

SCIENZE

## ANNIVERSARI/ Interrogate il PC: vi risponderà come una macchina di Turing

Angelo Montanari

lunedì 11 giugno 2012

Se discipline quali la matematica e la fisica possono vantare nutrite schiere di padri nobili, per una disciplina relativamente giovane qual è l'informatica, la celebrazione del centenario della nascita di Alan Turing rappresenta una novità assoluta. Non deve, perciò, sorprendere che, in questo 2012, quello che è unanimemente riconosciuto come il padre dell'informatica sia il protagonista di molteplici iniziative distribuite fra i vari continenti.

La conferenza ufficiale si terrà a Manchester a fine giugno, ospitata dall'università presso la quale Turing lavorò dal 1948 al 1954, anno della sua morte, e vedrà la partecipazione di numerosi vincitori del premio Turing (il premio Nobel per l'informatica a lui intitolato); accanto ad essa, interventi sull'influenza dei risultati di Turing su settori dell'informatica sono stati inseriti nel programma di moltissime conferenze internazionali di area informatica di quest'anno.

La vita di Turing è già di per sé un'avventura: l'infanzia in una famiglia che viveva a cavallo tra India e Inghilterra, gli studi al King's College di Cambridge e i primi lavori in settori tradizionali della matematica, come la teoria delle funzioni quasi periodiche, i lavori fondamentali da cui prenderà forma la scienza informatica, il successivo dottorato a Princeton, con lo studio dei gradi di insolubilità dei problemi, il coinvolgimento nel gruppo impegnato nella decodifica dei codici segreti utilizzati dall'esercito tedesco per le comunicazioni militari durante la seconda guerra mondiale, gli anni del dopoguerra presso l'Università di Manchester, le anticipazioni di temi che diventeranno oggetto di importanti ricerche in intelligenza artificiale, l'omosessualità e il suicidio a 42 anni. In questo articolo ci soffermeremo su alcuni dei suoi contributi scientifici più importanti.

La ricerca di un sistema formale (logico) in grado di catturare tutte le inferenze deduttive comunemente usate nella pratica matematica può essere fatta risalire agli studi di Frege nella seconda metà dell'ottocento. Il nucleo della logica di Frege è nella sostanza quella che viene oggi chiamata la logica del prim'ordine. Una cinquantina di anni dopo, in un lavoro scritto col suo allievo Ackermann, Hilbert solleva due questioni fondamentali relative a tale logica. La prima riguarda la completezza della logica del prim'ordine, ossia la sua capacità di derivare, mediante l'applicazione delle regole di sistema, tutte le formule valide; la seconda la possibilità di sviluppare una procedura in grado di stabilire la validità o meno di una qualsiasi formula della logica.

Questa seconda questione è nota come *Entscheidungsproblem* (problema della decisione). Nella sua tesi di dottorato, Gödel provò la completezza e la correttezza della logica del prim'ordine (teorema di completezza di Gödel), fornendo una risposta positiva alla prima delle due questioni. In virtù di tale risultato, l'*Entscheidungsproblem* può essere riformulato come il problema dell'esistenza di un procedimento di calcolo effettivo (un algoritmo) in grado di stabilire se, dati un certo insieme di premesse e una conclusione, la seconda sia o meno derivabile dalle prime.

Turing iniziò ad occuparsi del problema poco dopo la pubblicazione dei sorprendenti teoremi di incompletezza di Gödel. In essi, Gödel aveva dimostrato che all'interno di ogni sistema formale abbastanza potente da contenere l'aritmetica esistono proposizioni che il sistema non riesce a decidere, non riesce, cioè, a fornire una dimostrazione né di esse né della loro negazione. Inoltre, fra le proposizioni che tale sistema non riesce a dimostrare c'è anche quella che esprime la non-contraddittorietà (coerenza) del sistema.

Tali risultati rendevano improbabile l'esistenza di un algoritmo per *Entscheidungsproblem*. Turing concentrò, pertanto, il suo sforzo sulla ricerca di una dimostrazione di tale impossibilità e nel tentativo di fornire un modello adeguato del processo di calcolo arrivò a concepire quelle che da allora vengono chiamate le macchine di Turing. Una macchina di Turing manipola un insieme di dati contenuti nelle celle di un nastro di lunghezza infinita (in ogni fase della computazione, solo il contenuto di una porzione finita di tale nastro è significativo) sulla base di un insieme finito di regole prefissato.

La tesi di Turing-Church afferma che per ogni funzione calcolabile mediante un processo algoritmico, esiste una macchina di Turing che calcola tale funzione. A conferma di tale tesi, comunemente accettata, è stata dimostrata l'equivalenza di tutti i modelli di calcolo alternativi proposti in letteratura alla macchina di Turing. Sulla base di tale tesi, il problema originario può essere ridotto all'esistenza di una macchina di Turing in grado di risolverlo. Dalla prova della non esistenza di una tale macchina, segue l'indecidibilità dell'*Entscheidungsproblem*.

Sulla base di un'opportuna codifica numerica delle macchine di Turing e dei loro dati di input (codifiche analoghe erano state già utilizzate da Cantor e Gödel), Turing mostrò come il problema di stabilire se, data una macchina di Turing  $M$  e un insieme di dati di input  $I$  per essa, l'esecuzione di  $M$  su  $I$  termina o meno sia insolubile, ossia non esista un algoritmo (una macchina di Turing) in grado di risolverlo. La prova sfrutta il metodo della diagonale usato da Cantor per dimostrare che i numeri reali (infiniti) sono "più" dei numeri naturali (anch'essi infiniti). Dato che il problema dell'arresto della macchina di Turing può essere espresso in logica del prim'ordine, la non esistenza di un algoritmo per l'*Entscheidungsproblem* segue immediatamente (se un tale algoritmo esistesse, potrebbe essere usato per risolvere il

problema dell'arresto della macchine di Turing, che Turing ha mostrato essere indecidibile). Nei decenni successivi, tale tecnica di riduzione è stata impiegata con successo per dimostrare l'indecidibilità di un grande numero di problemi algoritmici di natura assai diversa.

Una conferma definitiva della potenza della nozione di macchina di Turing venne dall'idea rivoluzionaria di una macchina di Turing universale. La tesi di Turing-Church afferma l'esistenza di una macchina di Turing per ogni funzione calcolabile. La macchina di Turing universale è una particolare macchina di Turing che riceve in input una descrizione di una qualsiasi (altra) macchina di Turing  $M$  e di un input  $I$  per essa e restituisce in output il risultato dell'elaborazione di  $I$  da parte di  $M$ . Turing provò l'esistenza di una tale macchina in modo costruttivo, ossia fornendo l'insieme di regole che la governano. Una delle novità radicali dell'idea di macchina universale è la rimozione della tradizionale distinzione tra macchine e dati. Fra i suoi dati di input, la macchina di Turing universale ha la descrizione di una (altra) macchina di Turing. Se i calcolatori così come noi li conosciamo sono assai diversi dalla macchina di Turing (la loro architettura si basa su un modello di calcolo alternativo equivalente proposto da von Neumann), dal punto di vista concettuale non c'è alcuna differenza: il calcolatore è una macchina di Turing universale.

Nell'ultimo periodo della sua vita, Turing si occupò di questioni che di lì a pochi anni sarebbero diventate l'oggetto di interesse di uno dei settori più rappresentativi dell'informatica quale l'intelligenza artificiale. In particolare, si interrogò circa la possibilità per un calcolatore di sviluppare un'intelligenza di tipo umano.

In un articolo apparso sulla rivista *Mind* nel 1950, egli propose un test, o gioco dell'imitazione, oggi noto come test di Turing, basato sulla seguente assunzione: una macchina può essere definita intelligente se riesce a convincere una persona che il suo comportamento, dal punto di vista intellettuale, non è diverso da quello di un essere umano medio. Il test si svolge in tre stanze separate. Nella prima si trova l'esaminatore umano (A); nelle altre due vi sono rispettivamente un'altra persona e il computer che si sottopone al test. Dei due A conosce i nomi (B e C), ma ignora chi sia la persona e chi il computer. Sia B che C si relazionano separatamente con A attraverso un computer. Via computer A può porre domande a B e C e leggere le loro risposte. Compito di A è scoprire l'identità di B e C (chi è la persona, chi è la macchina?) entro un limite di tempo prefissato.

A può effettuare qualunque tipo di domanda. Il computer ovviamente cercherà di rispondere in modo tale da celare la propria identità. La macchina supera il test se A non riesce a identificarla nel tempo prefissato. Il test verrà ripetuto più volte, coinvolgendo anche esaminatori diversi, in modo da ridurre i margini di soggettività. Anche se risulta del tutto evidente l'influenza del funzionalismo e del comportamentismo nella formulazione del test, esso presenta diversi elementi di interesse. Fra questi, vogliamo sottolineare lo stretto legame che esso stabilisce tra intelligenza e capacità linguistiche: esso si basa su una interpretazione operativa/comportamentale dell'intelligenza che si manifesta attraverso la comunicazione linguistica. Una confutazione indiretta della validità del test di Turing verrà fornita alcuni decenni dopo dal famoso argomento della stanza cinese di Searle, ma questa è già un'altra storia.