne tolto il segreto nel 1996!**

verso la fine della guerra

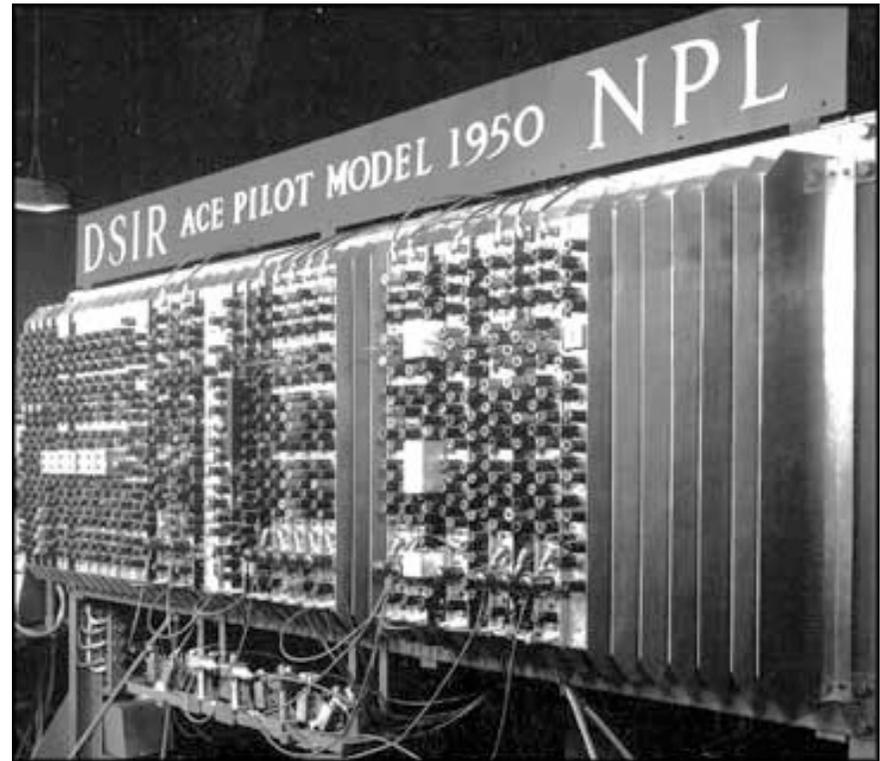
- Turing ottenne l'alta onorificenza OBE-Order of the British Empire
- nessuno si insospettì : dopo tutto, Turing era famoso per la macchina universale
- dedicava molto tempo all'elettronica, creando un suo personale elegante sistema di cifratura della comunicazione vocale
- e intanto coltivava un progetto ben più ambizioso...



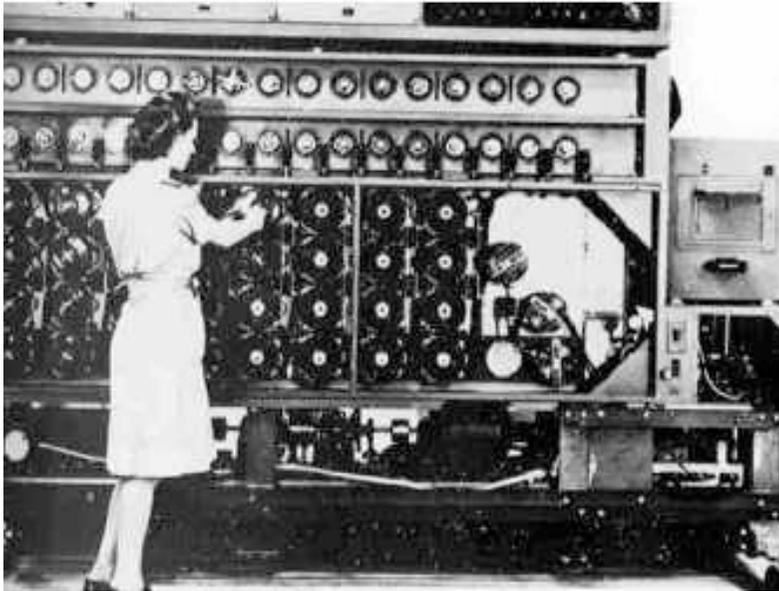
la quarta
macchina

la corsa a costruire il computer

dopo la guerra Turing fu invitato al British National Physical Lab dove progettò un **Automatic Computing Engine**, basato sulla sua idea di macchina universale U. Il progetto fu bloccato, e Turing passò all'università di Manchester



prototipo di ACE



**le enormi potenzialità
del computer erano
ormai ben chiare**

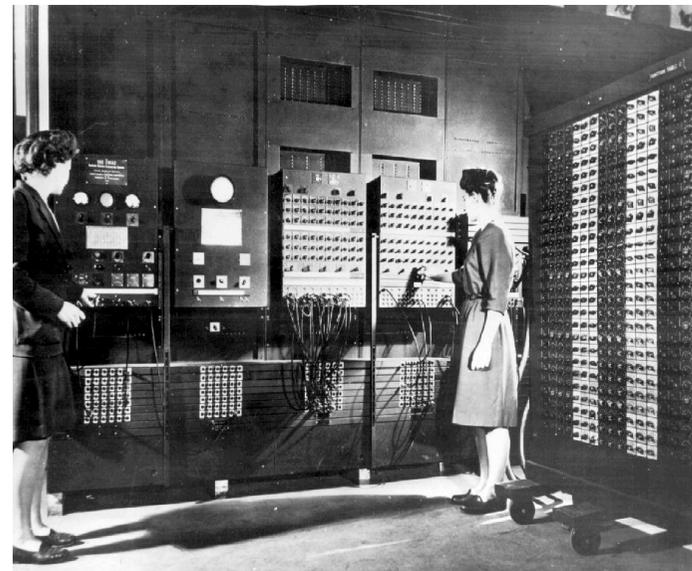
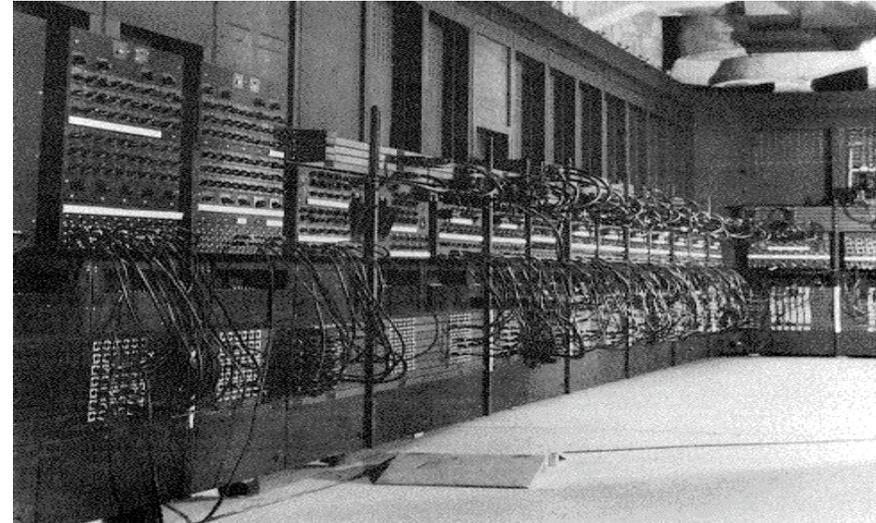
**meteo, fisica
atomica, strategia,
crittografia, balistica,
simulazione...**

**i vincitori della guerra puntavano a realizzare il
computer come nuova arma, di carattere
diverso dall'altra, ma non meno poderoso**

e qualcuno, in gran segreto, correva più veloce di Turing

ENIAC – 1946

- **E**lectronic **N**umerical **I**ntegrator **A**nd **C**omputer
- costruito nell' Università di Pennsylvania
- per applicazioni militari
- 5,000 operazioni/sec
19,000 tubi 30 tonnellate
- 150 kilowatts: Quando lo accendevano, le luci di Filadelfia si abbassavano un po', per il calo di tensione
- i banchi erano insetti veri che danneggiavano i tubi



Transistor - 1947

- Inventato presso i Bell Labs, USA
- Rispetto al tubo a raggi catodici, offriva:
 - dimensioni più piccole
 - maggiore affidabilità
 - minor consumo di energia
 - minore costo
- **tutti i computer moderni sono fatti di transistor opportunamente miniaturizzati**

Floppy Disk - 1950



- inventato presso l'Università Imperiale di Tokyo
- offriva più rapido accesso ai dati, rispetto ai nastri magnetici che si usavano prima
- infatti il vecchio nastro magnetico (come la musicassetta o la pellicola cinematografica) permetteva solo l'accesso sequenziale ai dati
- mentre il floppy disk permette un accesso diretto, proprio come il CD, o il pen driver

1951: non solo militari e politici

- fino all 1951 tutti i computer erano stati progettati per usi militari o governativi
- Nel 1951, appare il primo computer commerciale, prodotto su larga scala: UNIVAC 1
- **UNIV**ersal **A**utomatic **C**omputer (prodotto dalla Eckert & Mauchly Computer Company)
- UNIVAC 1 è il primo computer che manipola non solo numeri ma anche parole di testo
- Max velocità: 1905 operazioni/sec
- Costo: un milione di dollari
- 5000 tubi 8 tonnellate 100 kilowatts
- Negli anni 1951-57 ne furono venduti 48 esemplari

1952: il compilatore

- finora abbiamo parlato di hardware, ossia componenti fisiche
- ma l'invenzione chiave per l'evoluzione del computer è stata nella sua **programmazione**, con l'invenzione del **compilatore**
- prima dell'invenzione del compilatore, scrivere un programma era un'operazione noiosa; correggerne i banchi era peggio
- perché tale programma era scritto nel linguaggio-macchina, molto simile alle quintuple di una macchina di Turing
- il **compilatore** è un medium: trasforma in linguaggio-macchina, quello che vogliamo comunicare al computer—se gli parliamo con un livello di chiarezza paragonabile a quello matematico
- è come se il computer avesse fatto un corso di matematica, per evitare a noi di fare un arido corso di linguaggio-macchina

e Turing ?
vi ricordate dove
l'abbiamo lasciato?

All'università di Manchester

The Manchester Universal Electronic Computer



nominato professore di matematica pura nel 1945 all'università di Manchester, negoziò con la Royal Society per avere finanziamenti adeguati per un progetto di costruzione di un computer

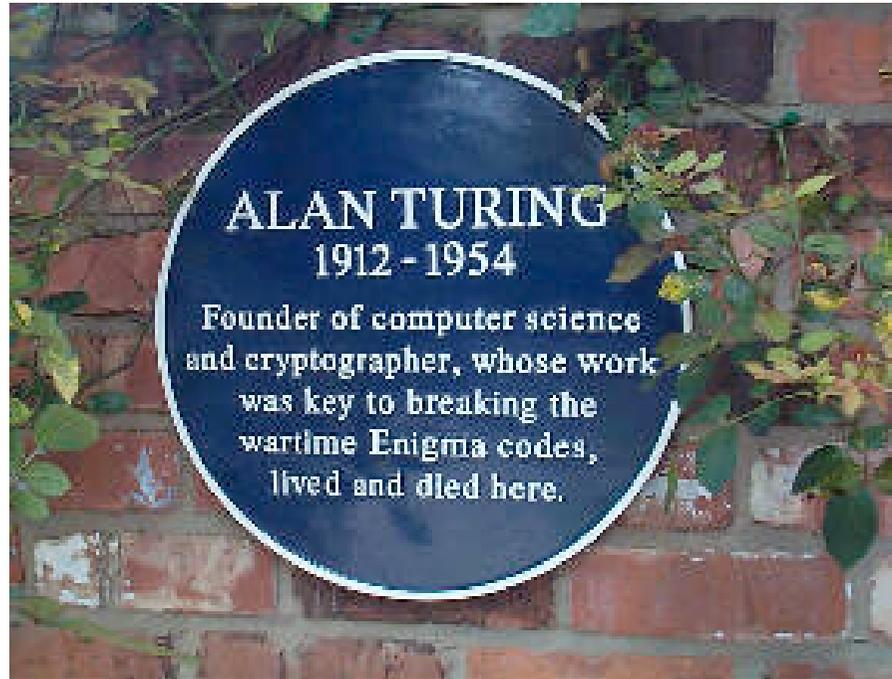
nel giugno 1948 dette una prima dimostrazione della sua realizzabilità

1952, la fine

- Alan Turing fu arrestato e processato il 31 Marzo 1952, per omosessualità
- per evitare la prigione accettò per un periodo di un anno di farsi iniettare estrogeni
- l' 8 giugno 1954 la donna delle pulizie trovò il suo cadavere
- Turing era morto il giorno prima: di fianco al suo letto c'era una mela morsicata, avvelenata al cianuro
- sua madre avvalorò la tesi che il figlio fosse morto accidentalmente, per un esperimento di chimica finito tragicamente
- il verdetto del coroner fu suicidio



- RIFERIMENTI BIBLIOGRAFICI
- D.M. (a cura di) **La Scienza dei Calcolatori**, Le Scienze Quaderni (Edizione italiana di *Scientific American*) vol. 56, ottobre 1990. (Contributi di Bennett, Landauer, Feynman, Wolfram, Rota, e altri).
- D.M., W. Sieg, **Paper Machines**, *Philosophia Mathematica*, 3 (1995) pp 5-30.
- D. M., W. Sieg, **Computability**, voce nella *Routledge Encyclopedia of Philosophy*.



grazie

per saperne di più sul computer
Mundici-Sieg, "*Paper Machines*"
Philosophia Mathematica, 3 (1995) pp. 5-30