Alan Turing: tra logica e informatica

1. Introduzione

Alan Mathison Turing nacque a Londra il 23 giugno 1912; nel 1931 fu accettato al Kingís College di Cambridge; nel 1936 pubblicò il primo articolo scientifico su un problema di logica matematica: ìOn computable numbers, with an application to the *Entscheidungsproblem*î[1]. [2]. La soluzione che propose lo avrebbe reso famoso nel campo della matematica e dellíinformatica.

Durante la Seconda Guerra Mondiale, Turing aiutò il suo paese e gli eserciti alleati producendo un metodo per decifrare i messaggi in codice degli avversari. Il 29 marzo 1951 fu eletto membro della Royal Society, il massimo riconoscimento per uno scienziato nel Regno Unito.

Turing era omosessuale e, in Inghilterra, líomosessualità era considerata una malattia mentale. Fu processato per questo nel 1952, riconosciuto colpevole e costretto a iniezioni di estrogeni per un anno. Lí8 giugno 1954 fu trovato morto dalla domestica: disteso sul letto, si era avvelenato mangiando una mela intrisa di cianuro di potassio. È difficile non pensare che il suicidio sia stato una conseguenza della punizione inflittagli líanno prima.

L'articolo del 1936 sui numeri calcolabili si sarebbe dimostrato di importanza fondamentale per gli sviluppi del calcolo meccanico e, in seguito, dell'informatica[2]. Ma la rilevanza storica di Alan Turing è legata in modo cruciale ad altri eventi di enorme importanza del secolo passato: nel 1938 fu reclutato dalla Government Code and Cyber School, il GC&CS, ramo del Military Intelligence - il servizio segreto inglese. Il suo contributo fu quello di sviluppare nuovi metodi per decifrare i messaggi del nemico: grazie a quei metodi, dal giugno 1941, líAlto Comando inglese era a conoscenza di tutte le comunicazioni tra le forze naziste e Winston Churchill riceveva ogni giorno una sintesi

---

1 Sui numeri calcolabili meccanicamente con un'applicazione al problema della decisioneî. L'Entscheidungsproblem (= il `problema della decisione´ per affermazioni matematiche) venne proposto, nel 1928, da David Hilbert: determinare un metodo meccanico che permetta di stabilire, per ogni possibile affermazione matematica, se questa è vera oppure no, appunto di deciderne la verità o meno.

2 Basti pensare che l'equivalente del Premio Nobel per l'informatica è il "Premio Turing".

dei messaggi scambiati dalle forze nemiche.

Alan Turing fu sicuramente uno dei più grandi eroi britannici, anche se il suo contributo per combattere il nemico del suo paese è difficile da valutare con gli schemi usuali: non si distinse per particolari azioni avventurose, ma fu l'uomo che rese possibile conoscere le intenzioni del nemico in tempo reale. Le navi mercantili venivano informate delle possibili rotte degli *U-boot*; le forze terrestri conoscevano i movimenti delle forze nemiche; nei giorni precedenti il D-Day, gli alleati furono in grado di verificare l'efficacia delle notizie erronee che stavano lasciando circolare per distogliere l'attenzione dei nazisti dalle spiaggie della Normadia.

Nel seguito, analizzeremo alcuni dettagli dei successi scientifici di Turing: la nozione di calcolo meccanico, l'analisi per la decrittazione del codice ENIGMA, la questione basilare dell'intelligenza artificiale: che cosa significa che una macchina possa \`pensare´?[3]

2. I numeri calcolabili meccanicamente

Nell'introduzione dell'articolo "Sui numeri calcolabili meccanicamente con un'applicazione al problema della decisione", Turing scrive:

I numeri "calcolabili meccanicamente" possono essere descritti in breve come quei numeri reali la cui espressione decimale si può determinare con strumenti finiti.

Che cosa intende dire con la frase "strumenti finiti"? Turing prosegue dichiarando:

Sebbene l'argomento di questo lavoro siano esplicitamente i numeri calcolabili, è quasi altrettanto facile definire e studiare le funzioni calcolabili di variabile intera o di variabile reale calcolabile, i predicati calcolabili, e così via. I problemi fondamentali sono, comunque, gli stessi in ogni caso, e ho scelto i numeri calcolabili per una trattazione esplicita perché richiedono la tecnica meno complicata.

3 Si veda [1] per ulteriori informazioni sulla vita e sull'opera di Turing.

Spero di trattare nel prossimo futuro le relazioni che intercorrono tra numeri calcolabili, funzioni e via di seguito. Questo includerà lo sviluppo della teoria delle funzioni di una variabile reale espresse in termini dei numeri calcolabili.

Purtroppo questa speranza non era destinata ad avverarsi. Poi, finalmente, afferma:

Nella mia definizione, un numero è calcolabile se la sua parte decimale può essere scritta da una macchina.

A questo punto il problema di comprendere l'intuizione che Alan Turing sta presentando si sposta sul significato di ìscritto da una macchina". E l'autore continua mantenendo un parallelo con un procedimento umano eseguito pedissequamente:

Per il momento, dirò soltanto che la giustificazione sta nel fatto che la memoria umana è necessariamente finita. Possiamo paragonare un uomo nell'atto di scrivere l'espansione decimale di un numero reale a una macchina che è in grado di considerare soltanto un numero finito di condizioni–diciamo che siano q_1, q_2,…, q_R *– che chiameremo "*stati*". La macchina è provvista di un "*nastro*", (l'analogo della carta per l'uomo) che scorre attraverso di essa ed è diviso in sezioni (chiamate "*caselle*") ciascuna in grado di riportare un "*simbolo*".*
*In ogni istante c'è esattamente una casella che è "all'interno della macchina". Possiamo chiamare questa la "*casella in lettura*". Il simbolo nella casella in lettura è detto il "*simbolo in lettura*".*
Il simbolo in lettura è l'unico di cui la macchina sia, per così dire, "a diretta conoscenza".

Per chiarire líintuizione proposta da Turing usiamo un semplice schizzo del nastro, dello stato e del simbolo in lettura. Il nastro è composto di caselle su cui si possono scrivere i simboli zero **0**, uno **1** e la virgola **,** oppure lasciar bianchi. La casella in lettura è marcato in rosso; sopra di esso viene riportata lo stato nel quale la lettura viene eseguita.

Nella figura 1 è rappresentato il nastro di una macchina che

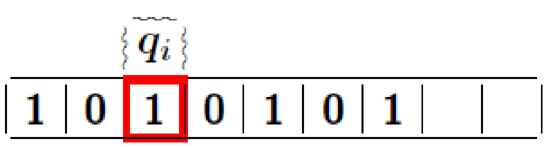

Figura 1.- Esempio di configurazione in una macchina di Turing.

- si trova nello stato q_i
- legge il simbolo **1** nella terza casella del nastro. La coppia costituita da uno stato e dal simbolo in lettura viene detta *configurazione*. Il comportamento istantaneo della macchina è determinato dalla configurazione in cui si trova: in base a questo la macchina può
- scrivere nella casella in lettura un altro segno, lasciare il segno trovato, oppure cancellarlo e lasciare la casella bianca,
- spostare la lettura sulla casella immediatamente a destra oppure sulla casella immediatamente a sinistra,
- cambiare stato.

Quello che la macchina deve fare viene stabilito da una tabella che fa corrispondere alle possibili configurazioni la terna di azioni da intraprendere.

Una volta prescritta la configurazione iniziale e approntato un nastro con eventualmente simboli scritti in alcune caselle, un'*esecuzione* consiste nel procedere attraverso la tabella delle istruzioni cercando la configurazione in cui ci si trova e obbedendo alle istruzioni, perciò scrivere nella casella in lettura, spostare la casella in lettura e passare a un nuovo stato.

Ad esempio, la tabella di istruzioni per scrivere il numero $\frac{1}{3}$ in notazione binaria – cioè $0,\overline{01}$ – su un nastro bianco con l'esecutore inizialmente nello stato q_0, può essere la seguente

configurazione		**terna di azioni**		
stato	lettura	scrittura	spostamento	nuovo stato
q_0		**0**	a destra	q_1
q_1		**,**	a destra	q_2
q_2		**0**	a destra	q_3
q_3		**1**	a destra	q_2

Figura 2 - Tabella di istruzioni per scrivere 1/3 in notazione binaria.

Si noti che le prime due istruzioni fanno scrivere **0,** e che le altre due istruzioni si ripetono alternate indefinitamente a produrre la scrittura

0,010101010…

Si noti anche che non vengono considerate configurazioni con caselle non bianche in lettura poiché si prevede che queste non saranno mai necessarie.

L'esempio riportato sopra è esattamente il primo esempio di numero reale calcolabile, riportato da Alan Turing nell'articolo; il secondo esempio è il numero reale (scritto sempre in notazione binaria)

$$0{,}010110111011110\ldots0\underbrace{1\ldots1}_{n}0\underbrace{1\ldots1}_{n+1}0\ldots$$

che non è periodico. La prima reazione, pensando alle istruzioni da impartire per far scrivere il numero, è che servano un numero infinito di stati, almeno uno per ogni gruppo di simboli **1**. Ma, prima di rinunciare a scrivere una lista finita di istruzioni, è opportuno notare che i simboli **1** da scrivere dopo un simbolo **0** sono *uno di più* rispotto a quelli nel gruppo che precede quel simbolo **0**; basterà *copiare* quel gruppo e scrivere un simbolo **1** in aggiunta.

È molto più utile *vedere* l'esecuzione della macchina che scrive il numero sopra piuttosto che leggere la lista di istruzioni: al sito http://www.disi.unige.it/person/RosoliniG/pG/ si trova un simulatore per l'esecuzione di macchine (che, dopo che Turing le ebbe introdotte nell'articolo in questione, vennero battezzate "macchine *di Turing*") e la lista di istruzioni da eseguire per i due numeri menzionati sopra.

Ad ogni passo di esecuzione, la tabella di istruzioni viene considerata dalla prima istruzione; dunque se ci sono in elenco due o più istruzioni che prevedono di eseguire azioni diverse a partire dalla stessa configurazione, l'esecuzione considererà sempre e soltanto la prima di quelle.

A chi oggi ha esperienza di programmazione appare chiaro che il nocciolo di base delle istruzioni di una macchina di Turing è la gestione di casi (in numero finito), ma anche chi non ha una tale esperienza riesce certamente ad apprezzare che i comandi sono veramente elementari: non è necessaria alcuna "intelligenza". Sono richieste solo attenzione, precisione e pazienza.

Il passaggio successivo nell'articolo di Turing è quello di notare che le azioni stesse di una qualunque macchina sono "calcolabili": per eseguire la lista di istruzioni di una macchina è necessario

- poter *riconoscere* il simbolo nella casella in lettura, cioè confrontarlo con i simboli **0**, **1** e **,**
- accoppiarlo con lo stato della configurazione
- trovare l'istruzione che coinvolge la configurazione
- modificare il simbolo

- modificare lo stato

Turing ha un'intuizione geniale: il nastro è utilizzato per riportare numeri, scritti cifra per cifra, ma il nastro potrebbe benissimo venir utilizzato per trascrivere istruzioni, scritte carattere per carattere. Si potrebbe pensare così di avere una macchina che, se sul nastro è trascritta la lista di istruzioni di una macchina *M*, produce sul nastro il numero che *M* stessa avrebbe scritto.

Turing spiega molto chiaramente che non è difficile immaginare quali istruzioni scrivere per una tale macchina e che questa sarebbe predisposta per eseguire una qualunque macchina di Turing. Proprio per questa caratteristica, Turing chiama tale macchina "universale".

Bisogna porre molta attenzione al fatto che una tale macchina universale opera in modo leggermente diverso dalla macchine introdotte inizialmente: quelle prevedono di iniziare la propria esecuzione su un nastro bianco, questa prevede di avere un *input* scritto su nastro (la macchina da eseguire, appunto). Mentre può sembrare ovvio che tale *input* venga messo a disposizione sul nastro, non è per niente ovvio come una macchina universale possa rientrare nella definizione data in cima: infatti non ci rientra e Turing spiega come la prima definizione sia stata usata perché era la più intutiva, ma la nozione che è necessario usare è quella di macchina che esegue a partire da *input* scritti su nastro.

I concetti presentati nell'articolo sui numeri calcolabili sono quelli che oggi noi troviamo totalmente comuni, nella vita di tutti i giorni, quando ci apprestiamo ad usare un computer: lo stesso strumento (il computer) ci permette di scrivere testi, fare calcoli, disegnare, e ben altro (leggere la posta, navigare in internet, ascoltare musica, guardare video, ecc.). È una versione tecnologicamente molto avanzata del nastro di Turing e della sua macchina universale che può eseguire ogni altra macchina. Il nucleo di ogni computer è una macchina universale come quella ideata per prima da Alan Turing.

3. La matematica astratta in guerra

Nel 1939, immediatamente dopo lo scoppio della Seconda Guerra Mondiale, quando era Fellow di King's College a Cambridge, Alan Turing prese servizio presso il GC&CS, i servizi segreti inglesi di stanza a Bletchley Park. Turing collaborava già con i servizi segreti per cercare sistemi per decifrare i messaggi tedeschi codificati con il sistema ENIGMA.

Il sistema ENIGMA, inventato nel 1918 dall'ingegnere tedesco A. Schrebius e modificato nei decenni successivi, era uno strumento ragionevolmente semplice da usare per scrivere e leggere messaggi in codice. Una macchina ENIGMA era contenuta in una scatola in legno delle dimensioni di una grossa scatola da scarpe e pesava circa 12 kg. L'utilizzatore aveva a disposizione una tastiera alfabetica su cui comporre il messaggio; il messaggio in codice usciva, lateralmente, su una striscia di carta; poteva, a questo punto, essere trasmesso senza timore che intercettatori nemici potessero comprenderne il contenuto.

Figura 3. - Un esemplare di macchina codificatrice/decodificatrice ENIGMA.

L'enorme comodità consisteva nel fatto che la decodifica di un messaggio avveniva mediante lo stesso strumento e con la medesima procedura: si batteva il messaggio cifrato sulla tastiera per ottenere in messaggio "in chiaro" sulla striscia di carta. Questo notevole risultato derivava dalla struttura interna dell'ENIGMA.

La cifratura prodotta dall'Enigma era una evoluzione del metodo di cifratura detto "di Giulio Cesare" perchè era stato il condottiero romano ad introdurlo per comunicare in segreto con i suoi generali. Si adotta una permutazione delle lettere dell'alfabeto. Giulio Cesare usava la seguente: al posto di una lettera si scrive quella che la segue di tre posti nell'ordine alfabetico. La parola "EFFETTO" verrebbe trasmessa come "HIIHZZR".

A prima vista, la lettura della parola cifrata scoraggia i curiosi che non conoscono il metodo adottato da Cesare, pur desiderando sapere che cosa Cesare stia trasmettendo. Ma, analizzando con calma la parola, si notano caratteristiche peculiari: ci sono due gruppi di coppie di lettere uguali. In italiano questo accade rarissimamente per vocali, ma molto spesso per consonanti. Dunque, è plausibile pensare che "I" e "Z" siano codici per consonanti distinte. Sotto questa ipotesi, è necessario che "H" sia il codice di una vocale e pure "R" è molto probabile che sia una vocale distinta dalla precedente.

Con queste considerazioni a disposizione, il tentantivo per indovinare la parola diventa ragionevole: oltre alla parola "EFFETTO" che ha effettivamente generato il codice, vi sono altre possibili soluzioni[4].

Ma, ci sono altre due considerazioni che riducono drasticamente le possibili soluzioni: il codice di Cesare *sposta* ogni lettera con lo stesso passo: "H" è il codice di una vocale e "I" è il codice della consonante che *segue* quella vocale in ordine alfabetico. Inoltre, le vocali codificate da "H" e "R" distano tra loro *esattamente* quanto le lettere che le codificano, cioè ci sono sette lettere nell'ordine alfabetico tra le due vocali. Le due vocali possono essere soltanto

4 Le possibili soluzioni, basate sulle prime due considerazioni della struttura della parola in codice, che ho trovato sono:

ABBASSI	ABBASSO	ABBATTE	ABBATTI	ABBATTO
ACCADDE	AFFACCI	AFFANNO	AFFATTO	ALLACCI
ALLATTI	ALLATTO	AMMACCO	AMMASSI	AMMASSO
AMMAZZI	AMMAZZO	ANNAFFI	APPANNI	APPANNO
ARRABBI	ARRAFFI	ARRAFFO	ASSAGGI	ATTACCO
AZZANNI	AZZANNO	ECCELLA	ECCELLI	ECCESSI
ECCESSO	ECCETTO	EFFETTI	EFFETTO	IMMILLA

Si noti come le vocali più frequenti sono A, E ed I, effettivamente le lettere più frequenti in un testo in italiano.

- "A" (codificata da "H") e "I" (codificata da "R"); in questo caso "I" è il codice di "B" e "Z" è il codice di "P". Ma la parola "ABBAPPI" non esiste in italiano.
- "E" (codificata da "H") e "O" (codificata da "R"); in questo caso "I" è il codice di "F" e "Z" è il codice di "T". E la parola "EFFETTO" esiste in italiano.

Per evitare che un attacco per decifrare il messaggio abbia un tale successo, si può pensare di usare una permutazione meno "strutturata" di quella di Cesare, diciamo questa:

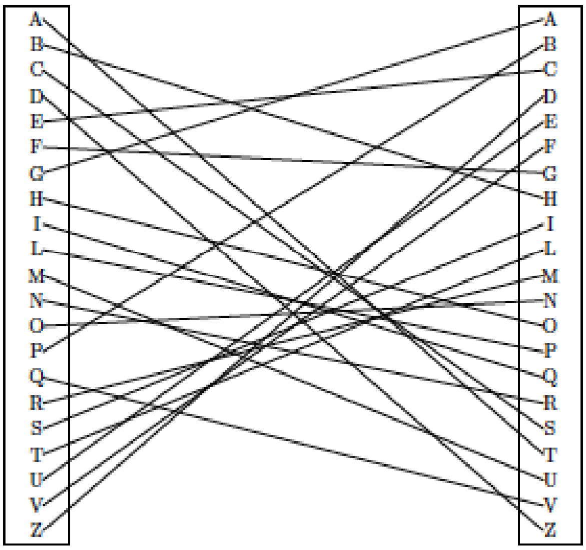

Figura 4. Una possible permutazione per cifrare messaggi.

La parola "EFFETTO" viene codificata come "CGGCLLN", ma la prima parte dell'analisi precedente rimane valida: "G" e "L" sono codici per consonanti distinte, "C" e "R" sono codici di vocali distinte. L'elenco delle possibilità è sempre quello che si trova in appendice. Certo non si riduce all'unica soluzione, ma in un testo di più di una parola si troveranno altri indizi che permetteranno di risalire alla permutazione.

Le macchine ENIGMA contenevano un meccanismo che preveniva un tale attacco: una pressione di un tasto sulla tastiera, oltre a scrivere il codice, produceva una diversa permutazione delle lettere. Il modo era molto ingegnoso: la permutazione finale era ottenuta mediante sei permutazioni ausiliarie del tipo considerato sopra. Ogni permutazione ausiliaria era codificata mediante un segnale elettrico attraverso un rotore che stava all'interno di una macchina ENIGMA. Oltre a tre rotori, c'era una tavola di riflessione che rispediva il segnale elettrico attraverso i rotori. Schematizziamo l'azione di ENIGMA in un caso particolare:

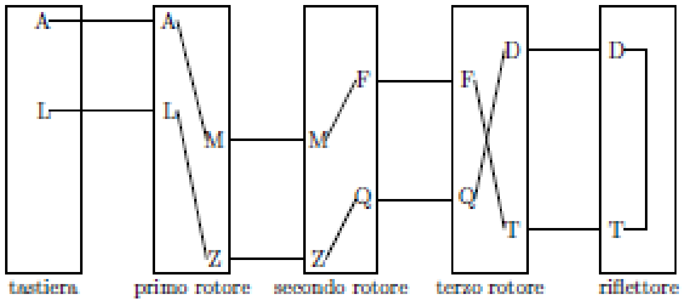

Figura 5 - Come agiva ENIGMA: un esempio.

Prememendo il tasto "A" sulla tastiera, l'operatore manda un segnale al primo rotore che scambia "A" con "M"; a questo punto il segnale arriva al secondo rotore che scambia "M" con "F"; poi il terzo scambia "F" con "T". Ora, il riflettore, ricevendo il segnale "T" lo scambia con "D" e rimanda il segnale attraverso i rotori. Alla fine il codice per "A" che appare sulla tastiera (e viene scritto sulla striscia di carta) è "L".

Ottenere una permutazione come *composizione* di sei permutazioni non aggiunge nulla alla sicurezza del metodo di codifica: è una codifica ottenuta per permutazione come le precedenti. Gli attacchi al codice fatti mediante l'analisi della struttura delle parole in codice rimangono un pericolo come prima. Ma la caratteristica aggiuntiva della macchina ENIGMA sta nei rotori che, dopo la pressione di un tasto e la codifica di una lettera, ruotano *modificando le permutazioni applicate*. Così la prossima lettera nella parola viene codificata con una permutazione completamente diversa dalla precedente.

Un'altra caratteristica è il riflettore: come appare chiaro dallo schema sopra, con i rotori nella medesima configurazione, la lettera "L" viene codificata dalla lettera "A". Cambiando il punto di vista, l'operatore che deve leggere il messaggio in codice ricevuto basta che scriva il messaggio, partendo con i rotori nella medesima configurazione con cui sono stati codificati. Come in precedenza, è molto più utile *vedere* l'esecuzione della codifica e decofica di un messaggio fatto da una macchina Enigma che immaginarne il comportamento: sempra al sito http://www.disi.unige.it/person/RosoliniG/pG/ si trova un simulatore di una tastiera ENIGMA.

L'organizzazione per la spedizione dei messaggi cifrati tedeschi era la seguente: ogni arma distribuiva ai propri operatori un diario con la disposizione iniziale dei tre rotori, assegnata con tre lettere di riconoscimento, per i messaggi, da seguire giorno per giorno. Perciò tutti i messaggi di una giornata venivano codificati (e decodificati) con quella specifica disposizione iniziale. La regola di trasmissione prevedeva poi che un messaggio dovesse iniziare con una sequenza di tre lettere, ripetuta due volte, che indicava un'altra disposizione che l'operatore avrebbe usato da quel momento in poi.

Contro questo castello di permutazioni e parametri modificati, Alan Turing lanciò il suo attacco con le armi della matematica astratta. La pur ingegnosa macchina ENIGMA presentava punti deboli, in effetti più deboli di quanto i suoi costruttori avessero saputo valutare: da come agisce lo schema rotori-riflettore si capisce che una lettera non può mai essere codificata da se stessa, così come la permutazione prodotta è fatta tutta da scambi disgiunti. In più la procedura iniziale di ogni messaggio – scrivere *due volte* la stessa sequenza di tre lettere – permetteva un attacco diretto al codice. Vediamolo attraverso un esempio: supponiamo che un operatore decida di usare la configurazione "BUH". Inizierà a scrivere il suo messaggio, usando i rotori nella configurazione prevista per quel giorno, battendo la sequenza "BUHBUH". Questa parola di sei lettere contiene in posizioni precise, le stesse lettere. Basta dunque analizzare quelle configurazioni iniziali che, dopo tre passi, codificavano la stessa lettera nel modo previsto usando le sequenze iniziali di sei lettere.

Già i matematici del servizio segreto polacco avevano determinato molte delle falle del sistema di cifratura tedesco, ma Turing riuscì a migliorare l'analisi strutturale al punto da rendere possibile ai servizi segreti inglesi di decodificare quasi ogni giorno il traffico radio tedesco. Oltre a sfruttare le proprietà matematiche disponibili, introdusse un sistema di valutazione probabilistica che permetteva di meccanizzare la ricerca del codice corretto: le configurazioni iniziali ricevevano *voti* che valutavano quanto potessero essere sospettate e la ricerca giornaliera si sviluppava seguendo la traccia del maggior sospetto. E questa traccia fu cruciale quando i tedeschi modificarono la procedura di inizializzazione.

4. Macchine pensanti?

Negli anni successivi alla Seconda Guerra Mondiale, laboratori di ricerca inglesi e americani iniziarono una sorta di competizione nella realizzazione di quelli che sarebbero stati i primi calcolatori. Molti degli scienziati che avevano lavorato a progetti scientifici di carattere bellico durante il periodo di guerra, progettando e realizzando macchine per calcoli automatici, mettevano ora a frutto le loro esperienze. Turing collaborò con laboratori inglesi, prima a Londra, poi a Manchester, e durante questo periodo affrontò anche un problema, principalmente di carattere etico e filosofico, piuttosto che tecnico. Considerò il problema di porsi in modo corretto la domanda "le macchine possono pensare?"

Nell'articolo "Computing Machinery and Intelligence"[5], pubblicato nel 1950, Turing propone:

di considerare la domanda: "Le macchine possono pensare?" Questo dovrebbe iniziare con le definizioni del significato dei termini "macchina" e "pensare". Le definizioni potrebbero essere studiate in modo da riflettere il più possibile l'uso comune delle parole, ma un tale metodo è pericoloso. Se si dovesse trovare il significato delle parole "macchina" e "pensare" esaminando come vengono usati normalmente sarebbe difficile evitare la conclusione che il significato e la risposta alla domanda "Le macchine possono pensare?" verranno trovati mediante un sondaggio statistico. Ma questo è assurdo.

Da buon matematico, sa come affrontare il problema di definire un concetto "intuitivamente ovvio"; dopo tutto nel 1936 aveva capito che era necessario definire quell'altro concetto "intuitivamente ovvio" di metodo meccanico per risolvere il problema proposto da David Hilbert–non l'avesse fatto, oggi non ci sarebbero i computer…

In effetti, Turing, avendo introdotto una nozione astratta di "macchina" che si stava rivelando estremamente precisa ed efficace, non ha problemi a spiegare in che senso si può interpretare il termine "macchina". Per quanto riguarda invece il problema di definire il termine "pensare" evita completamente una definizione astratta del concetto, ma prosegue dicendo che

5 "Macchinario di calcolo e intelligenza", vedi [3].

Invece di provare a dare una tale definizione, sostituirò la domanda con un'altra, che è strettamente connessa a quella e che si esprime in termini relativamente non ambigui. Si può descrivere la nuova forma della domanda in termini di un gioco che chiamiamo il "gioco dell'imitazione".
[…] *Si gioca in tre: un uomo* (*A*), *una donna* (*B*) *e un interrogatore* (C) *che può essere di sesso qualunque. L'interrogatore è in una stanza, separato dagli altri due. Lo scopo del gioco per l'interrogatore è quello di determinare chi sia l'uomo e chi la donna. Li conosce come X e Y e alla fine del gioco deve dire che "X è A e Y è B" oppure che "X è B e Y è A". L'interrogatore può fare domande ad A e B di qualunque genere.* […] *Lo scopo di A nel gioco è di far in modo che C sbagli le identificazioni.* […] *Lo scopo di B nel gioco è di aiutare l'interrogatore. La migliore strategia per B è probabilmente di dare risposte vere.*

Il gioco dell'imitazione è diventato un normale passatempo alle feste tra amici. Ma ora Turing si pone la domanda:

Che cosa succede quando una macchina prende il posto di A nel gioco?" Le identificazioni errate dell'interrogatore saranno tante quante quelle fatte nel gioco con un uomo e una donna? Queste domande sostituiscono la domanda originale "Le macchine possono pensare?

È chiarissimo che Turing non risponde alla domanda: intende soltanto porre una questione, che stava sorgendo negli ambienti scientifici, in un modo che gli sembra scinetificamente più corretto di quello che dà per scontato il significato di "pensare".

5. Conclusioni

Il rapidissimo tracciato che abbiamo percorso attraverso alcune realizzazioni di Alan Turing ci ha permesso di analizzare alcuni momenti cruciali del pensiero scientifico del XX secolo. Molto di questo è rimasto sconosciuto – vuoi perchè *Top Secret*, vuoi perchè altamente specialistico–al pubblico, ma le ricadute di quanto egli aveva ottenuto sono state enormi e tangibili per tutta l'umanità. Personalità scientifica di enorme valore, Turing è stato un genio che ha segnato un momento storico dando contributi filosofici e scientifici importantissimi che soltanto ora iniziamo tutti ad apprezzare. Gli spunti che abbiamo presentato prevedono (e pretendono) approfondimenti che rimangono estremamente attuali.

Bibliografia

[1] A. Hodges, *Alan Turing*: *Una biografia*, Bollati Boringhieri, Torino 2006.

[2] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, 42 (1936-7), pp. 230-265; correzioni *ibid*. 43 (1937), pp. 544-546.

[3] A. M. Turing, Computing Machinery and Intelligence, *Mind*, 49, 1950, pp. 433-460.

Giuseppe Rosolini
Università di Genova