Gödel e Turing. La nascita del computer e la società dell'informazione

Kurt Gödel

Kurt Gödel nacque a Brno, in Moravia, il 28 aprile 1906. Figlio di un industriale tessile, svolse i suoi studi nella città natale mostrando una vivace intelligenza e interessi molto diversificati. Nel 1924 raggiunse il fratello che frequentava la facoltà di medicina a Vienna per studiare matematica e filosofia, concentrando poi i suoi interessi sulla logica matematica; in quegli anni (1926-28) iniziò a frequentare il *Circolo di Vienna* (*Wiener Kreis*) composto da un gruppo di filosofi e scienziati che si riuniva attorno a Moritz Schlick, Hans Hahn e Rudolf Carnap, mantenendo fin da allora una distanza critica dalle idee del positivismo logico. Dopo la laurea nel 1930, ottenne una docenza all'università di Vienna (1933) e nello stesso anno fu invitato negli Stati Uniti all'*Institute for Advanced Study di Princeton*. In questo periodo iniziò a soffrire di disturbi mentali che si manifestavano in una forma di ipocondria, ossessione per la dieta, paura di avvelenamenti, che lo porteranno alla denutrizione. L'uccisione di Schlick nel 1936 da parte uno studente nazista e il precipitare degli eventi politici, con l'Anschlüss e la guerra, muteranno profondamente il corso della sua esistenza. Solo il matrimonio con la ballerina Adele Porkert nel 1938 sembrò alleviare le sofferenze del suo animo e la sua ipocondria. Trasferitosi definitivamente a Princeton nel 1940, nel 1948 ottenne la cittadinanza statunitense e divenne professore ordinario nel 1953. Le sue condizioni di salute non migliorarono e morì all'ospedale di Princeton il 14 gennaio 1978 di «*denutrizione e debilitazione derivanti da disturbi della personalità*».

I suoi lavori sono fondamentali dal punto di vista metodologico. Dopo aver dimostrato la completezza semantica della logica dei predicati, enunciò nel 1931 il teorema (teorema di incompletezza di Gödel) secondo cui la coerenza di un sistema formale è indimostrabile all'interno del sistema stesso, ponendo così un limite all'attuale concetto di decidibilità. Fra le sue conseguenze sono la generale affermazione di incompletezza delle teorie formali (in antitesi alle teorie hilbertiane) e, in intelligenza artificiale, l'esclusione teorica che l'elaborazione automatica, per quanto complessa, possa sostituirsi al cervello umano. Tra i suoi scritti: *Proposizioni formalmente indecidibili dei «Principia mathematica» e di sistemi affini* (1931); *La consistenza dell'ipotesi del continuo* (1940).

La macchina per fare calcoli

Se pensiamo al Settecento, e ci chiediamo qual è l'invenzione più importante fatta in quel secolo, subito il nostro pensiero corre all'orologio meccanico. Dagli esperimenti di Galileo e di Newton sul pendolo, e attraverso le invenzioni di Huygens e di tanti altri è nato, appunto, l'orologio meccanico. Che è diventato anche una metafora filosofica, perché ha permesso di costruire una metafisica meccanicistica dell'universo e del corpo umano.

Se ci facciamo la stessa domanda, chiedendoci qual è l'invenzione che ha caratterizzato il passaggio dal secondo al terzo millennio, la risposta non può che essere il computer, la macchina per i calcoli universale, quello che una volta veniva chiamato «cervello elettronico».

In questo volume vogliamo raccontare la storia di come è nata l'idea del computer. Spesso le tecnologie arrivano sulla scena prima della teorizzazione delle loro basi teoriche: si teorizza in seguito, per capire meglio quello che già si è fatto in pratica. Invece con il computer è successo esattamente il contrario: ha avuto prima una gestazione scientifica. e addirittura filosofica, di alcuni secoli, e poi è diventato una realizzazione tecnologica nel Novecento.

Si potrebbe partire da molto lontano, perché già gli antichi greci avevano costruito meccanismi che potrebbero essere considerati lontani antesignani del computer. Ma limitiamoci a iniziare dal Duecento, quando un signore di nome Raimondo Lullo sognò di costruire una macchina che permettesse di elaborare ragionamenti e, in subordine, di fare calcoli.

Lullo costruì uno strumento formato da cerchi concentrici, sui quali aveva disposto tutti i concetti che gli potevano venire in mente, nel tentativo di realizzarne una combinatoria. Riponeva grandi speranze in questa sua invenzione, tanto che pensò addirittura di usarla per convertire gli

infedeli. Andò nei paesi arabi e finì malissimo, perché gli tagliarono la testa. Questo primo sogno finì nel momento in cui la testa che lo sognava rotolò per terra.

Qualche secolo dopo, nel fatidico anno 1666, lo stesso in cui Newton stava concependo i germi delle sue teorie matematiche e fisiche, un celebre filosofo di nome Leibniz concepì un'idea simile a quella di Lullo, però questa volta in una forma più compiuta. Scrisse un libro intitolato *Ars combinatoria* (Lullo aveva intitolato il suo *Ars magna*), in cui sognava un meccanismo che permettesse di «calcolare i ragionamenti».

Oltre che un matematico, Leibniz era un diplomatico: girava l'Europa a parlamentare nelle corti, e il suo sogno era di non dover più discutere con coloro che rappresentavano gli interessi dei re o dei regni, ma di potersi sedere a tavolino e dire *calculemus*, «calcoliamo». La sua idea era dunque di non dover più ragionare, ma semplicemente calcolare, o addirittura far calcolare uno strumento in vece sua.

Questo sogno fu realizzato poi dal computer, nella storia che stiamo incominciando a raccontare. Facendo un balzo in avanti di circa centocinquant'anni, troviamo nell'Inghilterra del 1837 la prima macchina che possiamo considerare un' antesignana dei moderni calcolatori. Ovviamente, non era basata su circuiti elettronici, che ancora non esistevano: era in pratica un computer a vapore, realizzato da un uomo di nome Charles Babbage.

In realtà fu realizzata solo la sua prima versione, la «macchina differenziale», una sorta di potente calcolatrice. Il progetto più ambizioso di Babbage, la «macchina analitica», in grado di essere programmata tramite algoritmi, in maniera simile a un moderno calcolatore elettronico, non fu mai completata, a causa delle limitazioni tecnologiche dell'epoca.

Babbage non ebbe successo e il suo nome cadde nel dimenticato io. Soltanto centocinquant' anni dopo l'Inghilterra si accorse di aver cullato nel suo seno qualcuno che avrebbe potuto farla entrare nell' era dell'informatica con oltre un secolo di anticipo.

Boole e la logica matematica

Il primo pensatore della modernità che iniziò a realizzare il sogno di Leibniz, ed ebbe grande influenza sugli sviluppi successivi, fu ancora un inglese, George Boole. Nel 1847 e nel 1854 egli pubblicò due libri fondamentali, che rappresentarono il primo passo teorico fondamentale per arrivare a costruire una macchina in grado di «pensare». Il contributo di Boole è oggi chiamato «algebra booleana», in orfore del suo inventore, che vi costruì sopra tutta una filosofia.

Il suo secondo libro si intitolava *Le leggi del pensiero*, e suggerisce che egli fosse riuscito a isolare da un punto di vista matematico le leggi che sottostanno alla nostra attività raziocinante. In realtà, quello che fece Boole è molto semplice, e possiamo provare a spiegarlo brevemente.

Le affermazioni che facciamo possono essere soltanto di due tipi: vere oppure false. Se ignoriamo il loro contenuto semantico, e ci concentriamo soltanto sul loro essere vere o false, possiamo identificare tutte le proposizioni vere con un numero, per esempio 1, e tutte le proposizioni false con un altro numero, per esempio 0.

Ora, il problema è: come si possono rappresentare matematicamente le operazioni del linguaggio che corrispondono a quelli che tecnicamente si chiamano «connettivi»? I connettivi sono la negazione, la congiunzione, la disgiunzione, l'implicazione ecc.; cioè, i modi di unire frasi semplici per costruirne di più complesse. Cominciamo con il connettivo più semplice, la negazione, Qual è la proprietà fondamentale della negazione? E rendere falsa una frase vera, e viceversa. Abbiamo detto che le frasi false per noi sono tutte identificate col numero O, mentre le frasi vere sono identificate col numero 1. Quindi, la negazione dev' essere un' operazione matematica che, applicata al numero 1, restituisce il numero 0, passando dalla verità alla falsità, ma, applicata al numero 0, restituisce il numero 1, passando dalla falsità alla verità. Quale sarà l'operazione che trasforma lo 0 in 1 e l' 1 in 0? Se ci pensate un momentino, è molto semplice: è «1-», perché 1 - 1 fa 0, mentre 1 - 0 fa 1.

Boole capì che questo «1 -» era la struttura matematica, algebrica, della negazione. Si comportò in modo simile con la congiunzione, per fare un altro esempio. Quand'è che la congiunzione di due

affermazioni è vera? Se tutte e due sono vere. Ma basta che una delle due affermazioni sia falsa, perché sia falsa anche la congiunzione.

E qual è l'operazione che prende due proposizioni vere, cioè due 1, e restituisce come valore 1, una proposizione vera? Ma che quando prende due numeri, almeno uno dei quali è uno 0, cioè almeno una delle due proposizioni è falsa, restituisce come risultato lo 0, cioè una proposizione falsa? L'ho fatta lunga, ma la risposta è semplice: questa operazione è la moltiplicazione. Infatti 1 * 1 fa 1, ma 1 * 0, così come 0 * 1 e 0 * 0, fanno sempre 0.

Abbiamo capito che la sottrazione corrisponde alla negazione, e la congiunzione corrisponde alla moltiplicazione, e andando avanti così si possono ricostruire tutti i connettivi. Esistono dunque operazioni linguistiche sulle proposizioni, che corrispondono a operazioni matematiche sui numeri: queste operazioni, linguistiche e matematiche, sono in qualche modo fra loro isomorfe, hanno cioè la stessa struttura.

Boole scoprì che questo era l'essenza, il nucleo centrale, della questione. Potremmo definirlo il granello di sabbia che, messo all' interno dell'ostrica della logica, avrebbe poi prodotto la perla di cui stiamo parlando: la logica matematica, che costituisce le fondamenta dell'informatica.

Alla ricerca delle leggi del ragionamento

Abbiamo accennato all'opera di Boole, che aprì la strada per esprimere in linguaggio matematico la logica dei filosofi, cioè la struttura del pensiero. Passarono pochi anni, e nel 1879, questa volta in Germania, un filosofo tedesco, di nome Gottlob Frege, elaborò quella che chiamò «ideografia», *Begriffischrift*. Fece, cioè, un tentativo di costruire un linguaggio puramente simbolico.

Boole aveva identificato tutte le affermazioni con lo 0 e con l'1. Frege cercò invece di restituire la struttura del linguaggio naturale attraverso un altro linguaggio, completamente formalizzato, di cui fosse possibile descrivere fin dagli inizi quali erano i costituenti fondamentali, gli assiomi. E come potevano essere messi assieme per costruire proposizioni via via più complesse. E quali erano le regole fondamentali per dedurre proposizioni vere da altre proposizioni vere. Era un tentativo di assiomatizzazione della logica, analogo a quello che aveva fatto Euclide per la geometria.

Frege riuscì nel suo intento, ed enunciò alcuni assiomi fondamentali, le leggi fondamentali della logica, che sono poi le stesse formalizzate da Boole attraverso procedimenti algebrici. Una di queste leggi, tanto per fare un esempio, era il famoso modus ponens, già identificato da Aristotele e dagli stoici: la legge, cioè, secondo cui se sappiamo che A è vero, e che da A discende B, allora sappiamo che anche B è vero.

Assiomi e regole possono essere immaginati in grande quantità, ma Frege riuscì a isolare alcuni assiomi e alcune regole che, a suo dire, potevano essere definitivi, nel senso che permettevano di derivare tutte le verità logiche. Fu un passo da gigante, perché riuscire a isolare gli assiomi e le regole necessari per il ragionamento, è qualche cosa di importante.

Ma il problema che lui stesso si pose, e naturalmente anche noi ci possiamo porre a questo punto, è: Frege aveva ragione? Era riuscito veramente a isolare quello che serviva per dedurre, per ricavare tutte le verità logiche, oppure si era dimenticato di qualcosa?

Prima di rispondere a questa domanda, dobbiamo dire che Frege non fu l'unico in realtà a fare questo lavoro. Molti anni dopo la sua opera, che è del 1879, e precisamente tra il 1900 e il 1902, ci fu anche qualcun altro che cominciò a lavorare nello stesso campo e allo stesso problema. Questo signore era un connazionale di Boole, e divenne poi molto più famoso, perlo meno in quegli anni, di Frege stesso: si chiamava Bertrand Russell.

Il punto di partenza della carriera di Bertrand Russell fu lo stesso di Frege: cercare di isolare un sistema assiomatico per il pensiero. E il suo punto d'arrivo fu anch'esso lo stesso: Russell, infatti, arrivò a un sistema che era equivalente a quello di Frege.

Quindi, a questo punto, la domanda diventava: ma il sistema di Frege, e quello equivalente di Russell, sono effettivamente un modo per descrivere tutte e sole le verità logiche?

Logica e matematica

Per risolvere questo problema, per cercare di capire se effettivamente gli assiomi e le regole di Frege e di Russell erano complete, nel senso che tutte le verità logiche si potevano derivare dagli assiomi attraverso le regole, bisognò aspettare alcuni decenni. E la soluzione arrivò nel 1930, quando sulla scena della logica matematica, anzi sulla scena della matematica in generale, fece irruzione Kurt Gödel.

Il nome di Gödel è composto di God ed El, cioè dei nomi inglese ed ebraico di Dio. E lui fu effettivamente la divinità moderna della logica. E stato il più grande logico mai esistito, l'unico che possa essere paragonato a lui fu Aristotele nell'antichità. E che cosa fece Gödel per meritarsi l'appellativo di «*Dio della logica*»? E per arrivare a nel 2000 a vincere il titolo di «*matematico del secolo*», attribuitogli dalla rivista «*Time*»?

Da una parte dimostrò nel 1930, a ventiquattro anni, un famoso teorema, che si chiama «*teorema di completezza*». Questo teorema dice che il sistema di Frege, e quello equivalente di Russell, sono effettivamente completi: dai loro assiomi, attraverso le loro regole, si possono derivare tutte e sole le verità logiche. Detto altrimenti, la logica è completa.

Nel 1931 Gödel dimostrò invece il «*teorema di incompletezza*»: ovviamente, non più per la logica, ma per la matematica. In sostanza, mentre esistono sistemi completi per la logica, non esistono sistemi di assiomi e di regole che permettano di derivare tutte e sole le verità matematiche.

Oltre ai loro sistemi per la logica, sia Frege che Russell avevano cercato di individuare dei sistemi di assiomi e regole per l'aritmetica. Una delle regole aritmetiche più ovvie è la cosiddetta «regola di induzione», che permette di derivare una formula per tutti i numeri, nel momento in cui la si sa derivare per il numero 0, e si sa passare dal fatto che la formula è vera per un numero al fatto che è vera per il numero successivo. Infatti, per il numero 0 la formula è vera per ipotesi. Per il numero l la si deriva dal numero 0, per il numero 2 dal numero l, e così via. Cioè, un passo per volta, la si deriva per tutti i numeri.

Questa regola, insieme ad altre, era uno dei costituenti dei sistemi di Frege e di Russell per l'aritmetica. Ma nel 1931 Gödel dimostrò che i loro sistemi non erano completi, nel senso che non permettevano di derivare alcune verità aritmetiche. Anzi, non solo quei sistemi erano incompleti: lo erano tutti quelli che si potevano proporre! E appunto questo il teorema che fece diventare Gödel il matematico più famoso del Novecento: l'aver dimostrato che l'aritmetica è incompleta di natura, e che non esistono sistemi completi per le sue verità.

Consistenza e completezza

Per cercare di capire qual è l'essenza di questo famoso teorema di incompletezza di Gödel, facciamo un passo indietro e andiamo a rivedere per un momento la filosofia kantiana, in particolare la *Critica della ragion pura*.

Kant si propone il problema della logica, che lui chiama «ragion pura». E nota che la logica tende a considerare idee che oggi noi chiameremmo «al limite», e che lui chiamava, nel suo colorito linguaggio, «idee trascendentali». Queste idee trascendentali sarebbero il mondo, l'anima e Dio.

L'essenza del ragionamento di Kant è che, se noi vogliamo parlare di ciascuna di queste tre idee, allora finiamo nei guai: cadiamo, cioè, nelle famose antinomie della *Critica della ragion pura*. A un certo punto del libro si trovano delle pagine parallele, in cui si dimostrano cose opposte in pagine opposte. Si arriva, cioè, a delle contraddizioni.

Per esempio, a proposito del mondo, Kant dimostra in una pagina che esso deve aver avuto un inizio nel tempo e dev'essere limitato nello spazio. Ma nella pagina opposta dimostra che il mondo non può aver avuto un inizio nel tempo e non puo avere limiti nello spazio. Ed è chiaro che queste due cose non vanno d'accordo, non possono essere vere allo stesso tempo: si tratta, appunto, di una contraddizione, di un' antinomia.

Che cosa ha dunque dimostrato, Kant? In sostanza, ha dimostrato che se la ragione vuol essere completa, se pretende di poter trattare delle idee trascendentali, allora dev'essere inconsistente, perché finisce per cadere nelle antinomie. Detto in breve, la completezza implica la inconsistenza.

Un modo equivalente di dire la stessa cosa è che, se la ragione vuol essere consistente, se non vuole cadere nelle antinomie, allora deve accettare di essere incompleta. In particolare, deve evitare di parlare delle idee trascendentali. L'impianto della Critica della ragion pura si riduce dunque a. questo: se vogliamo avere la consistenza, se non vogliamo avere antinomie, allora non possiamo avere la completezza.

E questo è esattamente il contenuto del teorema di Gödel. Solo che, invece di dimostrarlo alla maniera filosoficamente farraginosa di Kant, con un'opera di settecento pagine, Gödel lo dimostra in una maniera matematicamente luciferina, con un articolo di una ventina di pagine. E lo dimostra non per l'astratta «ragion pura», che non si sa bene cosa sia, ma per i concreti sistemi formali della matematica.

Il contenuto del teorema di Gödel del 1931 è, dunque, che se un sistema matematico vuol essere consistente, allora dev'essere incompleto. Ci devono, cioè, essere delle verità che non è possibile dimostrare a partire dagli assiomi, e seguendo le regole del sistema.

Un curioso paradosso

Qual è la struttura del ragionamento di Gödel? Naturalmente c'è un'analogia con ciò di cui abbiamo appena parlato: la *Critica della ragion pura* e il ragionamento di Kant. Gödel fa un ragionamento molto semplice, simile a quello di Kant, ma ispirato ai vecchi paradossi dell'antichità: il paradosso del mentitore, in particolare.

Il paradosso del mentitore fu scoperto da Eubulide di Mileto, verso il IV secolo prima della nostra era, e sostiene semplicemente che ci sono dei problemi a dire: «io mento». I problemi stanno nel fatto che quella frase dovrebbe essere o vera o falsa. Ma se è vera, è vero che «io mento», e dunque sto dicendo il falso. E se è falsa, allora è falso che «io mento», e dunque sto dicendo il vero. E questo è un paradosso: se la frase è vera, allora è falsa, e se è falsa, allora è vera. Si entra in un circolo vizioso.

Gödel fece una specie di gioco equilibristico, quasi da prestigiatore, prendendo una frase analoga a quella di Eubulide, che adesso vedremo, e dimostrando che si otteneva non più un paradosso, bensì un teorema. La frase di Gödel è: «*io non sono dimostrabile*». Attenzione, non «*io non sono vera*», come diceva la frase di Eubulide, bensì «*io non sono dimostrabile*».

La frase di Gödel dice di se stessa di non essere dimostrabile. Supponiamo che lo sia: cioè, che sia dimostrabile in uno dei sistemi tipici della matematica, sistemi in cui si dimostrano solo cose vere. Se la frase fosse dimostrabile, poiché il sistema dimostra solo cose vere, sarebbe vera. Ma essa dice «*io non sono dimostrabile*», e allora non dovrebbe essere dimostrabile.

L'ipotesi che questa frase sia dimostrabile fa sì che essa sia vera, e dunque non dimostrabile. Ma allora, l'ipotesi è sbagliata, e la frase non può essere dimostrabile; ma questo è appunto ciò che essa dice, e dunque è vera.

Sembra un gioco di prestigio, sul quale naturalmente vi invito a riflettere, a meditare. Perché, detta così, sembra semplicemente un gioco di parole, e non la dimostrazione di un teorema. Vi invito, cioè, a meditare sul fatto che la frase «*io non sono dimostrabile*» non può essere dimostrabile in un sistema che dimostra solo verità, perché altrimenti sarebbe falsa. E allora non è dimostrabile, ma poiché dice appunto di non essere dimostrabile, è vera.

E allora, che cosa abbiamo trovato? Abbiamo un'affermazione che è vera, ma non dimostrabile: dunque un esempio di incompletezza del sistema. Ora, come mai questo non è vero già per la logica, ma solo per la matematica? Perché il succo, il nucleo della dimostrazione di Gödel consiste nel far vedere che questa frase, che dice di se stessa di non essere dimostrabile, si può rendere precisa, ma soltanto trascrivendo la in linguaggio matematico, attraverso un meccanismo che si chiama oggi «gödelizzazione», per ovvi motivi.

La gödelizzazione è un modo per trascrivere frasi del linguaggio in formule aritmetiche, e dunque matematiche. Ed è così che questo ragionamento linguistico e filosofico, che assomigliava molto ai vecchi paradossi da una parte e ai ragionamenti di Kant dall'altra, diventa un ragionamento matematico.

Però, per poter fare la trascrizione, c'è bisogno di avere almeno i numeri nel proprio sistema. Non lo si può fare nella sola logica, che è troppo debole. Ma questa debolezza risulta essere una forza, perché diventa la difesa contro l'incompletezza: è per questo motivo, che Gödel ha potuto dimostrare nel 1930 il teorema di completezza per la logica.

Ma non appena c'è un minimo di matematica, non appena c'è un pò dell'aritmetica dei numeri dentro il sistema, allora si può riprodurre questo nuovo teorema di Gödel del 1931, ed ecco che la matematica risulta invece essere incompleta. Abbiamo quindi, da una parte, la completezza della logica, e dall' altra parte, l'incompletezza della matematica.

Alan Turing

Alan Mathison Turing nacque a Londra il 23 giugno 1912. Figlio di un funzionario dell'amministrazione coloniale, crebbe presso una famiglia alla quale venne affidato durante i lunghi soggiorni dei genitori in India. Dimostrò da subito un grande interesse per le materie scientifiche e nel 1931 venne ammesso al *King's College* dell'università di Cambridge dove studiò meccanica quantistica, logica e teoria della probabilità, laureandosi nel 1934. Nel 1936-38 frequentò l'*Institute for Advanced Study di Princeton* dove conseguì un master. Le sue fondamentali ricerche in logica matematica e sulla computabilità costituiscono uno dei punti di partenza dell'intelligenza artificiale. Definì il concetto intuitivo di funzione computabile e di algoritmo (1937), tramite una macchina astratta cui è rimasto legato il suo nome e che costituisce un precursore teorico del calcolatore elettronico, e fornì una dimostrazione dell'indecidibilità del calcolo predicativo puro.

Tornato in Gran Bretagna, durante la seconda guerra mondiale applicò le sue ricerche alla messa a punto di una macchina in grado di decodificare i messaggi cifrati tedeschi con significativi successi. Nel dopoguerra si dedicò a una nuova ricerca partendo dall'idea che si potesse creare una macchina intelligente seguendo gli schemi del cervello umano. Nel 1952 venne arrestato e condannato per omosessualità (all'epoca considerata un reato nel suo paese) e per non essere incarcerato si sottopose a una cura a base di estrogeni che lo rese impotente. La condanna lo escluse anche dalla ricerca scientifica perché gli venne annullato il nullaosta dei servizi segreti che temevano potesse essere ricattabile e divulgasse segreti militari all'Unione Sovietica. Ipersensibile, incompreso, isolato, si suicidò a Manchester il 7 giugno 1954 mangiando una mela in cui aveva inserito una dose di cianuro.

Una drammatica vicenda

Sono sicuro che dopo aver sentito questo abbozzo di dimostrazione del teorema di Gödel, la maggior parte di voi avrà la testa che gira. Questo è ciò che succede sempre, agli inizi, non appena si vedono i paradossi in generale, ma soprattutto questi «quasi sofisrni» matematici in particolare. Ma non bisogna spaventarsi, ovviamente: sono cose sottili, complesse, complicate, sulle quali bisogna meditare.

E colui che ci meditò nella maniera più approfondita fu un giovane inglese che si chiamava Alan Turing. Nel 1936, quindi pochi anni dopo la dimostrazione del teorema di Gödel, Turing dovette scrivere la sua tesi di laurea: aveva solo ventiquattro anni, e questo dimostra che la matematica è effettivamente, come disse una volta Godfrey Hardy, uno «sport da giovani».

Alan Turing cercò di capire quello che Gödel aveva fatto, e per farlo scrisse una tesi che divenne il fondamento di quello che stiamo cercando di raccontare: la storia dell'informatica moderna. Turing è un personaggio meraviglioso, un personaggio sul quale è addirittura stato fatto un film, Enigma, che racconta una piccola parte della sua vita.

La storia risale ai primi anni Quaranta, durante i quali Turing fu a capo di un laboratorio a Bletchley Park, in cui si riuscirono a decifrare i codici segreti nazisti. Questi codici erano ottenuti attraverso una macchina che si chiamava, appunto, Enigma. E il film racconta la storia dell'Enigma, di questa macchina a rotori che il team guidato da Turing riuscì a decodificare. E questa decodifica diede agli inglesi e agli alleati un vantaggio essenziale nel corso della guerra.

Turing ebbe una fine piuttosto tragica, a soli quarantadue anni. Era omosessuale, in un periodo in cui in Inghilterra l'omosessualità era proibita, anche fra adulti consenzienti. Nei primi anni Cinquanta, una sera egli rimorchiò un ragazzino, se lo portò a casa, e la mattina dopo scoprì che il ragazzino gli aveva rubato dei soprammobili in casa.

Turing, molto ingenuamente, andò a denunciare il furto, e disse che credeva che a commetterlo fosse stato il ragazzino che aveva rimorchiato. Così confessò lui stesso un crimine, e fu arrestato, processato e condannato. Ma essendo un eroe di guerra, anche se all'epoca la cosa non era ancora di dominio pubblico, il governo permise al tribunale di offrirgli un'alternativa.

Invece di una pena detentiva, gli fu offerta una cura per «guarire» dall'omosessualità. Una cura di ormoni femminili, che Turing accettò di fare, ma che gli fece, tra l'altro, crescere il seno e cadere la barba. E lui, a un certo punto, decise che ne aveva abbastanza dell'Inghilterra e dei suoi pregiudizi, e si suicidò.

Però era molto legato alla madre. E non volendo che lei venisse a sapere, o potesse pensare, che lui si era suicidato, scelse una via molto strana. Decise di fare come Biancaneve: intinse una mela nel veleno, e la mangiò. Fece quindi apparire il suo suicidio come un incidente. Ed è proprio questa mela a cui manca un morso, che è oggi diventata il simbolo della Apple: la si vede su tutti i gadget della Apple, e ricorda il morso che ha provocato la morte di Turing.

La macchina di Turing

Che cosa fece questo personaggio singolare, che risponde al nome di Alan Turing, per meritarsi un posto nella storia dell'informatica, ed essere ricordato obliquamente nel simbolo di una delle maggiori aziende di computer?

Abbiamo accennato al fatto che Turing si mise a studiare il teorema di Gödel, il teorema di incompletezza della matematica, e cercò di capirlo a modo suo. Questo teorema parlava di sistemi matematici, e quindi di formule, assiomi, regole di deduzione ecc.: tutte cose molto poco intuitive, allora come ora. Turing cercò di riformulare il teorema di Gödel in una forma più intuitiva, e lo fece inventandosi un particolare tipo di macchina.

Egli iniziò chiedendosi come sarebbe possibile descrivere una macchina in grado di fare dei calcoli, e per la quale valesse un analogo del teorema di Gödel. E così facendo arrivò semplicemente a fare sulla carta un progetto di quello che oggi noi chiamiamo «computer», ma che gli informatici chiamano «*macchina di Turing universale*».

Turing notò che quando in un sistema formale si procede dagli assiomi mediante le regole, lo si fa effettivamente in maniera meccanica: cioè, l'uomo che sta derivando dei teoremi, sta lavorando come una macchina. E che cosa dovrebbe saper fare una macchina, per essere in grado di simulare ciò che sta facendo l'uomo mentre esegue deduzioni meccaniche e formali?

Dovrebbe, anzitutto, avere qualche cosa su cui scrivere, e con cui scrivere. E allora Turing decise che la sua macchina doveva avere una testina in grado di stampare delle lettere, o di cancellarle. Doveva avere un pezzo di carta, che lui si immaginò come un nastro infinito. E doveva saper compiere un certo numero di operazioni, quali saper leggere un simbolo, muovere la testina a destra o a sinistra lungo il nastro, cancellare, riscrivere altri simboli ecc.

Finora, una specie di macchina da scrivere sarebbe in grado di compiere operazioni simili. Ma la cosa fondamentale che Turing capì è che, oltre a questo armamentario che oggi noi chiameremmo hardware, cioè oltre alla macchina fisica, ci doveva essere qualche cosa che invece oggi noi chiamiamo software, cioè un programma.

Un programma è una serie di istruzioni che dicono alla macchina, che sta leggendo un simbolo e si trova in un certo stato interno, che cosa deve fare: per esempio, scrivere un altro simbolo, spostarsi a destra o a sinistra ecc. Le possibili azioni sono un numero finito, e un insieme finito di istruzioni di questo genere Turing lo chiamò appunto un programma.

L'osservazione fondamentale è che una macchina anche banale, come quella che abbiamo appena descritto, in grado soltanto di scrivere e leggere su un nastro un numero finito di simboli, e di trovarsi in un insieme finito di stati interni, è in grado di eseguire qualunque programma, e dunque

di fare tutto ciò che fa un matematico quando calcola. Perché Turing capì che questi programmi, benché a prima vista rudimentali, erano in grado di descrivere qualunque operazione meccanica, qualunque funzione calcolabile, qualunque algoritmo informatico.

Ora, poiché questi programmi sono oggetti finiti, analoghi a un testo composto di frasi di un linguaggio, si possono mettere in ordine alfabetico: dunque, è possibile enumerarli, metterli in lista. E l'idea della macchina universale di Turing, alla quale arriveremo tra poco, è semplicemente questa: che l'hardware rimane fisso e costante, mentre a cambiare è il software, cioè il particolare programma che viene scelto dalla lista.

A questo punto, Turing cominciò a porsi delle domande di tipo metamatematico. A chiedersi, per esempio: «*Siamo in grado, vedendo il programma, di capire se prima o poi si fermerà, su un dato input, oppure no?*». Ed è riformulando appunto il ragionamento del teorema di Gödel, che Turing riuscì a dimostrare che ci sono delle attività che una macchina di Turing non è in grado di fare.

Queste attività che una macchina di Turing non può compiere, sono limitazioni delle macchine, e costituiscono l'analogo delle limitazioni dei sistemi formali che Gödel aveva dimostrato esistere per la matematica. E queste limitazioni, a loro volta, costituivano l'analogo delle limitazioni della ragion pura che Kant aveva individuato nella filosofia.

Ci sono dunque tre tipi diversi di limitazioni: quelle della ragion pura di Kant, quelle dei sistemi matematici di Gödel e quelle delle macchine di Turing. Ma esse sono in realtà tenute insieme da un unico filo logico e da un’unica idea che appare In forme diverse, ma simili, nella filosofia, nella matematica e nell'informatica.

L'idea del calcolatore universale

Ciascuna delle macchine che Turing inventò, le cosiddette macchine di Turing, è in grado di risolvere un particolare algoritmo informatico, ed è in grado di simulare un particolare sistema matematico. Si potrebbe dunque obiettare che nessuna di queste macchine è effettivamente un computer, perché un computer è un sistema programmabile.

Ma qui arriviamo alla grande scoperta di Turing, nella sua tesi del 1936. Egli si accorse che è vero che ciascuna delle sue macchine era in grado di fare soltanto un c.ompito. Ma non appena una di quelle macchine raggiungeva una massa critica, cioè non appena una macchina risultava in grado di codificare e decodificare i programmi scritti, come se fossero semplicemente delle sequenze di simboli sul loro nastro di memoria e simulare le istruzioni di quei programmi, allora diventava quello che Turing chiamò una «macchina universale». In altre parole, non appena una macchina è in grado di fare un certo numero di operazioni, allora diventa in grado di farle tutte!

Questa fu veramente una grande sorpresa. Chi di voi non è più giovanissimo, ricorderà che tempo fa c'erano delle calcolatrici tascabili, in grado di fare le quattro operazioni fondamentali: somme, sottrazioni, prodotti e divisioni. Se uno aveva dei soldi in più, poteva comprare una calcolatrice un po' più grande, che oltre alle quattro operazioni magari sapeva fare le radici, gli esponenti ecc.

Con ancora più soldi si potevano fare ancora più cose, come le funzioni trigonometriche o i logaritmi, e così via. Si poteva immaginare che questo sarebbe stato un percorso senza fine: aggiungendo soldi dalla parte del compratore, e circuiti dalla parte del venditore, si potevano fare macchine sempre più grandi e potenti.

Ebbene, Turing capì che questa via era proprio quella che non bisognava seguire! Bastava semplicemente arrivare a una macchina che fosse in grado di decodificare e simulare i programmi, per arrivare a quello che era appunto una macchina universale. E i computer di oggi sono esattamente come Turing li ha immaginati: macchine che, come tutte le macchine, sanno fare soltanto un numero finito di operazioni, ma che fanno quelle giuste!

Cioè, un computer sa prendere dei programmi che gli vengono forniti dall'esterno, sa decodificarli, sa simulare le istruzioni contenute in questi programmi, e sa riprodurre i risultati di questi programmi. E allora qualunque macchina, per quanto grande e potente essa sia, può essere simulata da una piccola singola macchina, purché questa macchina sia una macchina universale.

In altre parole, Turing capì che c'era un *trade-off*: non era necessario cambiare e potenziare l'hardware di una macchina, l'hardware poteva rimanere, e si poteva invece cambiare e potenziare il software, i programmi.

E la macchina universale di Turing è proprio questo: una macchina in grado di emulare, di simulare qualunque programma che gli viene dato. Ed è esattamente così che oggi noi siamo abituati a considerare la macchina che noi abbiamo di fronte a noi sul tavolino, sulla scrivania: una macchina programmabile. Tutti i nostri computer sono dunque delle macchine universali di Turing.

Ma c'era ancora un passo da fare per arrivare alla vera macchina che noi usiamo, perché tutte queste cose Turing le aveva fatte solo sulla carta; aveva, cioè, fatto un progetto del computer, ma rimaneva ancora da realizzarlo in pratica. Come si passò dal progetto astratto alla realizzazione concreta del computer, è l'ultima parte della nostra storia.

Dall'idea alla costruzione pratica

Per realizzare praticamente, concretamente, fisicamente la macchina di carta, il progetto che Turing aveva stilato nel 1936, bisognava anzitutto incominciare a costruire una macchina che fosse in grado di eseguire le operazioni fondamentali che fanno di una macchina particolare una macchina universale. Ora quali sono queste operazioni fondamentali?

Quello che Turing scoprì, è che in realtà non sono nient'altro che le famose leggi della logica elementare, che Boole aveva scoperto a metà dell'Ottocento. Bisognava dunque realizzare una macchina in grado di eseguire le operazioni fondamentali dell'algebra booleana. Questo lo si fece negli anni Trenta e negli anni Quaranta dello scorso secolo, e il primo passo lo compì nel 1938 Claude Shannon, l'inventore della teoria dell'informazione.

Ricordiamo che Boole aveva identificato le proposizioni vere e false con i numeri 1 e 0. E aveva identificato le operazioni proposizionali elementari, come la negazione e la congiunzione, con operazioni algebriche, come la sottrazione e la moltiplicazione. Quello che Shannon intuì, è che i circuiti elettrici sono in grado di fare esattamente queste operazioni.

Per cominciare, c'è un analogo abbastanza ovvio dell'1 e dello 0, dentro un filo elettrico, ed è il passaggio o meno della corrente. Dunque, l'1 e lo 0 diventano semplicemente un filo elettrico dentro cui passa, oppure non passa, la corrente elettrica.

Quanto alle operazioni booleane sull'1 e lo 0, si possono facilmente realizzare con delle porte elettriche. Per esempio, la negazione, che fa passare dallo O all'1, e dall'1 allo O, è semplicemente una porta che fa passare la corrente quando non passa, e la ferma quando passa: una specie di interruttore, dunque, che accende ciò che è spento, e spegne ciò che è acceso.

Analogamente si può fare per la congiunzione. Ci dev'essere un interruttore, una porta che collega due fili in entrata a uno in uscita. Se entrambi i fili di entrata portano corrente, anche quello in uscita deve portarla. Se invece almeno uno dei due fili in entrata non porta corrente, nemmeno quello in uscita la porta.

Come Boole aveva tradotto nel 1854 la logica proposizionale nel linguaggio algebrico dell'algebra booleana, così Shannon tradusse nel 1938 il linguaggio algebrico nel linguaggio dei circuiti elettrici. Questo è dunque ciò che costituisce l'ossatura di una macchina di Turing universale: un circuito elettrico, con delle porte o degli interruttori.

Un nuovo passo avanti fu compiuto nel 1943 da un famoso neurofisiologo, di nome Warren McCulloch, che lavorò con un matematico di nome Walter Pitts. McCulloch voleva fare un modello del sistema nervoso centrale: cercare, cioè, di rappresentare matematicamente una versione molto semplificata di quello che succede nel nostro cervello, e nel nostro sistema nervoso.

Ora, dentro il cervello ci sono dei neuroni, che sono collegati fra loro da una rete di collegamenti nervosi. I neuroni provocano delle scariche elettriche, degli impulsi nervosi, che si trasmettono lungo i collegamenti. E come si può intuire, il modello che McCulloch e Pitts svilupparono risultò alla fine semplicemente essere una versione dei circuiti elettrici di Shannon, e dunque dell'algebra booleana.

A questo punto si capì che si erano messe le mani su qualcosa di veramente fondamentale. C'era un primo nucleo centrale della logica, scoperto dagli stoici, che permetteva di mettere insieme le proposizioni, e si chiamava appunto «calcolo proposizionale». C'era poi un secondo nucleo centrale della logica, scoperto da Aristotele, che era invece il calcolo dei sillogismi, o dei predicati a una sola variabile.

Tradotti questi due calcoli logici in linguaggio algebrico, si scoprì che i due nuclei centrali della logica erano semplicemente due versioni diverse di una stessa cosa, perché le stesse algebre di Boole li traducevano entrambi.

Poi si scoprì che queste algebre si potevano realizzare attraverso semplici circuiti elettrici, da una parte, e attraverso semplici reti neurali, dall'altra. Realizzare la scatola nera del computer, la scatola logica di comando, diventava dunque un semplice problema di ingegneria elettronica. Bastava mettere insieme dei fili in maniera tale da poter fare un numero sufficiente di operazioni logiche, che venrvano tradotte attraverso l'algebra booleana in un numero sufficiente di collegamenti elettrici.

E questo fu il punto di partenza. Naturalmente era soltanto l'involucro, il nucleo centrale del computer, e mancava ancora qualcosa, che fu aggiunto subito dopo la guerra.

La nascita dell'informatica

Abbiamo visto che finora c'è stata una confluenza di idee veramente formidabile: idee che arrivano dalla filosofia, dalla logica, dalla teoria dell'informazione e dalla neurofisiologia, si stavano tutte incontrando e interagendo, a metà del Novecento.

L'ultimo passo fu fatto da un grande matematico, uno dei più grandi del secolo scorso: un personaggio singolare, di nome John von Neumann. Un ungherese, emigrato negli anni Trenta negli Stati Uniti e diventato famosissimo come grande risolutore di problemi.

Si dice che, di fronte al suo ufficio all'*Institute for Advanced Study di Princeton,* ci fosse la coda di gente che andava a chiedergli informazioni sui problemi più disparati, perché von Neumann aveva intuizioni formidabili, anche in campi di cui non era specialista, ed era velocissimo a capire e pensare.

Si racconta, per esempio, che un giorno gli chiesero di risolvere un problema che qualcuno di voi avrà sentito. C'è una macchina, che sta andando verso una certa meta. E c'è una mosca, che sta volando avanti e indietro tra la macchina e la meta. Qual è il percorso che alla fine la mosca ha fatto, volando a una certa velocità, se la macchina si muoveva a una certa altra velocità?

C'è un semplice trucco per risolvere il problema: basta calcolare il tempo che la macchina impiega per arrivare alla meta, e dalla velocità della mosca si deduce il percorso che essa ha compiuto durante quel tempo. Poiché von Neumann diede immediatamente la risposta, gli dissero: «*Ah! Conoscevi il trucco!*». Ma lui rispose: «*Che trucco? Ho semplicemente calcolato la serie infinita dei percorsi della mosca, avanti e indietro*».

Si raccontano su di lui moltissimi altri aneddoti, ma che cosa fece veramente von Neumann per la storia dell'informatica? Nel 1945 entrò a lavorare a due prototipi di calcolatori elettronici che si stavano costruendo all'epoca: l'EDVAC e l'ENIAC. E portò con sé un enorme bagaglio tecnico, che tra le tante cose conteneva anche un bel po' di logica matematica.

Von Neumann era presente nel 1930, quando Gödel aveva fatto l'annuncio dei suoi risultati a un famoso convegno, e li capì immediatamente. Nel giro di un paio di settimane, con la sua proverbiale velocità, disse addirittura a Gödel come si potevano migliorare questi risultati.

Von Neumann aveva anche conosciuto Turing, quando questi era andato a studiare a Princeton per un paio d'anni, tra il 1936 e il 1938. Quindi von Neumann sapeva benissimo quale fosse il problema teorico che un computer doveva risolvere. Doveva essere una macchina universale, cioè essere in grado di prendere dei programmi, decodificarli, simularli e restituire gli stessi output che questi programmi avrebbero restituito se avessero eseguito questi compiti individualmente. E l'implementazione di questi aspetti fu appunto il contributo fondamentale che egli diede all'informatica, con la cosiddetta «*architettura di von Neumann*»

Il computer fu dunque realizzato mettendo insieme due ingredienti. Da una parte, la scatola nera, realizzata attraverso l'algebra di Boole in astratto, e i circuiti elettrici, o le reti neurali, in concreto. E dall'altra parte, la programmabilità, grazie all'idea ast.ratta di macchina universale di Turing, e l'implernentazione concreta dell'architettura di von Neumann.

Nel 1945 nacque così il primo vero computer universale, che fu appunto l'ENIAC: in quel momento è nata l'informatica. Quello che è successo dopo è stato semplicemente un «mettere i puntini sulle i» di questa grande impresa intellettuale, che abbiamo raccontato partendo da Leibniz e arrivando fino a oggi.

Cosa non può fare un computer

Abbiamo dunque raccontato una lunga storia. Una storia che è partita nel 1200, coi tentativi di Lullo di costruire una grande arte, la *Ars Magna*. E continuata nel 1666 col sogno di Leibniz della *Ars combinatoria*, della caratteristica universale. Ed è poi esplosa in rapida successione, nel corso di un solo secolo, tra metà Ottocento e metà Novecento, attraverso l'algebra di Boole, i teoremi di Gödel, la macchina universale di Turing, i circuiti elettrici di Shannon, le reti neurali di McCulloch e Pitts e l'architettura di von Neumann.

Dall'ENIAC a oggi sono passati più di cinquant' anni, e l'informatica è ormai fiorita, o esplosa. Ma i computer che essa offre al mercato e ai consumatori, sono tutte e sole macchine universali. I miglioramenti che ci sono stati dal 1945 a oggi, non sono teorici, ma solo tecnologici. Le macchine diventano sempre più veloci, sempre più piccole, sempre meno care.

Ma dal punto di vista delle loro potenzialità di calcolo, non hanno fatto passi avanti, perche più di quanto fa una macchina universale, non si può fare. I computer del 1945 sapevano fare tutte e sole le cose che sanno fare i nostri computer: cioè, calcolare le funzioni calcolabili. Sembra quasi un gioco di parole, ma le funzioni calcolabili oggi si possono definire all'inverso, dicendo semplicemente che sono le funzioni che uno qualunque dei nostri computer che abbiamo sulla scrivania è in grado di calcolare.

Naturalmente c'è un piccolo trucco: stiamo parlando dei computer come se fossero macchine di Turing universali, ma ricorderete che il nastro della macchina di Turing in realtà era un nastro infinito. Le nostre macchine non hanno una memoria veramente infinita: ce l'hanno «potenzialmente» infinita, perché può essere espansa quanto si vuole. Dunque, i computer sono solo macchine universali «potenziali».

Per concludere, possiamo però cercare di rispondere a un interrogativo che si pose lo stesso Turing nel 1950, pochi anni prima di morire: quali sono i limiti delle macchine universali, dei computer? Sappiamo che le macchine universali di Turing possono calcolare tutte le funzioni calcolabili, ma il nostro problema ora è: che cos'è veramente calcolabile?

Uno dei grandi problemi che Turing si pose, alla fine della sua breve vita, fu quello dell'«intelligenza artificiale». Fino a che punto è possibile simulare il pensiero umano, attraverso una macchina come il computer? Questa, naturalmente, è una domanda alla quale non si è data ancora una risposta definitiva. L'Intelligenza Artificiale è un'impresa che dura ormai anch'essa da più di cinquant'anni, e molti risultati sono stati ottenuti. Per esempio; si sono riusciti a fare dei programmi di scacchi che ormai battono sistematicamente i campioni del mondo. Quindi le macchine sono oggi in grado di giocare a scacchi meglio degli scacchisti stessi.

Dal punto di vista della matematica, le macchine ormai sono in grado di dimostrare teoremi. Anche se non hanno ancora dimostrato dei grandissimi teoremi da sole, sono state essenziali nel permettere la dimostrazione di alcuni grandi teoremi. Un esempio fra tutti è il famoso «teorema dei quattro colori»: il fatto, cioè, che bastano quattro colori per colorare le nazioni di una carta geografica, in modo tale che queste nazioni non abbiano mai lo stesso colore se sono confinanti. Il teorema fu dimostrato a metà degli anni Settanta, usando un numero enorme di ore di computer, circa 2000, a Urbana-Champaign.

C’è però una cosa che, sorprendentemente, si è dimostrata molto difficile, ed è forse impossibile, da far fare alle macchine: simulare ciò che noi facciamo con la nostra sensorialità, col nostro corpo.

Gli stoici definivano l'uomo come «un animale razionale». Quando l'Intelligenza Artificiale incominciò, si pensava che simulare le attività animali sarebbe stato molto facile, e simulare invece l'attività tipicamente umana, cioè la razionalità, sarebbe stato molto difficile.

Oggi si è scoperto, sorprendenternente appunto, che è vero esattamente il contrario. E facile per le macchine, o perlorneno è possibile, simulare la nostra parte razionale. E invece molto difficile far simulare a un computer, a una macchina, cose che noi facciamo automaticamente, e che fanno benissimo anche gli animali: per esempio, riconoscere le forme.

Qualunque animale riconosce il suo padrone, semplicemente in base a delle impressioni sensoriali. Ma le macchine ancora hanno enormi difficoltà a riconoscere le facce, a fare quella che in gergo viene chiamata *pattern recognition*. E questa è una scoperta filosoficamente interessante: le macchine sembrano essere in grado di fare bene ciò che è tipicamente umano, e male ciò che è tipicamente animale.

Si assiste così a un completo capovolgimento della filosofia cartesiana, che pensava che gli animali fossero delle pure macchine, e che l'uomo fosse qualche cosa di più. Oggi noi stiamo scoprendo che gli animali non sono affatto delle macchine, e che noi ci distinguiamo da loro solo per il fatto di avere una macchina nel cervello. In altre parole, l'informatica ci ha insegnato che l'uomo non è altro che «*un animale dotato di un computer*».