

37

---

SEMINARI  
E CONVEGNI

*Giornata di studio*  
*Pisa, Scuola Normale Superiore*  
*2 ottobre 2012*

---

# Menti e macchine

Alan Mathison Turing  
a cento anni dalla nascita

a cura di  
Hykel Hosni



EDIZIONI  
DELLA  
NORMALE

© 2014 Scuola Normale Superiore Pisa

ISBN 978-88-7642-XXX-X

# Sommario

---

Avvertenza HYKEL HOSNI	7
Nota biografica e scientifica GABRIELE LOLLI	9
Secondo solo a Newton nel Pantheon di Cambridge... GABRIELE LOLLI	59
Perenni indecisioni PIERGIORGIO ODIFREDDI	71
Alan Turing e il programma di ricerca in intelligenza artificiale ROBERTO CORDESCHI, GUGLIELMO TAMBURRINI	85
APPENDICE Si può dire che i calcolatori automatici pensano? M.H.A. Newman, A.M. Turing, G. Jefferson, e R.B. Braithwaite,	125



# Avvertenza

---





# Nota biografica e scientifica

---

## *Date principali*

- 1912 (23 giugno): Nascita, Paddington, London
- 1926-31: Sherborne School
- 1930: Morte dell'amico Christopher Morcom
- 1931-34: *Undergraduate* al King's College, Cambridge University
- 1932-35: Studi di meccanica quantistica, probabilità, logica
- 1935: Eletto *Fellow* del King's College, Cambridge
- 1936: La macchina di Turing, calcolabilità, macchina universale
- 1936-38: Princeton; algebra, teoria dei numeri, crittografia
- 1939: Ph.D. in logica, ritorno a Cambridge
- 1939-40: Bletchley Park; le *Bombe* per la decifrazione di *Enigma*
- 1939-42: Decifrazione di *Enigma* per *U-boats*, la battaglia dell'Atlantico
- 1943-45: Principale consulente anglo-americano per problemi di decifrazione; elettronica
- 1945: London, National Physical Laboratory
- 1946: Progetto di computer e software all'avanguardia mondiale
- 1947-48: Programmazione, reti neurali e intelligenza artificiale
- 1948: Manchester University
- 1949: Primi usi matematici nuovi del computer
- 1950: Il test di Turing per l'intelligenza delle macchine
- 1951: Eletto *Fellow* della Royal Society. Teoria non lineare della crescita biologica
- 1952: Condannato per omosessualità
- 1953-54: Lavoro non terminato in biologia e fisica
- 1954 (7 giugno): Morte per avvelenamento da cianuro, Wilmslow, Cheshire.

## 1. *Infanzia*

Alan Mathison Turing nacque il 23 giugno 1912 da Julius Mathison Turing, funzionario in India, e Ethel Sara Stoney. La famiglia apparteneva alla classe media superiore e i figli erano destinati alle *public school* per poter accedere all'università<sup>1</sup>.

Turing fu lasciato in Inghilterra con il fratello; affidato a una famiglia, crebbe praticamente come un orfano, pur mantenendo rapporti epistolari con la madre, e frequentò una *preparatory school*. I suoi insegnanti disperavano che fosse in grado di accedere alla *public school*, per la sua trascuratezza espositiva, i metodi poco ortodossi, l'interesse quasi esclusivo per esperimenti chimici; Turing era uno studente mediocre, ultimo in inglese, insensibile alle regole di comportamento e esposizione. Al secondo esame tuttavia, nel 1926 fu ammesso alla Sherborne School, nel Dorset.

## 2. *La scoperta della scienza*

«Se vuole diventare uno scienziato specializzato, perde il suo tempo in una *public school*», dirà il direttore della scuola. Turing era stato molto influenzato dal libro di E.T. Brewster, *Natural Wonders Every Child Should Know* (1912), che lesse nel 1922 e lo portò a un interesse esclusivo per la natura e la scienza. Il libro aveva un'impostazione laica, e si dilungava sui problemi biologici e della vita. Un esempio, che sembra significativo di quanto la sua lettura possa avere avuto un'influenza durevole, è l'accenno ivi contenuto al corpo come una macchina:

Perché naturalmente il corpo è una macchina. Una macchina estremamente complessa, molte, molte volte più complicata di qualsiasi macchina sia mai stata costruita con le mani, ma alla fine pure sempre una macchina. Il corpo è stato paragonato a una macchina a vapore, ma questo prima che si sapesse

---

<sup>1</sup> Le *public school* erano scuole private dove veniva svolta una preparazione speciale, anche sociale, in vista degli esami di ammissione ai *college* che formavano la classe dirigente in Inghilterra e Galles. L'iscrizione alle scuole stesse era subordinata a un esame. Le più famose tra esse erano un elemento di prestigio nel curriculum non inferiore a quello dell'università frequentata.

tutto quello che ora sappiamo su come funziona. Piuttosto è un motore a benzina, come quello di un'automobile, di un motoscafo, di un aereo<sup>2</sup>.

Da Brewster Turing assorbì anche l'interesse per gli esperimenti chimici, che mantenne tutta la vita. In una lettera del 1925 alla madre diceva: «Sto facendo una serie di esperimenti secondo un ordine che ho stabilito io. Quello che vorrei è fare le cose partendo da ciò che è più comune in natura e con il minimo spreco di energia». Era naturale che una tale disposizione lo portasse ad apprezzare ancor meglio la visione matematica.

Intorno al 1930 per esempio Turing fece un lavoro per il padre dell'amico Victor Beuttell, un imprenditore nel campo della illuminazione; Beuttell aveva inventato un sistema per inquadrare foto o disegni dietro una parete di vetro, e illuminarli dall'alto, ma voleva che la luce cadesse in modo uniforme su tutta l'estensione. Si trattava di calcolare la curvatura del vetro che garantisse questo risultato, cosa che fece Turing, pur non essendo capace di spiegare la formula impiegata, indicando anche come eliminare riflessioni dovute allo spessore del vetro.

Questo esempio è paradigmatico del modo di Turing di affrontare i problemi, lo vedremo ripetutamente in seguito, e mostra come fin da giovane avesse capacità eccezionali e spontanee di matematizzare fenomeni naturali non applicando formule precostituite ma inventando direttamente la soluzione del problema in termini matematici (vedremo diversi casi). Era la disperazione dei suoi insegnanti perché tendeva ad affrontare direttamente problemi avanzati (per esempio trovò da solo una rappresentazione in serie per  $\tan^{-1}$ ), ma trascurava i compiti elementari.

### 3. *Il corpo e lo spirito*

Più avanti, verso il 1930, quando già studiava la relatività e la meccanica quantistica, Turing fu molto colpito dall'interpretazione che ne dava Arthur S. Eddington (1882-1944) in *The Nature of the Physical World* (1928). Eddington era convinto che la meccanica quantistica lasciasse spazio alla volontà umana; in questo periodo Turing era ossessionato dal problema di come la mente umana fosse radicata nella materia e come potesse essere rilasciata dalla materia alla

---

<sup>2</sup> Citato da Hodges 1983, p. 13.

morte. La sua preoccupazione era legata a un fatto traumatico, la morte improvvisa di Christopher Morcom, con cui aveva fatto amicizia nel 1927: il primo legame era stato il comune interesse scientifico, ma Morcom era molto diverso da Turing, preciso e ordinato, sistematico e gentile. L'amore infelice per Chris fu probabilmente l'occasione per Turing per riconoscere la propria omosessualità. Le lettere di Turing alla madre di Chris rivelano la sua disperazione, e l'ammirazione che lo aveva spinto a imitare e adottare le migliori qualità dell'amico. Nel 1932 Turing mandò alla madre di Chris un piccolo saggio sulla natura dello spirito in cui si riconosce l'influenza delle idee di Eddington.

#### NATURA DELLO SPIRITO

In passato gli scienziati supponevano che se fosse stato possibile conoscere tutto dello stato dell'universo in un momento determinato, allora sarebbe stato possibile prevederne con esattezza anche ogni momento futuro. Questa idea trovava alimento nel grande successo delle predizioni astronomiche. Oggi, invece, la scienza moderna è giunta alla conclusione che quando ci occupiamo di atomi ed elettroni, ci è del tutto impossibile conoscerne lo stato esatto, in quanto gli strumenti che usiamo sono fatti essi stessi di atomi ed elettroni. Su piccola scala quindi, l'idea di essere in grado di conoscere l'esatto stato dell'universo è destinata inevitabilmente a fallire. Ciò significa, di conseguenza, che anche la teoria secondo la quale le nostre azioni sono predeterminate al pari delle eclissi ecc., non regge. Noi possediamo una volontà in grado di determinare l'azione degli atomi probabilmente di una piccola porzione di cervello, o forse anche di tutto il nostro cervello. Il resto del corpo agisce in modo da amplificare tale volontà. Si deve rispondere allora alla domanda su come siano regolate le azioni degli altri atomi dell'universo. Probabilmente dalla stessa legge e semplicemente dagli effetti remoti dello spirito, ma siccome esse non hanno un apparato che le amplifichi sembra che siano regolate dal puro caso. La non predestinazione apparente della fisica è quasi una combinazione di fatti casuali.

McTaggart ha mostrato che la materia è priva di significato in assenza dello spirito (in questa esposizione con materia io non intendo quella che può essere un solido un liquido o un gas quanto piuttosto quella che è studiata in fisica, per esempio la luce, e anche la gravitazione, cioè quella che forma l'universo)<sup>3</sup>. Personalmente penso che lo spirito sia davvero eternamente con-

---

<sup>3</sup> John Ellis Mc Taggart (1866-1925) è stato un filosofo neoidealista, quasi tutta la vita al Trinity College di Cambridge. Il testo letto da Turing potrebbe essere *Nature*

nesso alla materia ma certamente non sempre allo stesso genere di corpo. Ho creduto possibile che uno spirito alla morte andasse in un universo completamente separato dal nostro, ma ora penso che materia e spirito siano così legati che questa sarebbe una contraddizione in termini. Non è escluso tuttavia che un tale altro universo possa esistere.

Allora per quanto riguarda il legame reale tra spirito e corpo io ritengo che il corpo, in ragione del suo essere un corpo vivente possa ‘attirare’ e ‘trattenere’ uno spirito, finché il corpo è vivo i due sono fermamente connessi. Quando il corpo dorme non riesco a immaginare cosa succeda, ma quando il corpo muore il ‘meccanismo’ del corpo, che trattiene lo spirito, è fuori uso e lo spirito trova un nuovo corpo prima o poi e forse immediatamente.

Per quel che riguarda la domanda sul perché noi dobbiamo avere un corpo, perché non viviamo o non possiamo vivere come spiriti e comunicare in questa forma, forse probabilmente potremmo farlo, ma non ci sarebbe nulla da fare. Il corpo fornisce allo spirito qualcosa di cui interessarsi e da usare<sup>4</sup>.

Al momento del suo ingresso all’Università, nel 1930, al King’s College di Cambridge, sua seconda scelta, Turing manifestava dunque posizioni volontaristiche e spiritualistiche. L’osservazione è interessante in vista della curiosa circostanza che in seguito, quando Turing non sarà più, Gödel gli rimprovererà un errore filosofico, quello di pensare che non esista una mente separata dalla materia. Ma in effetti in seguito Turing cambierà atteggiamento proprio in relazione a questo problema.

#### 4. La laurea

I problemi matematici affrontati da Turing durante l’università e che lo portarono ai suoi primi contributi originali erano in teoria dei gruppi e in probabilità; un risultato di teoria dei gruppi, pubblicato nel 1935, aveva un’applicazione alle funzioni quasi periodiche, risolvendo un problema trovato leggendo il libro *Mathematische Grundlagen der Quantenmechanik* di John von Neumann (1903-57); questi visitò Cambridge nella primavera del 1935 e apprezzò il lavoro di Turing.

---

*of Existence*, 2 voll., Cambridge Univ. Press, 1921; trad. it. *La natura dell’esistenza*, Pitagora, Bologna, 1999.

<sup>4</sup> *Nature of Spirit*, manoscritto non datato conservato nell’archivio del King’s College.

Il secondo interesse fu stimolato da una serie di lezioni di Eddington sulla metodologia della scienza del 1933. Scontento del modo approssimativo nel quale Eddington parlava della tendenza dei dati a organizzarsi nella forma della curva a campana, Turing diede una formulazione rigorosa e una dimostrazione del teorema ~~del limite centrale~~. Questo fu l'argomento della tesi, molto apprezzata nonostante risultasse che il teorema era stato dimostrato da Jarl W. Lindeberg (1876-1932) nel 1922. L'incidente è tipico del lavoro di Turing, il quale quando trovava un problema che lo interessava non si preoccupava di considerare la bibliografia e i tentativi già fatti, ma lo affrontava di petto con i suoi strumenti, o con quelli che si costruiva apposta. L'originalità era comunque garantita, se non nell'enunciato, nel metodo seguito. La situazione si ripeterà in altre occasioni. La tesi fu peraltro valutata positivamente al punto che sulla base di essa Turing venne eletto nel 1935 *Fellow* del King's College.

Nel 1933 Turing aveva acquistato e letto *Introduction to Mathematical Philosophy* di Bertrand Russell (1872-1970), e aveva svolto un intervento al *Moral Science Club* esponendo l'idea che il punto di vista puramente logicista in matematica non fosse adeguato: pensava che esistessero svariate interpretazioni delle proposizioni matematiche, tra cui quella logicista era una soltanto di quelle possibili.

## 5. *Le macchine di Turing*

Nel 1935 Turing seguì un corso sui fondamenti della matematica tenuto dal topologo Maxwell H.A. Newman (1897-1984), dove veniva esposto il programma di David Hilbert (1862-1943) di utilizzazione della formalizzazione e della logica per le dimostrazioni di non contraddittorietà, arrivando fino ai risultati di incompletezza di Kurt Gödel (1906-78) del 1930. Tra i problemi messi a fuoco dal lavoro della scuola di Hilbert e di Gödel, restava aperto quello della decisione, o *Entscheidungsproblem*, come era stato formulato nel libro di D. Hilbert e Wilhem Ackermann (1896-1962), *Grundzüge der theoretischen Logik*, 1928: «determinare se una data formula qualsiasi del calcolo dei predicati è universalmente valida o no». Gli autori precisavano che il problema si può ritenere risolto solo se «è conosciuto un procedimento (*Verfahren*) per mezzo del quale per una data espressione logica si possa decidere la validità (o rispettivamente la soddisfacibilità) con un numero finito di operazioni».

La difficoltà del problema stava nell'indeterminatezza del concetto

di «procedimento», o sinonimi quali metodo effettivo, algoritmo, procedura, regola finita, regola meccanica, dove «meccanico» era usato in senso metaforico. Turing portò a Newmann la soluzione (negativa) dell'*Entscheidungsproblem* nell'aprile del 1936; la sua soluzione era basata su una originale definizione matematica di macchina.

Turing raccontò di aver concepito l'idea un pomeriggio mentre stava sdraiato sui prati di Grantchester (sul fiume Cam). Le macchine sono definite trasportando ad esse le competenze e le azioni di una persona che esegua un calcolo, come descritto in quello che è diventato un famoso pezzo da antologia:

Un calcolo è normalmente eseguito [da un *computer*, calcolatore umano] scrivendo certi simboli su carta. Possiamo supporre il foglio diviso in quadrati come il quaderno di aritmetica di un bambino. [...]

Supporrò che il numero dei simboli che possono essere scritti sia finito. [...]

Il comportamento del calcolatore in ogni istante è determinato dal simbolo che sta osservando e dal suo «stato della mente» in quel momento. Possiamo supporre che ci sia un limite al numero di simboli o quadrati che il calcolatore può avere sott'occhio in ogni momento. Se vuole osservarne di più, userà osservazioni successive. Supporremo anche che *il numero di stati della mente che devono essere presi in considerazione sia finito* [...], la restrizione non influisce seriamente sul calcolo, perché l'uso di stati della mente più complicati può essere evitato scrivendo più simboli sul nastro.

Ora

Immaginiamo di spezzare le operazioni compiute dal calcolatore in «operazioni semplici» che siano così elementari che non è facile immaginare di scomporle ulteriormente. Ogni operazione del genere consiste di qualche cambiamento nel sistema fisico costituito dal calcolatore e dal suo foglio. Noi conosciamo lo stato del sistema se conosciamo la successione di simboli scritti, quali di questi sono osservati dal calcolatore (eventualmente in un ordine speciale) e lo stato della mente del calcolatore. Possiamo supporre che in un'operazione semplice non più di un simbolo venga alterato. [...]

Si arriva così alla nozione di istruzione come una quintupla

$$q_i \alpha \beta M q_j$$

da intepretarsi nel seguente modo: se a un istante  $t$  la macchina si trova nello stato  $q_i$  e legge (con la sua testina di lettura e scrittura che esplora

un nastro potenzialmente infinito diviso in caselle) nella casella osservata il simbolo  $\alpha$ , allora sovrascrive il simbolo  $\beta$  ad  $\alpha$  (eventualmente  $\beta = \alpha$ ), si sposta a destra o a sinistra di una casella a seconda che  $M$  sia  $D$  o  $S$  e passa allo stato  $q_j$  (eventualmente  $q_i = q_j$ ).

Una macchina di Turing è un insieme finito e non contraddittorio di istruzioni, cioè tale che non esistano due istruzioni diverse che iniziano nello stesso modo  $q_i\alpha$ . La definizione nella sua semplicità è elegante come un *haiku*<sup>5</sup>. Ammettendo macchine che per certi argomenti non si fermano, quindi non danno un valore, e definiscono funzioni parziali, non c'è bisogno di alcuna verifica non costruttiva sulla ammissibilità di insiemi di istruzioni, ma solo quella puramente sintattica di non contraddittorietà; le macchine di Turing possono essere enumerate in modo effettivo come insiemi finiti di quintuple di un alfabeto numerabile (i simboli di lettura e scrittura possono essere assunti come finiti, non così quelli per gli stati, non ponendosi un limite ai numero di stati di una eventuale macchina), e sia  $\{M_x\}_{x \in \mathbb{N}}$  una fissata enumerazione. Che l'enumerazione sia effettiva significa che una macchina è in grado di associare a ogni  $x$  le istruzioni di  $M_x$ .

Nelle esposizioni si usa ora fare riferimento alla formulazione posteriore di Stephen C. Kleene (1909-94), dove le macchine sono usate per calcolare i valori di funzioni dai numeri naturali nei numeri naturali, mentre Turing aveva presentato le macchine come finalizzate a calcolare la rappresentazione decimale infinita di un numero reale. Si scrive  $M_i(\sigma) \downarrow$  per dire che la macchina  $M_i$  applicata all'argomento  $\sigma$  si ferma dopo un numero finito di passi, e  $M_i(\sigma) \uparrow$  per dire che non si ferma mai.

Una funzione numerica  $f$  è calcolabile da una macchina  $M$  se, per ogni  $n \in \mathbb{N}$  scritto con opportuna rappresentazione  $\underline{n}$  sul nastro iniziale, si ha che  $M(\underline{n}) \downarrow$  se e solo se  $n \in \text{dom}(f)$  e il valore  $M(\underline{n})$  presente sul nastro finale è la rappresentazione di  $f(n)$ .

Le funzioni calcolabili quando si trattano problemi matematici o logici sono anche dette ricorsive, e ricorsivi sono detti gli insiemi decidibili, o quelli la cui funzione caratteristica è ricorsiva.

La dimostrazione di Turing consiste nel mostrare anzitutto l'esistenza di problemi, relativi alle macchine stesse, a cui non può essere data risposta con una macchina. Il problema effettivamente insolubile classico, nella formulazione di Martin Davis (1928-) del 1958 è quello

---

<sup>5</sup> Giudizio di M. Saler, *The Times Literary Supplement*, n. 5725/5726, 21-28 dicembre 2012, p. 31.



dell'*halt*, o dell'arresto, il problema di stabilire se una macchina qualunque messa in funzione su un argomento qualunque si ferma dopo un numero finito di passi o no<sup>6</sup>.

La tecnica è quella di Gödel, nel teorema di incompletezza del 1930, della diagonalizzazione con autoriferimento e colpo di coda, *with a twist*: se  $\varphi_x$  è la funzione calcolata da  $M_x$ , si osserva che la funzione

$$\xi(x) = \begin{cases} 0 & \text{se } \varphi_x \text{ calcolata da } M_x \text{ per } \underline{x} \text{ dà } \varphi_{\underline{x}(\underline{x})} = 1 \\ 1 & \text{altrimenti,} \end{cases}$$

non è calcolabile da una macchina di Turing. Se  $\xi$  fosse una  $\varphi_r$ , si vedrebbe subito una contraddizione per  $\varphi_r(r)$ .

Ma la funzione  $\xi$  sarebbe calcolabile se esistesse una macchina  $H$  tale che per ogni  $i, x \in \mathbb{N}$ :

$$H(\underline{i}, \underline{x}) = \begin{cases} 0 & \text{se } M_i(\underline{x}) \downarrow \\ 1 & \text{se } M_i(\underline{x}) \uparrow . \end{cases}$$

Infatti per calcolare  $\xi$ , dato  $x$  si calcolerebbe  $H(\underline{x}, \underline{x})$ , e se la risposta fosse 0 si passerebbe a calcolare  $U(M_x, \underline{x})$  dove  $U$  è una macchina universale (si veda oltre), cioè a calcolare  $\varphi_x(x)$ ; si farebbe quindi un test sul risultato di questo calcolo, e se esso fosse 1 si darebbe a  $\xi(x)$  il valore 0, non fosse 1, o non fosse un numero, si darebbe a  $\xi(x)$  il valore 1; se invece la risposta  $H(M_x, \underline{x})$  fosse 1, si porrebbe direttamente  $\xi(x) = 1$ .

Quindi non esiste una macchina di Turing come  $H$ , cioè si dice che il problema dell'arresto è effettivamente insolubile (non risolubile con una macchina), o indecidibile. La macchina universale è una macchina  $U$  tale che se sul suo nastro sono scritti una opportuna codifica delle istruzioni di una macchina  $M$  e un argomento  $\sigma$  nel linguaggio della macchina  $M$ , allora la macchina calcola,  $U(M, \sigma)$ , lo stesso risultato che otterrebbe  $M$  applicata a quell'argomento,  $M(\sigma)$ .

La macchina universale serve a Turing come lemma tecnico per dimostrare l'esistenza di problemi non meccanicamente risolubili, cioè non risolubili mediante una delle sue macchine. Infatti si noti che tut-

---

<sup>6</sup> Nel lavoro originario Turing dimostrava che è effettivamente insolubile il problema di stabilire se una macchina è *circle-free* o no, cioè scrive infinite cifre oppure ne scrive solo un numero finito pur continuando a lavorare all'infinito e ripetendo le stesse cifre.

ti i calcoli sopra ipotizzati dovrebbero essere svolti sul proprio nastro della macchina universale, quelli di  $H$ , i test sui valori, la scrittura di  $M$  che è possibile perché l'enumerazione è effettiva, fatta cioè da una macchina.

Il problema dell'arresto si può ridurre al problema della validità logica, nel senso che a ogni macchina  $M$  e argomento  $\sigma$  si associa in modo effettivo una formula  $A_{M,\sigma}$  in modo che  $M(\sigma) \downarrow$  se e solo se  $A_{M,\sigma}$  è logicamente valida.

Dunque se il problema della validità logica fosse decidibile, anche il problema della fermata lo sarebbe<sup>7</sup>.

Non è questo tuttavia l'interesse del risultato di Turing, quanto la definizione delle macchine e soprattutto della macchina universale.

Nell'aprile del 1936, quando ancora Newman non aveva finito di esaminare il manoscritto di Turing, arrivò da Princeton (USA) la notizia che Alonzo Church (1903-95) aveva preceduto Turing dimostrando l'esistenza di problemi aritmetici indecidibili, derivandone la soluzione negativa anche per l'*Entscheidungsproblem* della logica. Si dà per scontato che Turing sia stato preceduto da Church, anche se le date non sono del tutto chiare. L'applicazione di Church al problema della decisione arrivò alla redazione del neonato *Journal of Symbolic Logic*, diretto da Church stesso, il 16 aprile, con correzioni nell'agosto; l'articolo di Church con un problema indecidibile è leggermente precedente, sempre del 1936<sup>8</sup>. Purtroppo Turing non ha lasciato nessuna indicazione dello sviluppo e dei tempi del suo lavoro per poter fare un confronto.

Ad ogni modo la questione della priorità è secondaria rispetto al fatto che le due soluzioni sono basate su una definizione completamente diversa di funzione effettivamente calcolabile. Church aveva definito un formalismo per trattare funzioni e regole di calcolo orientate alla trasformazione di espressioni in una forma detta normale (il  $\lambda$ -calcolo). Le funzioni definibili nel  $\lambda$ -calcolo coincidono con le funzioni calcolabili mediante macchine di Turing (quest'ultimo termine introdotto proprio da Church nella recensione di Turing)<sup>9</sup>.

La dimostrazione della equivalenza delle due definizioni fu aggiunta da Turing in appendice al suo lavoro. Newman non ebbe difficoltà a

<sup>7</sup> Per maggiori dettagli si veda il contributo di P. Odifreddi in questo volume.

<sup>8</sup> A. Church, "An Unsolvble Problem of Elementary Number Theory", *The American Journal of Mathematics*, 58/2, 1936, pp. 345-63.

<sup>9</sup> Coincidono anche con le funzioni ricorsive generali secondo la definizione di Gödel del 1934, già dimostrate equivalenti alle  $\lambda$ -definibili da Kleene nel 1935.

convincere gli editori dei *Proceedings* della London Mathematical Society dell'opportunità di pubblicare comunque il contributo di Turing, per l'originalità dei suoi metodi e della sua impostazione. Il lavoro di Turing, con il titolo "On computable numbers, with an application to the *Entscheidungsproblem*", uscì in due parti nel 1936, ma in seguito alla revisione dell'ordine di numerazione dei volumi della London Mathematical Society fu ristampato intero nel volume del 1937 (di qui una certa ambiguità, e l'imprecisione da parte di chi vi si riferisce come lavoro del 1936-37).

## 6. Princeton

Turing si recò a Princeton, all'Institute for Advanced Study, nel 1936, per due motivi, per interagire con Church e perché interessato alla prospettiva di frequentare von Neumann. Non vinse la borsa Procter di Princeton, nella domanda per la quale aveva indicato come campi di interesse la teoria dei gruppi e la probabilità, ma poté inizialmente sostenersi con la retribuzione del King's College. Nel primo periodo scrisse un articolo sul  $\lambda$ -calcolo e due di teoria dei gruppi, risolvendo in particolare un problema postogli da von Neumann, anche se alla madre (lettera del 3 novembre 1936) dichiarava di non prendere questi argomenti sul serio come la logica. Non avendo ottenuto come desiderava un incarico di insegnamento a Cambridge, dove aveva trascorso l'estate del 1937, facendo la conoscenza nell'occasione di Ludwig Wittgenstein (1889-1951), decise di restare ancora un anno a Princeton e di scrivere una tesi di dottorato. Una lettera di raccomandazione di von Neumann per la borsa Procter menzionava due campi di lavoro nei quali, secondo von Neumann, Turing aveva dimostrato di poter dare contributi importanti, la teoria delle funzioni quasi periodiche e la teoria dei gruppi continui. Non citava il lavoro sui numeri calcolabili. Qualche anno dopo tuttavia dimostrerà di conoscerlo bene.

Un altro argomento a cui Turing iniziò a dedicarsi a Princeton fu quello degli zeri della funzione zeta di Riemann<sup>10</sup>; l'interesse partiva

---

<sup>10</sup> La funzione  $\zeta(s) = \sum_1^\infty \frac{1}{n^s}$ ,  $s$  numero complesso con  $Re(s) < 1$  notata per la prima volta da Euler, ha una grande importanza nella teoria analitica dei numeri; i suoi zeri, cioè i punti dove si annulla, a parte i numeri interi pari negativi, sono distribuiti simmetricamente intorno alla retta con  $Re(s) = 1/2$ . L'ipotesi di Riemann è che essi cadano tutti su detta retta.

dalla distribuzione dei numeri primi, e dalla questione della sua stima mediante la funzione logaritmica; questa sembrava sempre sovrastimare il numero di primi, ma John E. Littlewood (1885-1977) aveva provato che esisteva un punto molto grande dove la sottostimava; Stanley Skewes (1899-1988) nel 1933 era riuscito a dare una stima del punto dove cambiava l'approssimazione, assumendo l'ipotesi di Riemann. Turing inventò un nuovo modo per calcolare gli zeri della funzione zeta di Riemann; si interessò anche al loro calcolo effettivo (nel 1939 a Cambridge sembra che abbia perfino progettato una macchina elettrica per approssimare una serie di Fourier per la funzione zeta) e suggerì a E.C. Titchmarsh, che prima aveva usato un sistema di calcolo con schede perforate adattato dai calcoli astronomici, l'uso di una macchina analogica utilizzata a Liverpool per il calcolo delle maree.

Infine Turing iniziò a esplicitare il suo interesse per la crittografia. In una lettera alla madre della fine del 1936, per spiegarle possibili applicazioni della matematica astratta, le parlava di un lavoro che lo stava occupando al momento: «Risponde alla questione “Quale è il tipo più generale di codice o cifrario possibile”, e allo stesso tempo (piuttosto naturalmente) mi permette di costruire un sacco di codici particolari e interessanti. Uno di essi è praticamente impossibile da decodificare senza avere la chiave, ed è molto veloce da codificare. Mi aspetto che potrei venderli al Governo di Sua Maestà per una somma piuttosto sostanziale, ma ho dei dubbi sulla moralità di questi comportamenti. Cosa ne pensi?».

Verso la fine del 1937 era impegnato con la tesi, ma trovò tempo da dedicare alla criptanalisi discutendone con un fisico canadese, Malcolm MacPhail. Una sua idea era quella di moltiplicare i numeri che codificavano i messaggi per un numero spaventosamente grande che richiedesse ai tedeschi (i tradizionali nemici ora sempre più minacciosi) 100 anni di lavoro al calcolatore da tavola per otto ore al giorno per scoprire il fattore segreto. Forse l'idea era ingenua, ma di fatto Turing progettò anche una moltiplicatrice in binario formata da relé elettromagnetici costruendone i primi stadi relativi ai connettivi logici per vedere se e come funzionava; e funzionava. Può darsi che cominciasse a pensare di realizzare una macchina di Turing con una rete di interruttori.

Risulta chiaro tuttavia che egli era preoccupato dai preparativi di guerra tedeschi, e forse già pensava a dare il suo contributo nel campo della crittografia.

## 7. La tesi di dottorato

La tesi di dottorato sulle logiche ordinali riguardava un argomento suggerito da Church, per ridurre le conseguenze dell'incompletezza messa in luce da Gödel. L'incompletezza si esprime dicendo che nessuna procedura effettiva può dimostrare tutte le verità aritmetiche. Quindi è impossibile esprimere tutta la matematica in un sistema unico. Tuttavia i matematici sembrano avere la capacità di decidere tutti gli enunciati matematici.

Dal punto di vista formale, si possono aggiungere assiomi a un sistema di aritmetica in modo che diventino dimostrabili enunciati che sono «veri ma indimostrabili» nel sistema dato. Ma si ripropone la stessa situazione e nuovi enunciati risultano indecidibili nel sistema allargato. Il lavoro di Turing riguardava la possibilità di ottenere la completezza aritmetica non con un solo sistema, né ovviamente con un numero finito di estensioni, ma con una successione infinita non costruttiva di sistemi.

Una delle proposte sviluppate da Turing realizzava almeno un livello di completezza, quello per le formule  $\Pi_1^0$  cioè della forma  $\forall \vec{x} A(\vec{x})$  dove  $A$  rappresenta una relazione ricorsiva. Questi enunciati sono quelli che sono tipicamente prodotti dal metodo di Gödel, ma il risultato parve deludente a Turing. Il lavoro fu ripreso e ripulito nel 1958 da Solomon Feferman<sup>11</sup>.

Turing propone una definizione ristretta e una più ampia di logiche basate su ordinali. Quella ristretta consiste nell'estendere un sistema con tutte le formule  $Prov(\ulcorner \varphi \urcorner) \rightarrow \varphi$ <sup>12</sup>, dove  $Prov$  è il predicato «essere dimostrabile» nel sistema dato, e nell'iterare la procedura [questi enunciati esprimono in un certo senso la correttezza del calcolo, e implicano la non contraddittorietà del sistema]. Per assicurare l'enumerabilità ricorsiva di tutti i teoremi di ogni estensione, sono usati ordinali ricorsivi per descrivere l'iterazione<sup>13</sup>. [...] Feferman ha precisato questi risultati dando un metodo non ambiguo per costruire tali predicati di dimostrabilità e ha dimostrato in modo rigoroso che se si inizia l'iterazione con l'aritmetica classica  $\mathbf{Z}$  la classe di teoremi è precisamente quella ottenuta aggiungendo a  $\mathbf{Z}$  tutte le formule  $\Pi_1^0$  vere [nella struttura  $\mathbf{N}$ ]. Di conseguenza, (i) il predicato di dimostrabilità per l'intera iterazione è arit-

<sup>11</sup> S. Feferman, "Ordinal logics re-examined" e "On the strength of ordinal logics", *Journal of Symbolic Logic*, 23, 1958, pp. 105-6.

<sup>12</sup> [ $\ulcorner \varphi \urcorner$  è il codice numerico della formula  $\varphi$ ].**lasciare quadre?** testo, ma aggiunte da Lolli

<sup>13</sup> [Gli ordinali ricorsivi sono quelli, numerabili, che sono associati a una definizione di un buon ordine ricorsivo dei numeri naturali].

si, sono note non presenti nel

meticamente definibile, mentre il predicato «essere una dimostrazione» non lo è, (ii) le stesse funzioni ricorsive sono dimostrabili nella logica ordinale come in  $Z$  [...].

La definizione più ampia associa semplicemente un sistema formale a ogni ordinale ricorsivo (per esempio aggiungendo per ogni ordinale ricorsivo il principio di induzione transfinita relativo). Come è stato dimostrato da Wang, Shoenfied e [Kreisel], ogni formula aritmetica *verun* questi sistemi. [...] Turing mirava alla completezza [...]. I risultati citati sopra mostrano in modo conclusivo che sulla base della definizione ristretta non si ottiene la completezza neanche per formule con due quantificatori  $(\forall x \rightarrow \exists y)$  davanti a una relazione ricorsiva primitiva, e che in base a quella più ampia la si ottiene troppo facilmente<sup>14</sup>.

La nozione di dimostrazione non era decidibile in questi sistemi. L'ostacolo tecnico era quello di decidere se una formula era una formula ordinale, e la decisione non poteva essere meccanica, pena la riproposizione dell'incompletezza.

Turing introdusse in questo lavoro l'idea di una macchina di Turing con oracolo; in esse è permesso di interrompere un calcolo e interrogare l'oracolo chiedendo se un determinato argomento appartiene o no a un insieme non decidibile (nel caso specifico, chiedendo se una formula fosse una formula ordinale o no). Il concetto di macchina con oracolo avrebbe avuto grandi sviluppi nella teoria della calcolabilità relativa: se un insieme è decidibile da una macchina con oracolo per l'insieme  $A$ , l'insieme è ricorsivo in  $A$ , o relativamente ad  $A$ .

A parte le questioni tecniche, il lavoro è importante per capire l'atteggiamento di Turing in questa fase rispetto ai problemi della mente. Appare chiaro l'interesse di Turing ad approfondire un problema complementare rispetto a quello affrontato nel 1936, cioè quello delle azioni della mente matematica che non consistono nel seguire un metodo definito, e che si potrebbero chiamare creative o originali (almeno rispetto alla decisione sulla verità degli enunciati indecidibili di Gödel).

In un inciso che non sembra essenziale per lo sviluppo logico e potrebbe benissimo essere saltato Turing include una discussione di intuizione [*intuition*] e ingegnosità [*ingenuity*] che sembra proprio una

---

<sup>14</sup> G. Kreisel, "Ordinal Logics and the Characterization of Informal Concept of Proof", *Proceed. Intern. Math. Congress*, Cambridge University Press, Cambridge 1958, pp. 289-99.

dichiarazione programmatica del suo voler allargare la propria analisi della mente.

Il ragionamento matematico può essere considerato piuttosto schematicamente come l'esercizio di una combinazione di due facoltà, che possiamo chiamare *intuizione e ingegnosità*.<sup>(a)</sup>

L'attività dell'intuizione consiste nel proporre giudizi spontanei che non sono il risultato di percorsi consapevoli del pensiero [...]

L'intuizione, intesa come espressione di un «giudizio spontaneo che non è il risultato di percorsi consapevoli del pensiero», interviene nel riconoscimento della verità degli enunciati di Gödel, ma può anche intervenire nell'affermazione di verità non ancora dimostrate ma che sono dimostrabili (per esempio, prima che fosse dimostrata, la fattorizzazione unica dei numeri naturali in numeri primi). La dimostrazione riduce il ricorso all'intuizione ma non la elimina del tutto, solo la confina in passi più facilmente accettabili e meno soggetti a critiche. I passaggi intuitivi vanno riconosciuti, distinti, e ridotti al minimo, ma comunque sono ineliminabili.

L'esercizio dell'ingegnosità in matematica «consiste nell'aiutare l'intuizione mediante opportune disposizioni delle proposizioni, ed eventualmente figure o disegni. Si intende che quando questi sono davvero ben organizzati la validità dei passi intuitivi non può essere messa in dubbio».

Le parti giocate da queste due facoltà variano da caso a caso e da matematico a matematico. Tale arbitrarietà può essere rimossa con l'introduzione di una logica formale. La necessità di usare l'intuizione è allora grandemente ridotta stabilendo regole formali per eseguire inferenze che sono sempre intuitivamente valide. Quando si lavora con una logica formale, l'idea dell'ingegnosità prende una forma più definita. In generale, una logica formale sarà formulata in modo da permettere una notevole varietà di possibili passi a ogni stadio della dimostrazione, L'ingegnosità allora determina quali passi sono più opportuni allo scopo di dimostrare una particolare proposizione.

Prima di Gödel qualcuno pensava che sarebbe stato possibile svilup-

<sup>(a)</sup> Lasciamo da parte la facoltà più importante, che è quella di saper distinguere gli argomenti interessanti dagli altri; in effetti, stiamo considerando la funzione del matematico come se fosse semplicemente quella di stabilire la verità o falsità di proposizioni.

pare il programma di formalizzare il ragionamento matematico a un punto tale che tutti i giudizi intuitivi sarebbero stati sostituiti da un numero finito di regole. «La necessità dell'intuizione sarebbe quindi stata del tutto eliminata»<sup>15</sup>.

Non essendosi ciò dimostrato possibile, si pone un obiettivo minore: «L'introduzione della logica ordinale ha lo scopo di evitare il più possibile gli effetti del teorema di Gödel», o almeno isolare dove si manifesta l'intuizione.

Vista l'impossibilità di trovare una logica formale che elimini *in toto* la necessità di ricorrere all'intuizione non possiamo che rivolgerci a sistemi logici «non costruttivi», nei quali non tutti i passaggi di una dimostrazione sono di tipo meccanico, essendo alcuno di tipo intuitivo. [Un esempio è fornito dalle sue logiche ordinali]. Quali proprietà dovrà avere una logica non costruttiva se vogliamo usarla per l'espressione di dimostrazioni matematiche? Dovrà essere in grado di lasciare capire molto chiaramente quando un passaggio è un passaggio intuitivo e viceversa quando è di tipo esclusivamente formale. L'intuizione dovrebbe comunque essere ridotta al minimo. Ma ciò che più conta è che non vi deve essere alcun dubbio sul fatto che tale logica porta a risultati corretti laddove i passaggi di carattere intuitivo sono corretti.

Sembra chiaro che in questo periodo Turing non pensava che la mente fosse una macchina: se la mente matematica fosse una macchina, allora esisterebbero problemi assolutamente indecidibili (quelli formulabili con il metodo di Gödel applicato alla macchina che è la mente); è interessante confrontare la posizione di Turing con quella di poco posteriore di Gödel<sup>16</sup>, che dall'ipotesi che la mente matematica

---

<sup>15</sup> Segue una frase che a una prima lettura alcuni giudicano ambigua, addirittura contraddittoria: «Nella nostra discussione, tuttavia, siamo andati all'estremo opposto e abbiamo eliminato non l'intuizione, ma l'ingegnosità, e questo nonostante la nostra ricerca mirasse in gran parte allo stesso scopo. Abbiamo cercato di vedere fino a che punto è possibile eliminare l'intuizione, e lasciare solo ingegnosità». Ma non c'è contraddizione. La prima frase allude alle ricerche sistematiche, basate sull'elencazione di tutte le possibili derivazioni, dove non si manifesta alcuna ingegnosità. L'ultima frase spiega come l'obiettivo fosse comune, perché anche Turing vuole capire «fino a che punto è possibile eliminare l'intuizione».

<sup>16</sup> Nella Josiah Willard Gibbs Lecture tenuta nel 1951 alla riunione della AMS alla Brown University, "Some basic theorems on the foundations of mathematics and their implications", pubblicata nelle opere complete di K. Gödel, *Collected Works. Vol. III*,



sia una macchina, e quindi esistano problemi assolutamente indecidibili approda, rifiutando tale eventualità per una disposizione di ottimismo razionalistico, a una posizione di realismo.

Anche Turing non credeva che esistessero problemi assolutamente indecidibili, ma non per un motivo filosofico; la riflessione di Turing non coinvolgeva le tradizionali categorie dell'idealismo e del realismo, si basava piuttosto sull'attenzione a quello che fanno i matematici, che quando sono bloccati cambiano i metodi.

Già nel 1936 aveva osservato che se anche un numero  $\delta$  non è calcolabile «è possibile (per quanto ne sappiamo al presente) che un qualunque numero prefissato di cifre di  $\delta$  possa essere calcolato, ma non con un processo uniforme. Quando sono state calcolate un buon numero di cifre di  $\delta$ , è necessario un metodo essenzialmente nuovo per ottenerne altre».

Nel 1940 scriverà a Newman:

I risultati diretti di insolubilità o incompletezza dei sistemi di logica ammontano a questo:

$\alpha$  Non ci si può aspettare di risolvere l'*Entscheidungsproblem* per il sistema in questione.

$\beta$  Non ci si può aspettare che il sistema copra tutti i metodi di prova.

Noi costruiamo le dimostrazioni [...] proponendone una e verificando se è giusta [...] Se si considera  $\beta$  si deve ammettere che non uno ma molti metodi di verifica sono necessari. Scrivendo sulle logiche ordinali avevo questo in mente.

In un'altra lettera rimproverava Newman di essere più hilbertiano di lui: «Pensi che ci sia o debba esserci, ma nessuno l'ha fatta, una macchina fisica e tutto l'armamentario [*outfit*] formale che riguarda la macchina?». Allora «c'è poco da aggiungere: dobbiamo semplicemente abituarci a usare questa tecnica e rassegnarci al fatto che ci sono problemi che non avranno mai risposta».

Se pensi a differenti macchine non vedo la tua difficoltà. Si immagina che macchine differenti forniscano diversi insiemi di dimostrazioni e scegliendo una macchina opportuna si può approssimare «verità» con «dimostrabilità» meglio che con una meno adatta, e in un certo senso si può approssimare tanto bene quanto si vuole. La scelta di una macchina richiede intuizione [...]

oppure in alternativa si può cercare direttamente la dimostrazione, e questo di nuovo richiede intuizione.

Nel 1938 il ruolo dell'intuizione è affermato senza discussioni, anche se non è data alcuna indicazione sulle possibili origini della stessa: «Non mi addenterò nel tentativo di fornire una spiegazione più chiara di questo concetto di 'intuizione'». In seguito Turing non tornerà più sul problema dell'intuizione, non userà più questa parola, e quando discuterà della possibile intelligenza delle macchine userà come vedremo concetti diversi e una nuova prospettiva.

## 8. *La guerra, Bletchley Park*

Nel 1938 von Neumann gli offrì un posto di assistente a Princeton, ma Turing preferì tornare a Cambridge, anche senza un incarico di insegnamento. Nella primavera del 1939 tenne comunque un corso libero sui fondamenti della matematica; seguì anche un ciclo di lezioni di Wittgenstein, nella cui trascrizione si trova un famoso scambio tra i due<sup>17</sup>.

Wittgenstein parlava del paradosso del Mentitore, all'interno della sua discussione sulla inspiegabile importanza data dai matematici e filosofi al problema della non contraddittorietà, e Turing osservò acutamente che di solito si usa una contraddizione come criterio per stabilire che qualcosa è andato storto; ma in questo caso non si trova nulla di sbagliato. Wittgenstein tuttavia insisteva sulla tesi che gli stava a cuore: e allora che male ne può venire dalla contraddizione? e Turing: «Non risulterà un vero pericolo, a meno che non ci sia un'applicazione nella quale un ponte potrebbe crollare, o qualcosa del genere». Ma sembra un dialogo tra sordi. Turing trovava Wittgenstein «a *very peculiar man*»: dopo che una volta avevano discusso di logica, Wittgenstein aveva detto che doveva ritirarsi in una stanza vicina per pensare a quello che era stato detto.

Turing aveva probabilmente fatto conoscere attraverso i canali del King's College la sua disponibilità a lavorare sui problemi di crittografia e nel settembre del 1939 iniziò a lavorare a Bletchley Park, dove era

---

<sup>17</sup> C. Diamond (ed.), *Wittgenstein's Lectures on the Foundations of Mathematics*. Cambridge 1939, Harvester Press, Hassocks, 1976, lectures 21 e 22; trad. it. *Lezioni di Wittgenstein sui fondamenti della matematica*, Boringhieri, Torino, 1982.

sistemata la *Gouvernement Code and Cipher School*, insieme a scienziati di varie aree, linguisti, esperti di scacchi o di parole incrociate.

Il primo problema a cui Turing scelse di dedicarsi, potendo manifestare la sua libertà perché il problema era ritenuto insolubile, fu quello della decifrazione dei messaggi criptati dai tedeschi con le macchine *Enigma*, relativi alle operazioni navali nel Nord Atlantico. Il metodo era quello tradizionale della traslazione delle lettere, ma realizzato con notevoli complicazioni. La macchina originale aveva tre rotori azionati elettricamente posizionati ogni giorno su lettere iniziali diverse e diversamente combinati, per un totale di  $6 \times 26^3 = 6 \times 17576$  possibili stati della macchina. Ogni rotore scattava di un passo quando il precedente aveva completato un giro. Negli usi militari si aggiungeva talvolta un mescolamento iniziale e finale delle lettere dell'alfabeto. I messaggi contenevano una parte iniziale che spiegava a disposizione dell'apparato, per la decodificazione. Nessun avanzamento sarebbe stato possibile senza accadimenti casuali che succedono in guerra, come errori di codifica che costringevano a inviare due volte un messaggio o riconoscimento di una parola grazie al carattere ripetitivo dei messaggi di argomento militare.

Il controspionaggio polacco aveva costruito macchine, chiamate *Bombe*<sup>18</sup>, per riconoscere il settaggio dei rotori in assenza del mescolamento preventivo e finale. A Bletchley Park vennero costruite su disegno di Turing *Bombe* più perfezionate di quelle polacche. Inoltre, nel caso di presenza della complicazione ulteriore del mescolamento, negli usi militari di *Enigma*, Turing propose una ricerca basata sullo sviluppo delle conseguenze di un'ipotesi di corrispondenza attraverso una successione di implicazioni che portavano a una contraddizione o a un risultato di consistenza<sup>19</sup>.

Nell'agosto 1940 le nuove *Bombe* lavoravano con successo nell'impostazione data loro da Turing<sup>20</sup>.

Il metodo di Turing era un esempio di ricerca delle probabilità inverse, cioè la valutazione della probabile causa di un effetto, invece del probabile effetto di una causa. Ogni esito di un esperimento aumenta

---

<sup>18</sup> In polacco *Bomba*, pl. *Bomby*, non è chiaro se dal rumore che si udivano o dalla forma di un gelato. Le *Bombe* simulavano il lavoro di diverse copie di *Enigma* coordinate.

<sup>19</sup> Gordon Welchman contribuì a un miglioramento essenziale.

<sup>20</sup> Per i dettagli tecnici si rinvia alla biografia di Hodges citata in bibliografia, capp. *The Relay Race* e *Bridge Passage*, pp. 160-255.

la verosimiglianza di un'ipotesi. C.S. Peirce (1839-1914) nel 1878 aveva chiamato peso di evidenza il (logaritmo del) valore da aggiungere a un'ipotesi in seguito a un esperimento riuscito.

Turing usava una interpretazione in termini di scommesse e lavorava con poste invece che con probabilità. Sia  $p \in [0; 1]$  la probabilità di un evento di interesse  $H$ . Si può dire che determina la posta  $p/(1-p)$  in favore di  $H$ ,  $o(H)$ , o per *odds*.

Sotto questa interpretazione poste di 1 («evens») corrispondono a probabilità 1/2, poste di 2 a probabilità 2/3, e così via. Se denotiamo con  $p(H | E)$  la probabilità dell'ipotesi  $H$  data l'evidenza  $E$ , allora

$$\frac{o(H | E)}{o(H)} = \frac{p(H | E)}{p(H | \bar{E})}$$

dove  $E^c$  è il complemento di  $E$ .

A proposito della parte sinistra dell'equazione, I. J. (Jack) Good nota, in una presentazione del lavoro di Turing fatta anni dopo, che è «naturale chiamarlo il fattore a favore di  $H$  ammesso  $E$ , e questo era il nome datogli da Turing in una cruciale applicazione criptanalitica durante la seconda guerra mondiale nel 1941»<sup>21</sup>.

Good definì poi un peso di evidenza come il logaritmo dell'espressione di sinistra:

$$W(H : E) = \log \frac{o(H | E)}{o(H)}.$$

Turing fu il primo a riconoscere l'opportunità di dare un nome all'unità di misura dei pesi di evidenza. «Il metodo venne chiamato *Banburismus* perché i pesi di evidenza erano registrati su fogli di carta stampata nella città di Banbury; allora Turing propose il nome 'ban' per l'unità di peso di evidenza quando la base del logaritmo è 10. [...] Un decimo di un ban era chiamato deciban in analogia con il decibel in acustica, e usavamo l'abbreviazione db» (Good, 1985).

Un ban di evidenza è qualcosa che rende un'ipotesi dieci volte più probabile di quello che era prima. La misura serve a giudicare il valore di un esperimento dall'ammontare di peso di evidenza che esso in media

---

<sup>21</sup> I.J. Good, "Weight of Evidence: A Brief Survey", in J.M. Bernardo, M.H. de Groot, D.V. Lindely and A.F.M. Smith (eds), *Bayesian Statistics 2*, Elsevier, 1985. I.J. Good (1916-2009) era un giovane assistente di Turing a Bletchley Park.

produce. Turing introdusse anche la varianza del peso di evidenza prodotta da un esperimento, una misura di quanto imprevedibile esso fosse.

Queste idee venivano usate per valutare la probabilità che un messaggio fosse trasmesso due volte, o che la stessa configurazione fosse stata usata più volte o che un particolare rotore fosse in una posizione particolare.

Inoltre Turing organizzò anche gli esperimenti nel modo che anticipava l'analisi sequenziale: invece di decidere in partenza quanti esperimenti fare, impostarli e continuare a farli finché non si raggiungesse un obiettivo fissato di peso di evidenza.

Turing non conosceva sicuramente le riflessioni di Peirce, forse invece quelle di Thomas Bayes (1702-61), certo l'equazione sopra riportata è una versione in termini di scommesse del teorema di Bayes. A questo proposito Good commenta che: «Turing non citava il teorema di Bayes, con cui [l'equazione] è naturalmente strettamente legata, perché preferiva sempre sviluppare da solo ogni argomento [come già abbiamo detto]. Quando gli feci notare che il concetto era essenzialmente un'applicazione del teorema di Bayes, mi rispose "Può darsi"».

Nei lunghi periodi di relax a Bletchley Park Turing non spreca il tempo: diceva che fino ad allora la logica era stata il suo mestiere e la crittografia il suo hobby, ora la crittografia era il suo mestiere e la logica l'hobby. Scrisse alcuni lavori di logica, in particolare sulla teoria dei tipi, stimolato da Newman, anch'egli arruolato a Bletchley Park.

Turing intanto sviluppava in varie direzioni la curiosità su quello che potevano fare le macchine. Con Jack Good giocava a scacchi e a Go, e discuteva della possibilità di usare macchine per decidere le mosse degli scacchi. Una macchina secondo Turing poteva simulare la scelta della prossima mossa secondo il criterio minimax: la meno peggiore, ovvero quella alla quale l'opponente è costretto a scegliere la meno vantaggiosa per lui. La decisione si può fare esplorando l'albero delle mosse a una profondità fissata, con i mezzi allora disponibili solo tre per essere fattibile, e assegnando punteggi di valutazione ai nodi delle configurazioni finali per tornare indietro valutando i nodi intermedi fino al punto di partenza.

Turing era un personaggio rispettato e ammirato, come prova il soprannome attribuitogli di «Maths Brain», nonostante le sue stranezze (la giacca del pigiama sotto la giacca sportiva, una maschera antigas per difendersi dal polline, una bicicletta che solo lui poteva usare perché aveva un difetto alla catena che solo Turing aveva capito come evitare, un modo peculiare di parlare, un po' in falsetto, interrotto da serie di ah-ah-ah... quando cercava una parola che non veniva).

Peter Hilton ha raccontato un aneddoto significativo per il carattere di Turing; tutti coloro che lavoravano a Bletchley Park dovettero a un certo punto sottomettersi allo svolgimento di preparazione militare; Turing era interessato a imparare a sparare e volentieri si iscrisse ai corsi di tiro. Il modulo da compilare comprendeva una domanda, che chiedeva se l'interessato fosse consapevole che nell'isciversi diventava sottomesso alla legge militare, con i suoi obblighi conseguenti. Turing pensò bene di rispondere di no, e nessuno se ne accorse. Dopo aver imparato a sparare molto bene, Turing perse interesse, e non partecipò più alle varie attività, che includevano marce ed esercitazioni. Fu naturalmente chiamato a rapporto e gli si contestò di dover obbedire agli ordini avendo accettato di essere soggetto alle leggi militari; Turing invitò a consultare il modulo dove aveva scritto la sua risposta, e le autorità militari non poterono fare altro che eliminarlo dai loro ranghi.

Turing si fidanzò in quel periodo con Joan Clarke, giovane matematica che dichiarava di non essere preoccupata delle pulsioni omosessuali confessate da Turing; questi tuttavia in seguito concluse di non poter seriamente coinvolgere Joan in una vita comune.

Nel 1941 la Gran Bretagna fu per un breve periodo di nuovo padrona dei mari grazie alla capacità di decodificare i messaggi tedeschi, con la possibilità di scegliere le rotte migliori nell'Atlantico e anche di attaccare gli *U-boat*. Il momento favorevole non durò molto, perché i tedeschi introdussero piccole ma impegnative modifiche, per esempio nel modo di indicare le posizioni degli obiettivi. Nel febbraio del 1942 cambiarono *Enigma* aggiungendo un nuovo rotore; per quanto gli inglesi sapessero come era settato, grazie a disattenzioni dei codificatori tedeschi, il carico computazionale diventava troppo pesante, non bastavano più reti di relé elettromagnetici e ci si cominciava a orientare sull'uso di valvole elettroniche; queste aumentavano la velocità dell'ordine di migliaia di volte, ma erano all'inizio molto fragili. Sul momento, i progressi in campo ingegneristico sembravano deludenti.

Nel corso del 1942 Turing si trovò a essere meno occupato, mentre altri, in particolare Newman, affrontavano un nuovo problema, un sistema di codifica diverso, non basato su *Enigma*, e chiamato *Fish*. Anche l'organizzazione generale del lavoro era diventata più burocratica e altre persone con migliori doti organizzative sostituivano di fatto l'autorità morale di Turing, che pure restava immutata in campo scientifico.

A questa seconda fase di lavoro crittografico, con l'attacco a *Fisch*, Turing contribuì solo con la teoria, perfezionando l'impostazione statistica che aveva elaborato nel 1940-41 e che era chiamata *Turingismus* dai giovani Peter Hilton e Donald Michie.

## 9. L'America e la corsa al calcolatore


Alla fine del 1942 Turing fu inviato negli Stati Uniti a fare da legame e mediatore con gli sviluppi del lavoro crittografico che anche gli americani avevano intrapreso in forze. A parte il lavoro diplomatico, per cui riferiva direttamente a Churchill, Turing sfruttò la permanenza in America frequentando i Bell Laboratories di New York, informandosi sul lavoro crittografico, e dando anche contributi relativamente alle ricerche sul linguaggio parlato. Si familiarizzò con tutti gli apparati elettronici usati, acquisendo in particolare notevoli conoscenze su oscilloscopi e analizzatori di frequenza. Fece la conoscenza di Claude Shannon (1916-2001) il creatore della teoria dell'informazione<sup>22</sup>. La misura di informazione di Shannon aveva analogie con il ban di Turing: un bit di informazione era qualcosa che raddoppiava la precisione. Shannon, che aveva precisato nella tesi di laurea il legame tra i circuiti elettrici con interruttori e le algebre di Boole, aveva un forte interesse per la problematica del cervello umano, per il quale aveva anche compiuto studi di neurologia. Turing ne fu certamente influenzato.


Si racconta un aneddoto relativo a una uscita ad alta voce di Turing alla mensa dei laboratori, dove erano presenti molti dipendenti allibiti: «No, non sono interessato a costruire un cervello potente. Mi basta un cervello mediocre, come per esempio quello del Direttore della ATTC [American Telephone and Telegraph Company]».

Nel marzo del 1943 fu pubblicato il lavoro di McCulloch e Pitts che con le reti neurali stabilivano un legame tra macchine logiche e la fisiologia del cervello<sup>23</sup>. McCulloch dichiarò nel 1948, nella discussione susseguente a una conferenza di von Neumann di presentazione della sua teoria degli automi, che la loro ispirazione era stata il lavoro di Turing "On Computable Numbers".

Turing tornò in Europa nel marzo del 1943, quando da poco i tedeschi si erano arresi a Stalingrado, iniziava la costruzione dei laboratori di Los Alamos per la bomba atomica, e quando la battaglia dell'Atlantico si avviava alla conclusione, concludendosi praticamente nell'estate, con il contributo decisivo del lavoro di decifrazione dei messaggi di

---

<sup>22</sup> C.E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, 27, luglio e ottobre 1948, inserito in C.E. Shannon e W. Weaver, *The*  *Mathematical Theory of Communication*, Wiley, New York, 1949; trad. it. *La teoria matematica delle comunicazioni*, Etas Kompass, Milano, 1971.

<sup>23</sup> W.S. McCulloch e W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bull. Math. Biophys.*, 5, 1943, pp. 115-3 

*Enigma* che dopo alti e bassi aveva assunto una piega del tutto favorevole agli alleati. Gli errori dei tedeschi furono molteplici: dal ritorno all'uso di tre rotori, o di quattro con il quarto in posizione fissa, alla produzione di messaggi identici sia con *Enigma* sia con altri sistemi facilmente penetrabili.

A Bletchley Park continuava il lavoro coordinato da Newman sul codice *Fish*. Una prima macchina in parte elettromagnetica e in parte elettronica venne sostituita da *Colossus*, interamente elettronica, con 1600 valvole; la realizzazione fu resa possibile da un'idea di T. H. Flowers di eliminare uno di due nastri, contenente dati da messaggi *Fish*, che dovevano essere sincronizzati in modo delicato, memorizzando i dati internamente in modo elettronico. *Colossus* fu costruito alla fine del 1943 in meno di undici mesi, dimostrando che un simile ammasso di valvole poteva lavorare in modo affidabile.

Turing pur al corrente del lavoro non vi partecipò, se non con la sua teoria statistica per l'utilizzazione pratica di *Colossus*; non si sa molto di quest'anno della vita di Turing, che era spesso cercato per consulenze ma non aveva un compito definito. L'andamento della guerra ormai richiedeva sempre meno il lavoro del controspionaggio. Turing finì per dedicarsi a un altro progetto che lo interessava, un sintetizzatore automatico di voce, *Delilah*, parzialmente analogico nella trasmissione delle altezze; per questo lavoro si trasferì presso il centro militare di Honslope Park collaborando con il giovane ingegnere Donald Bailey. Turing approfondì così molto le sue conoscenze elettroniche.

L'esperienza di questi anni con vari tipi di macchine e la padronanza dell'elettronica spinsero infine Turing a prendere in considerazione la possibilità di realizzare una versione pratica della macchina universale del 1936. Discuteva con Bailey la possibilità di realizzare con valvole il nastro della macchina; gli confermava la sua intenzione di «costruire un cervello»; la madre ricorda di averlo sentito parlare nel 1944 dell'intenzione di costruire un computer universale e del servizio che tale macchina avrebbe potuto rendere alla psicologia nello studio del cervello.

La progettazione di macchine calcolatrici sempre più potenti era all'ordine del giorno. Si sono andati poi a cercare i predecessori, e uno lo si è trovato in Charles Babbage (1791-1871) e la sua Macchina Analitica del 1837. Questa, meccanica, riceveva le istruzioni su schede perforate ispirate dai telai; la caratteristica che permetteva al suo inventore di pensarla capace di svolgere tutte le operazioni matematiche era la capacità di muoversi avanti e indietro tra le istruzioni, saltandone o ripetendone alcune quante volte era necessario secondo criteri che la



macchina stessa decideva a seconda del progresso dei calcoli: antesignano del *conditional branching*.

L'*Analytical Engine* non ebbe tuttavia influenza sulle successive realizzazioni di macchine calcolatrici come per esempio quelle di K. Zuse in Germania, che ne progettò varie versioni (che lavoravano in binario) dal 1937 alla fine della guerra, prima meccanica poi elettronica e magnetica.

Negli Stati Uniti, la sinergia tra imprese e università produsse un fiorire di iniziative, come la collaborazione tra Harvard e IBM per un progetto di H. Aiken, o le ricerche alla Moore School dell'Università di Pennsylvania.

Il gruppo dell'Università della Pennsylvania diretto dagli ingegneri elettronici J. Presper Eckert, Jr. e da John W. Mauchly aveva già costruito il calcolatore tradizionale ENIAC, costituito da oltre 19.000 valvole (anche perché lavorava con la rappresentazione decimale dei numeri); rispetto ai progetti di Zuse o Aiken aveva incorporata la possibilità di una specie di *conditional branching*. Eckert e Mauchly si orientarono nel successivo progetto dell'EDVAC ad utilizzare per la memoria le linee di ritardo acustiche e a sperimentare la memorizzazione di alcuni programmi. Nel 1944 si univa al gruppo von Neumann, che cercava la disponibilità di grande potenza di calcolo per progetti militari, come lo studio delle onde d'urto e in generale i calcoli per la bomba atomica.

John von Neumann è un personaggio che svolge in America un ruolo analogo a quello svolto da Turing in Gran Bretagna; anche egli veniva dalla logica, da Göttingen, dove negli anni Venti aveva lavorato con Hilbert, e aveva dato contributi fondamentali in particolare alla teoria degli insiemi; aveva abbandonato la logica dopo i risultati di Gödel, convinto che non ci fosse più nulla di importante da fare: von Neumann era un matematico universale, con interessi pratici. In seguito, oltre che per gli scopi militari, il calcolatore gli interessò per metterlo alla prova soprattutto su problemi complessi, sulle equazioni differenziali non lineari alle derivate parziali, problemi di dinamica dei fluidi, di meteorologia, di fisica nucleare. Il primo problema su cui lavorò il calcolatore costruito dall'Institute di Princeton su progetto di von Neumann, nel 1946, fu un problema di fisica nucleare per i laboratori di Los Alamos, che lo tenne impegnato per sessanta giorni ininterrotti.

Nel 1944 von Neumann scrisse un rapporto di presentazione dell'EDVAC, che Herman H. Goldstine distribuì e rese pubblico col solo nome di von Neumann, provocando qualche risentimento e compromettendo così tra l'altro le possibilità di brevetto.

Il salto decisivo del *Draft Report on the EDVAC* di von Neumann era la introduzione dell'*internal program storage*, un'unica memoria con-

tenente dati e programmi entrambi manipolabili. Eckert e Mauchly sosterranno di averla anticipata loro, ma pare che si trattasse solo di alcune realizzazioni specifiche. Nel rapporto invece l'idea si presentava come caratteristica strutturale della macchina.

Per il disegno logico dei circuiti von Neumann si basava sulle reti di McCulloch e Pitts; pur lavorando con altri formalismi, von Neumann riconosceva esplicitamente con i suoi collaboratori che la concezione fondamentale era nello scritto di Turing del 1936. Lo testimonia il suo collaboratore S. Frankel, che lavorava a Los Alamos: «Nel 1943 o 44 von Neumann era ben consapevole dell'importanza fondamentale dell'articolo di Turing del 1936 [...] mi indicò quell'articolo e sotto la sua insistenza lo lessi con attenzione [...] egli sottolineava decisamente a me, e sono sicuro anche ad altri, che la concezione fondamentale era dovuta a Turing, per tutto quello che non era stato anticipato da Babbage [...]».

Nel giugno del 1945 il direttore della Divisione Matematica del National Physical Laboratory J. R. Womersley ritornò da un viaggio negli Stati Uniti con una copia del rapporto sull'EDVAC; Womersley conosceva il lavoro di Turing del 1936, e probabilmente anche la sua attività a Bletchley Park; dall'America scrisse alla moglie che gli sembrava di aver visto «Turing in hardware». Turing venne invitato e accettò di associarsi al NPL per costruire un calcolatore inglese.

## 10. *Il calcolatore di Turing*

Il *Proposal* per lo sviluppo di un *Automatic Computing Engine* (ACE) fu presentato da Turing nel marzo del 1946. Esso non venne mai del tutto realizzato (a parte una variante in seguito) perché ritenuto troppo ambizioso.

La macchina era concepita fin dall'inizio come una macchina universale, per la quale l'aritmetica rappresentava solo una delle sue applicazioni ed era accompagnata da una teoria della programmazione in cui le istruzioni potevano essere manipolate come i dati.

Le macchine calcolatrici nel passato sono state progettate per eseguire in maniera accurata e moderatamente veloce piccole parti di calcoli che ricorrono di frequente. [...] Il calcolatore [*calculator*] dovrebbe essere qualcosa di diverso, in quanto sarà in grado di affrontare e risolvere problemi nella loro interezza. Invece di servirsi ripetutamente il lavoro umano per estrarre materiale dalla macchina e reinserirlo al momento opportuno, di tutto questo si prenderà cura la macchina stessa. [...]

Secondo le intenzioni del progetto, il lavoro di impostazione della macchina per affrontare nuovi problemi sarà praticamente solo una questione di carta e matita [*paper work*]. Oltre al lavoro cartaceo niente altro dovrà essere fatto se non preparare un pacco di schede di Hollerith<sup>24</sup>, secondo quanto deciso sulla carta, e passarle attraverso un lettore di schede collegato alla macchina. Non dovremo assolutamente fare alcuna alterazione interna, anche se passeremo di punto in bianco dal calcolo dei livelli di energia dell'atomo del neon alla enumerazione dei gruppi di ordine 720. Può sembrare un po' sorprendente che questo si possa realizzare. Come ci si può aspettare che una macchina sola compia tutta questa molteplice varietà di lavori? La risposta è che noi dovremmo considerare la macchina come capace di fare qualcosa di molto semplice, vale a dire eseguire un insieme di ordini dati in una forma standardizzata che essa è in grado di capire.


La composizione della macchina prevedeva: una memoria cancellabile di grande capacità (memoria dinamica); varie unità di memoria di veloce riferimento temporaneo; organi di input e output; il controllo logico, «il vero cuore della macchina» che aveva lo scopo di interpretare le istruzioni e di dar loro esecuzione; la parte aritmetica centrale, che eseguiva le operazioni aritmetiche fondamentali e varie altre operazioni di copiatura, sostituzione e simili; vari «alberi»<sup>25</sup> per la selezione dell'informazione richiesta in ogni momento tra l'unità di controllo e quella aritmetica; un orologio, per la sincronizzazione di tutta la macchina; un sistema di controllo della temperatura per le linee di ritardo; convertitori dal binario al decimale e viceversa; un apparecchio per la messa in moto; fornitura di energia.

Questi aspetti sono analizzati in dettaglio nel progetto. Dopo di che, nel paragrafo 8 sono discussi i problemi che possono essere affrontati dalla macchina, con precisa attenzione ai limiti delle risorse previste.

Sono i problemi che possono essere risolti da lavoro umano subordinato, seguendo regole fisse, e senza bisogno di comprensione, purché siano soddisfatte le seguenti condizioni:

(a) L'ammontare di appunti scritti che devono essere tenuti a disposizione ad ogni stadio sia limitato all'equivalente di 5.000 numeri reali, approssimativamente, cioè all'incirca a quello che può comodamente essere scritto su 50 fogli di carta.

---

<sup>24</sup> Schede perforate 

<sup>25</sup> Termine dell'elettronica.

(b) L'operatore umano, nel fare i suoi conti aritmetici senza aiuto meccanico, non necessita più di centomila volte il tempo disponibile sul calcolatore, cifra che rappresenta il rapporto tra la velocità di esecuzione dei due metodi.

(c) Sia possibile descrivere le istruzioni date all'operatore in linguaggio ordinario nello spazio di un ordinario romanzo. Queste istruzioni non saranno proprio le stesse che vengono normalmente date a un calcolatore umano, e che tengono conto della sua intelligenza. Le istruzioni devono coprire ogni eventualità.

Seguono esempi di problemi che soddisfano queste condizioni e di problemi che non le soddisfano, con commenti e spiegazioni. Tra i primi sono elencati:

1. costruzione di tavole di tiro, che coinvolge il calcolo delle traiettorie per piccoli archi per differenti elevazioni del quadrante e velocità di propulsione allo sparo;

2. trovare la distribuzione di potenziale all'esterno di un conduttore carico della forma di un cubo (usando un metodo di approssimazioni successive, e il metodo di rilassamento: siccome in questo metodo ogni passo principale dipende dai risultati dei passi precedenti, lo si è sempre considerato inadatto alla meccanizzazione, ma il controllo logico del calcolatore è stato concepito proprio per affrontare questi problemi);

3. soluzione di sistemi di equazioni lineari (se si ha una capacità di memoria di 6400 numeri non ci si può aspettare di risolvere sistemi con più di 50 incognite, ma la maggioranza dei problemi hanno matrici degeneri – per esempio il precedente, se trattato non nel continuo ma nel discreto);

4. calcolare la radiazione dall'estremità aperta di una guida d'onda rettangolare;

5. eseguire il prodotto di due matrici di ordine minore di 30 i cui coefficienti sono polinomi di grado minore di 10;

6. calcolare la risposta di un complicato circuito elettrico, note le caratteristiche delle sue componenti.

Non si può invece integrare un'area sotto una curva, per la difficoltà di presentare i dati d'ingresso adatti; si potrebbe calcolare il numero di combattenti da smobilitare nel giugno 1946, a partire dai dati contenuti nei registri dell'esercito, ma sarebbe uno spreco perché la velocità della macchina sarebbe molto superiore a quella della lettura delle schede, e si sprecherebbero le potenzialità del calcolatore; sarebbe meglio farlo con le comuni macchine di Hollerith.

Un problema che il calcolatore può risolvere è quello di trovare le soluzioni di un *puzzle* a incastro: «questo particolare problema non ha

grande importanza, ma è rappresentativo di una classe molto ampia di problemi non numerici che possono essere trattati dal calcolatore. Alcuni hanno importanza militare, altri sono di enorme interesse per i matematici». Infine

*Problema 10* – Data una posizione in una partita a scacchi la macchina potrebbe fare un elenco di tutte le «combinazioni vincenti» fino a una profondità di tre mosse per ogni giocatore. Questo problema non è dissimile dal precedente, ma fa sorgere la domanda: «Può la macchina giocare a scacchi?» Sarebbe molto semplice far sì che la macchina giocasse partite piuttosto brutte. Brutte perché gli scacchi richiedono intelligenza. Abbiamo detto all'inizio di questo paragrafo che la macchina dovrebbe essere trattata come se fosse del tutto priva di intelligenza. Ci sono indicazioni, tuttavia, del fatto che è possibile far sì che la macchina manifesti intelligenza, al prezzo di farle correre il rischio di commettere ogni tanto gravi errori [*occasional serious mistakes*]. Seguendo queste indicazioni si potrebbero probabilmente ottenere dalla macchina prestazioni piuttosto buone nel gioco.

Si vede a cosa Turing già stava pensando, e che presto renderà più esplicito. Pare che addirittura fin dal 1941, in un manoscritto fatto circolare a Bletchley Park e non conservato, Turing parlasse di intelligenza delle macchine che si manifestava nella soluzione di problemi attraverso ricerche guidate nello spazio delle soluzioni.

Intanto fece un'escursione nell'analisi numerica studiando l'accumulo degli errori; lo statistico H. Hotelling aveva dato una valutazione pessimistica sulla crescita degli errori nella soluzione di sistemi di equazioni (o nell'inversione di una matrice), a ogni eliminazione di un'equazione; tale valutazione sembrava mettere a repentaglio l'utilità dell'uso anche di macchine come ACE. Turing prima partecipò a prove fatte con diciotto equazioni, quindi, constatata sperimentalmente la grandezza trascurabile dell'errore finale iniziò a studiare teoricamente il problema della propagazione degli errori; il lavoro fu completato alla fine del 1947 e, pubblicato nel 1948, viene considerato la fondazione dell'analisi numerica moderna<sup>26</sup>. Turing dovette chiedere al NPL il permesso per la pubblicazione.

Come impegno principale, mentre si susseguivano varie versioni del progetto di ACE, e poche decisioni operative, Turing si dedicò intensamente alla parte di programmazione.

---

<sup>26</sup> A.M. Turing, "Rounding-off errors in matrix processes", *Quarterly Journal of Mechanics and Applied Mathematics*, 1, 1948.

## 11. *La scienza della programmazione*

La caratteristica distintiva di ACE era la minimalità dell'hardware e la prevalenza del software, dovuta alla chiara ispirazione alla macchina universale.

Alcuni anni fa [scriveva nel febbraio 1947] io riflettevo su quella che potrebbe ora essere chiamata una ricerca sulle possibilità teoriche e sui limiti delle macchine calcolatrici numeriche. Considerai un tipo di macchina che aveva un meccanismo centrale, e una memoria infinita che era contenuta su un nastro infinito. Questo tipo di macchina sembrava sufficientemente generale. Una delle mie conclusioni fu che i concetti di processo secondo «regole empiriche» [*rule of thumb*] e di «processo meccanico» erano sinonimi<sup>27</sup>.

Secondo Turing bisognava sfruttare a fondo la proprietà dell'universalità; egli considerava addirittura preferibile non avere unità aritmetiche, o di traduzione dalla rappresentazione binaria alla decimale e viceversa. Quando fosse necessaria un'operazione aritmetica veniva richiesto l'intervento di una «tavola sussidiaria» (programma). «Dobbiamo solo pensare una volta a come [qualcosa] può essere fatta, e dimenticarci di come viene fatta». BURY e UNBURY erano i comandi per chiamare e abbandonare le subroutine.

Si aveva così una gerarchia di tavole, o programmi. Turing immaginava anche di scrivere le istruzioni in una forma abbreviata, espandendo ogni tavola quando fosse necessario con un comando (a sua volta tavola di istruzioni) che chiamava EXPAND; se le istruzioni non fossero state usate più volte e se l'intera memoria fosse stata occupata da istruzioni, la macchina avrebbe lavorato solo sedici secondi. Sottolineò l'importanza dei cicli di esecuzione di istruzioni fino a che una misura opportuna prende un determinato valore: «È come un aereo che gira su un aeroporto chiedendo il permesso dopo ogni giro. L'idea è molto semplice, ma della massima importanza».

Con l'insieme delle sue soluzioni Turing mise le basi della programmazione. In linea di principio «si può comunicare con le macchine in un qualunque linguaggio purché sia esatto [...] si dovrebbe poter

---

<sup>27</sup> Con «regole empiriche» Turing si riferiva a processi che non sono espressi da leggi: un «processo secondo regole empiriche» è da intendersi come un processo che può essere svolto seguendo alla lettera una lista di istruzioni che dicono cosa fare caso per caso.

comunicare in qualunque logica simbolica, purché alla macchina siano date istruzioni che le permettano di interpretare il sistema logico». Nello stesso tempo Turing sperimentava la soluzione programmata di vari problemi dalla risoluzione di equazioni al calcolo delle matrici, trovando algoritmi originali<sup>28</sup>.

Una caratteristica di ACE a prima vista sorprendente è l'apparente mancanza del *conditional branching*, perché il controllo logico può contenere solo un indirizzo per volta, e non più di uno tra cui decidere dove andare. Ma il *conditional branching* IF... THEN... ELSE... è realizzato direttamente non con l'hardware ma con una istruzione apposita, del tipo per esempio:

$$D \times (50) + (1 - D) \times (30),$$

se IF dipende dall'essere D uguale a 0 o a 1 e 50 e 30 contengono numeri che sono istruzioni. Questo particolare è di grande importanza perché mette in piena evidenza la possibilità del programma di modificare le sue proprie istruzioni.

Negli altri progetti l'idea dello *stored program* era riconosciuta solo come una semplificazione nella preparazione della macchina; von Neumann stesso, pur consapevole della possibilità di far modificare le istruzioni, preferiva pensare che fosse limitata al minimo (solo nel 1947 la prenderà seriamente in considerazione). Invece per Turing si trattava di una mossa decisiva: «La macchina ha la possibilità di costruirsi i suoi stessi ordini [...] Questo può essere molto potente».

Turing immaginava anche le trasformazioni sociali connesse all'uso delle nuove macchine, in particolare le ripercussioni sulla matematica<sup>29</sup>. Nasceranno nuove figure di matematici e nuovi compiti. I calcolatori si troveranno ad affrontare problemi prima intrattabili e «per presentare questi problemi alla macchina ci sarà bisogno di un gran numero di matematici di grande abilità». Questi dovranno fare ricerche preventive sui problemi per metterli nella forma adatta alla computazione.

Ci sarà bisogno di un gran numero di analisti; questi dovranno per esempio stimare in modo preciso, con disuguaglianze, la bontà di una

---

<sup>28</sup> A lui si deve per esempio la decomposizione LU di una matrice in due matrici triangolari (Lower-Upper).

<sup>29</sup> Nella Conferenza del febbraio 1947 alla London Mathematical Society.

risposta invece di affidarsi al buon senso; la stima dell'errore potrà essere sostituita da una stima statistica ripetendo il lavoro più volte.

I matematici che lavoreranno con ACE si potranno dividere in signori e servi. I signori scriveranno le tavole di istruzioni, i servi le presenteranno alla macchina, faranno le necessarie correzioni, raccoglieranno dati. «Col tempo, il calcolatore stesso assumerà le funzioni sia dei signori che dei servi. I servi saranno rimpiazzati da apparecchiature meccaniche o elettriche, per esempio plotter per presentare graficamente una curva invece di individuarla con miriadi di punti. Anche i signori saranno sostituiti man mano che le tecniche diventano stereotipe e si potranno scrivere istruzioni per cui il calcolatore applichi direttamente le tecniche. In questa prospettiva, torna la domanda fino a che punto una macchina può simulare attività umane».

Turing impostava la questione ammettendo che l'intenzione primaria nel costruire queste macchine era di trattarle come schiavi, ma non necessariamente devono essere sempre trattate in questo modo. Si può supporre che una macchina abbia certe tavole di istruzioni iniziali concepite in modo che queste tavole possano, all'occorrenza, per buoni motivi, modificare se stesse.

Si può immaginare che dopo che la macchina ha lavorato per un po' di tempo le istruzioni si siano alterate al di là di ogni possibilità di riconoscimento, ma tuttavia siano ancora tali che si debba ammettere che la macchina sta ancora facendo calcoli molto interessanti e validi. Potrebbe anche stare ancora ottenendo risultati del tipo desiderato all'inizio, ma in modo molto più efficiente. In tal caso si dovrebbe ammettere che il progresso della macchina non era stato previsto quando le erano state fornite le istruzioni iniziali. Sarebbe come un allievo che avesse imparato molto dal suo maestro, ma che avesse aggiunto molto di più di suo. Se questo si verifica, la mia sensazione è che si sia obbligati a considerare la macchina come capace di mostrare intelligenza.

Probabilmente la memoria umana ha una capacità di diecimila milioni di cifre binarie, tuttavia molte destinate per esempio alle immagini visive. Dovrebbe bastare una capacità di alcuni milioni di cifre per compiere esperimenti, in settori limitati come il gioco degli scacchi. Turing era informato del fatto che Shannon aveva vinto alcune partite con una macchina (anche se non è nota l'abilità dell'avversario). «Non considererei tali vittorie molto significative. *Ciò che vogliamo è una macchina che possa imparare dall'esperienza*<sup>30</sup>. La possibilità di

---

<sup>30</sup> Corsivo nostro.



permettere alla macchina di alterare le proprie istruzioni fornisce un meccanismo in questa direzione, ma naturalmente non ci fa fare molti passi avanti».

Consapevole che parlare di intelligenza delle macchine sembra una contraddizione, visti gli usi comuni del linguaggio, Turing osservava che le macchine a cui si pensa di solito non avevano sufficiente memoria per possedere capacità di discernimento.

L'argomento potrebbe tuttavia essere affrontato in modo più coraggioso. Per esempio è stato mostrato che con certi sistemi logici non ci può essere nessuna macchina che distingua le formule dimostrabili del sistema da quelle non dimostrabili [...] Così se una macchina è costruita con questo obiettivo deve in certi casi non riuscire a dare una risposta. D'altra parte se un matematico fosse messo di fronte a un tale problema egli si guarderebbe intorno e cercherebbe e troverebbe nuovi metodi di prova, in modo che alla fine dovrebbe raggiungere una decisione relativamente alla data formula. [...] A questo argomento dunque opporrei che la macchina deve essere trattata in modo equo e leale.

Invece di lasciare che qualche volta non sia in grado di dare una risposta, potremmo disporre le cose in modo che ogni tanto essa dia risposte sbagliate. «Il matematico umano ugualmente prenderebbe qualche cantonata quando sperimentasse nuove tecniche. È facile per noi considerare queste cantonate come non rilevanti e dargli un'altra possibilità, ma alla macchina non viene riservata alcuna pietà. In altre parole, se si aspetta che la macchina sia infallibile, non può anche essere intelligente».

Per continuare il mio appello al '*fair play* per le macchine' nel verificare il loro quoziente di intelligenza, voglio osservare che un matematico umano è sempre stato sottoposto a un addestramento prolungato. L'effetto dell'addestramento può essere considerato non dissimile dall'inserire tavole di istruzioni dentro a una macchina. Non ci si deve perciò aspettare che una macchina faccia molto nella direzione di costruire tavole di istruzioni sue proprie. Nessun uomo aggiunge molto al corpo generale delle conoscenze, perché dovremmo aspettarci di più da una macchina? Detto in altro modo, alla macchina deve essere permesso di avere contatti con esseri umani affinché possa adattarsi ai loro criteri. Il gioco degli scacchi può forse essere particolarmente adatto a questo scopo, dal momento che le mosse dell'avversario della macchina forniranno automaticamente il contatto desiderato.

Queste dichiarazioni definivano l'agenda di Turing per l'immediato futuro.

Una visita negli Stati Uniti alla fine del 1946 non apportò novità tecniche; Turing ritornò convinto della superiorità di ACE sui progetti americani, a parte qualche caratteristica del calcolatore di Princeton; la molteplicità delle iniziative gli sembrava uno spreco, e riteneva più produttivo concentrarsi su un obiettivo unico. Apprezzava invece il fatto che in America ingegneri e matematici lavorassero nello stesso edificio.

## 12. *L'educazione delle macchine*

Dopo aver inutilmente cercato di far costruire al NPL un moderno centro computazionale, Turing nel settembre del 1947 tornò a Cambridge per godere di un sabbatico dal NPL. Si allenava per la maratona, ottenendo tempi quasi meritevoli dell'Olimpiade. Partecipò anche a numerose gare su varie distanze. Continuava tuttavia a lavorare sulla programmazione, in particolare su quelle che chiamava «abbreviated code instructions».

Nel 1948 scrisse una relazione sulle architetture e il problema dell'intelligenza delle macchine per il NPL. L'abstract recitava:

Sono presi in considerazione e discussi i possibili modi per far manifestare alle macchine un comportamento intelligente. Come guida si usa l'analogia con il cervello umano. Si mette in evidenza che le potenzialità dell'intelligenza umana possono essere realizzate solo se viene fornita un'adeguata educazione. L'indagine si concentra soprattutto su un analogo processo di insegnamento applicato alle macchine. Si definisce l'idea di una macchina non organizzata e si suggerisce che la corteccia cerebrale del neonato sia di questa natura. Sono presentati semplici esempi di tali macchine, ed è discussa la loro educazione per mezzo di premi e punizioni. In un caso il processo educativo è portato avanti finché l'organizzazione diventa simile a quella di un ACE.

L'evoluzione del pensiero di Turing sulla mente termina con questo lavoro: dall'idea giovanile che lo spirito comandi determinati neuroni alla considerazione dei comportamenti meccanici della mente umana calcolante, nel 1936, alla riflessione sulla presenza dell'intuizione non meccanizzabile del 1939, fino alla dichiarazione di fiducia nella possibilità di educare le macchine a mostrare intelligenza, senza più distinguere tra momenti meccanici e momenti di intuizione.

Una novità importante del lavoro del 1948 era la descrizione di un'architettura che anticipava quella del connessionismo: si trattava di macchine «costituite da un gran numero  $N$  di unità simili. Ciascuna unità ha due terminali di ingresso e un terminale di uscita che può essere connesso ai terminali di ingresso di (o o più) altre unità. Tutte le unità sono connesse a una unità centrale di sincronizzazione da cui sono emessi impulsi di sincronizzazione a intervalli più o meno uguali di tempo. Gli istanti in cui questi impulsi arrivano saranno chiamati 'momenti'. Ad ogni momento ogni unità può essere in uno di due stati. Gli stati possono essere chiamati 1 e 0. Lo stato è determinato dalla regola secondo cui gli stati delle unità da cui provengono gli stimoli di ingresso devono essere presi al momento precedente».

Le macchine non organizzate sono interessanti per il motivo che sono forse il più semplice modello di un sistema nervoso con una distribuzione casuale di neuroni. Turing ne descrive due tipi, quello più generale di tipo A e uno più particolare di tipo B. Queste gli sembrano realizzare un apprendimento di competenze simile a quello umano artigianale, quando le persone imparano a svolgere un compito senza essere in grado di enunciare le regole soggiacenti.

Una proprietà molto generale delle macchine di tipo B è che «con opportune condizioni iniziali esse faranno qualsiasi lavoro, dato un tempo sufficiente e purché il numero delle unità sia adeguato. In particolare con una macchina non organizzata di tipo B con sufficienti unità si possono trovare condizioni iniziali che la renderanno una macchina universale con una capacità di memoria fissata»<sup>31</sup>.

Il saggio codificava poi quelli che saranno in effetti i primi e principali campi di studio dell'Intelligenza Artificiale<sup>32</sup>:

Ci proponiamo [...] di cercare di vedere cosa possa essere fatto con un «cervello» che sia più o meno senza un corpo, provvisto al massimo di organi di vista, parola e udito. Dobbiamo affrontare allora il problema di trovare campi adatti del pensiero in cui la macchina possa esercitare i suoi poteri. I seguenti argomenti mi sembrano presentare vantaggi:

- (i) Diversi giochi, come scacchi, vari tipi di filetto, bridge, poker
- (ii) L'apprendimento delle lingue
- (iii) La traduzione delle lingue

---

<sup>31</sup> Si è notato che l'affermazione è errata, ma rimediabile con ragionevoli modifiche.

<sup>32</sup> Sull'Intelligenza Artificiale si veda il contributo di R. Cordeschi e G. Tamburrini in questo volume.

(iv) La crittografia

(v) La matematica.

Di questi, (i), (iv) e in misura minore (iii) e (v) si prestano bene in quanto richiedono poco contatto con il mondo esterno.

Venivano anche discusse le ricerche esaustive e la formazione di ipotesi per induzione (come realizzeranno negli anni Ottanta i programmi *Bacon* di Herbert Simon [1916-2001]).

Erano ripresi gli argomenti della conferenza del 1947 rispetto al *fair play* che dovrebbe guidare il giudizio sulle macchine. Anche se si rinuncia a costruire una persona intera, «faremo bene a confrontare ogni tanto le condizioni di una macchina con quelle di una persona»:

Non sarebbe leale aspettarsi che una macchina uscita fresca dalla fabbrica competi con un laureato su un piede di parità. Il laureato ha avuto contatti con altri esseri umani per venti e più anni. Questo contatto ha continuato a modificare il suo modello di comportamento lungo tutto il periodo. I suoi insegnanti si sono sforzati intenzionalmente di modificarlo. Alla fine di quel periodo un largo numero di rigide procedure sono state superimposte sullo schema originale del suo cervello.

[...] dovremmo incominciare con una macchina che abbia una capacità minima di eseguire operazioni elaborate o di reagire in modo disciplinato ai comandi (che prendono la forma di interferenza). Nel seguito, applicando un'interferenza adatta, mimando l'educazione, dovremmo sperare di modificare la macchina finché si potesse far affidamento sul suo produrre reazioni definite a certi comandi.

L'addestramento di un bambino dipende in larga misura da un sistema di premi e punizioni, e questo suggerisce che dovrebbe essere possibile riuscire a realizzare l'organizzazione con solo due stimoli di interferenza, uno per «piacere» o «premio» e uno per «dolore» o «punizione».

Turing descriveva un esperimento fatto con una macchina del tipo delle calcolatrici logiche, ma largamente incompleta (macchina non organizzata di tipo P). «Quando si raggiunge una configurazione per cui l'azione è indeterminata, si fa una scelta casuale del dato mancante, che viene regolarmente inserito, provvisoriamente, nella descrizione e si attiva il risultato. Quando occorre uno stimolo di dolore tutti gli inserimenti provvisori sono cancellati, mentre quando occorre uno stimolo di piacere tutti sono resi permanenti».

La tecnica effettiva attraverso cui l'organizzazione della macchina

era stata portata a compimento risultò tuttavia per Turing deludente. Non era sufficientemente analoga al tipo di processo con cui si insegnerebbe davvero a un fanciullo, per le troppe applicazioni di dolore. «Io vorrei indagare altri tipi di macchine non organizzate, e anche cercare di elaborare metodi di organizzazione che si avvicinino più fedelmente ai nostri metodi di educazione».

Turing avvertiva infine che se la mente non addestrata del neonato deve diventare intelligente, essa dovrà acquisire sia disciplina sia capacità di iniziativa. Convertire un cervello o una macchina in una macchina universale è la forma più estrema della disciplina. Ma la disciplina da sola non è certamente sufficiente a produrre intelligenza, anche se Turing confessava di non essere ancora riuscito a immaginare come stimolare l'iniziativa.

Il lavoro terminava con un paragrafo dal titolo “L’intelligenza come un concetto emotivo”:

Quanto siamo disposti a considerare intelligente il comportamento di qualcosa dipende tanto dal nostro stato mentale e dalla nostra preparazione quanto dalle proprietà dell’oggetto in esame.

Se siamo capaci di spiegare e predire il suo comportamento, o se ci sembra che dietro ad esso ci sia poca pianificazione, non siamo molto tentati di immaginare un’intelligenza in azione. Lo stesso oggetto perciò è possibile che una persona lo consideri intelligente e un’altra, quella che ha scoperto le regole del suo comportamento, no. È possibile fare un piccolo esperimento in questa direzione, anche allo stato presente delle nostre conoscenze. Non è difficile preparare una macchina di carta che giochi non male a scacchi. Si prendano ora tre persone A, B, C come soggetti dell’esperimento. A e C devono essere giocatori di scacchi mediocri, B l’operatore che manovra la macchina di carta. [...] Si usano due stanze separate con accorgimenti del caso per comunicare le mosse, e si gioca un gioco tra C e o A o la macchina di carta. C può trovare notevole difficoltà a dire contro chi sta giocando. (Questa è una forma idealizzata di un esperimento che io ho in effetti compiuto.)

L’esperimento sarà elaborato in seguito e chiamato gioco dell’imitazione.

### 13. *L’intelligenza delle macchine*

Nel 1948 Turing accettò l’invito di Newman a trasferirsi a Manche-

ster dove Newman aveva avuto il finanziamento e stava costruendo un calcolatore più piccolo di ACE ma completamente ispirato alla visione di Turing. Anche gli obiettivi per il suo uso erano in sintonia con la visione di Turing: li aveva espressi Newman in una lettera a von Neumann del febbraio 1946 spiegando di voler avere a Manchester un calcolatore «per problemi matematici di un genere completamente differente da quelli finora affrontati dalle macchine, per esempio il teorema dei quattro colori o vari teoremi sui reticoli, gruppi, etc.».

Turing doveva sovrintendere all'uso della macchina, già impostata senza la sua collaborazione; per esempio nell'agosto del 1948 egli scrisse un programma per la divisione da provare sulla macchina, quindi un altro per la fattorizzazione.

Intanto aveva anche scritto con David Champernowne (che dai tempi di Bletchley Park umiliava Turing nelle partite a scacchi) un programma *Turochamp* basato sull'idea tradizionale del minimax, che si riprometteva di scrivere nei dettagli per giocare contro il programma *Machiavelli* di Shaun Wylie e Michie. La prima partita registrata e analizzata fu nel 1951 contro Allick Glennie, giocatore medio, che vinse alla ventinovesima mossa<sup>33</sup>.

Il 1949 fu ricco di episodi minori nella carriera di Turing. Nel 1948 era uscito il libro *Cybernetics* di Norbert Wiener (1894-1964) che ebbe molta risonanza e diede il via a un movimento per la cibernetica seguito anche in Gran Bretagna. Wiener dichiarava di aver discusso le tesi del libro con Turing, che invece era scettico sulla nuova scienza. Venne in Inghilterra McCulloch come rappresentante del movimento cibernetico, e incontrò Turing, che lo considerava un ciarlatano. Nell'occasione di un simposio sulla teoria dell'informazione si ebbe anche una visita di Shannon, che si ritrovò con Turing a discutere di un articolo sul gioco degli scacchi appena scritto.

Il neurochirurgo Sir Geoffrey Jefferson in una conferenza nel giugno 1949 attaccò l'idea di studiare separatamente i meccanismi nervosi; scettico in generale, concludeva che finché una macchina non comporrà un sonetto non si potrà dire che sia uguale a un cervello. Interpellato dal *Times* Turing rispose con il suo solito spirito osservando che un sonetto scritto da una macchina potrebbe essere meglio valutato da una macchina che non da una persona. Ammetteva che la ricerca

---

<sup>33</sup> L. Pachman e V.I. Kuhnmond,  *puter Chess*, Routledge & Kegan Paul, 1985, pp. 22-4. 

in corso era rivolta a riconoscere di quali attività intellettuali una macchina è capace e fino a che punto può pensare da sola.

Intervennero anche i rappresentanti delle scuole cattoliche, lodando Jefferson come esempio di scienziato che prende le distanze da vaneggiamenti blasfemi. Il tutto costituì una buona pubblicità, ed era un segno della sempre maggiore attenzione di intellettuali e *media* alla problematica.

Il 24 giugno Turing partecipò alla presentazione del calcolatore EDSAC di Cambridge leggendo "Checking a large routine" (scritto pare l'anno prima, nel 1948), il primo esempio di dimostrazione matematica di correttezza di un programma.

Peter Hilton, che era arrivato a Manchester, lo introdusse al problema della parola<sup>34</sup>, che Turing non conosceva e che lo attrasse. Erano molti gli argomenti di matematica che Turing non conosceva, sia per aver smesso praticamente di tenersi aggiornato dopo il 1938, sia per la sua attitudine a concentrarsi su problemi specifici risolvendoli con una soluzione originale.

Per questo motivo non ebbe mai la statura di grande matematico. Dopo pochi giorni Turing comunicò di aver dimostrato insolubile il problema per semigrupperi; poco prima del seminario di presentazione si accorse di un errore, sicché la dimostrazione rimase valida solo per semigrupperi con cancellazione.

La tecnica dimostrativa era più complicata di quella per il problema della decisione logica. Turing mandò il lavoro a von Neumann per la pubblicazione e questi nella risposta gli comunicò che il suo calcolatore non era ancora del tutto a posto, almeno ancora per un anno, e si informò su cosa stesse facendo.

Nell'ottobre del 1949 Turing partecipò a un seminario su mente e macchine calcolatrici al dipartimento di filosofia di Manchester organizzato da Michael Polanyi, un epistemologo di Manchester di formazione chimica contrario alla esaltazione della macchina di Newman come 'cervello'; Turing discusse soprattutto con J.Z. Young, fisiologo del sistema nervoso, sulle cellule del cervello, continuando in seguito a interpellarlo con scambi epistolari.

Nello stesso mese la macchina che sarà chiamata Ferranti Mark I era pronta per passare in costruzione. I primi calcoli eseguiti sulla macchi-

---

<sup>34</sup> Il problema della parola, per strutture con regole di riscrittura, è il problema di decidere se due stringhe dell'alfabeto sono equivalenti o no, cioè trasformabili con le regole l'una nell'altra.

na di Manchester riguardarono gli zeri della funzione di Riemann e la cristallografia a raggi X. Turing scrisse un *Programmer's Handbook*, finito nel 1951, circa 100 pagine, ricco di utili consigli.

Nell'autunno del 1949 si realizzò l'ultimo contributo di Turing di tipo ingegneristico, un generatore di numeri casuali, costruito con l'aiuto di Geoff Tootill, che produceva veri numeri casuali, non semplici numeri pseudocasuali, a partire dal rumore esterno.

Nel 1950, pare su suggerimento di Michael Polanyi, Turing scrisse un articolo per la rivista filosofica *Mind*, "Computing Machinery and Intelligence". L'articolo fu apprezzato da Bertrand Russell. Si tratta forse del lavoro più conosciuto presso il largo pubblico, sia perché a lungo il più accessibile, visto il segreto per anni imposto sulla maggior parte dei suoi altri scritti, sia per il carattere degli argomenti. Turing riprendeva e arricchiva, nel numero e nella profondità della discussione, le obiezioni alla possibilità dell'intelligenza delle macchine, che aveva già affrontato nel lavoro del 1948. Le obiezioni sono confutate tutte sulla base del gioco dell'imitazione.

Il gioco, anch'esso un perfezionamento di quello presentato alla fine del lavoro del 1948, è un modo di sostituire una definizione di intelligenza, la quale Turing comunque dubita che si possa dare, neanche con una vasta analisi statistica degli usi di questa parola. «La questione originale "Possono pensare le macchine?" credo che sia troppo priva di significato per meritare una discussione».

Il gioco si gioca con tre persone, un uomo, una donna e un interrogante che, attraverso domande e risposte scritte, senza vedere, deve capire chi è l'uomo e chi è la donna. Ora Turing si chiede cosa succede se l'uomo è sostituito da una macchina: «L'interrogante darà risposte errate altrettanto spesso di quando il gioco è giocato tra un uomo e una donna? Queste questioni sostituiscono l'originale "Possono pensare le macchine?"».

Il gioco dell'imitazione è stato chiamato in seguito test di Turing. Questi ha usato il termine soltanto una volta, in un dibattito radiofonico, in modo colloquiale. Lo si è riformulato come se una macchina che riuscisse a ingannare un umano sulla propria identità supererebbe un test di intelligenza. Turing era interessato invece a capire la portata del linguaggio mentalistico che noi inevitabilmente (lui per primo) siamo portati a utilizzare parlando delle macchine.

Turing non era particolarmente ottimista sul comportamento delle macchine nel gioco dell'imitazione. In seguito, a tutt'oggi, si organizzeranno tornei dove le macchine partecipanti cercano di superare il test di Turing; per ora pare che di nessuna si possa dire che lo ha supe-



rato, ma si cita con rispetto il computer Watson dell'IBM che nel 2011 ha vinto contro avversari umani il gioco televisivo *Jeopardy!*

Nell'articolo del 1950 Turing arrivava anche a definire il quadro entro il quale si dovevano considerare le prestazioni dei nuovi calcolatori e, per confronto, quelle del cervello. Il concetto generale prescelto è quello delle macchine a stati discreti, tra cui rientrano i calcolatori digitali. «Queste macchine, in ultima analisi, non esistono, nel mondo reale tutto si muove con continuità. Esistono tuttavia numerosi tipi di macchine che possono essere *considerate*, per ragioni di utilità, come macchine a stati discreti. [...] I calcolatori digitali, [nella loro capacità] di imitare qualsiasi macchina a stati discreti, sono *macchine universali*».

Il sistema nervoso centrale non è certamente una macchina a stati discreti, per il ruolo che svolgono nel suo funzionamento chimica ed elettrica. Tuttavia le caratteristiche del cervello rilevanti per il pensiero e l'intelligenza rientrano nel livello di descrizione della macchina a stati discreti, e le operazioni di una macchina a stati discreti sono computabili.

«Se proprio vogliamo trovare delle somiglianze [tra cervello e una macchina a stati discreti] faremmo forse meglio [invece di pensare per esempio al possibile uso dell'elettricità] a cercare analogie matematiche a livello delle funzioni». Questa è la fonte di quello che sarà chiamato funzionalismo alla Turing nelle scienze cognitive.

Il filosofo John Searle ha costruito la sua fortuna su una trentennale contestazione del test di Turing e del funzionalismo, argomentando che se una macchina fatta di lattine di birra fosse in grado di condurre una conversazione in cinese, non per questo diremmo che conosce il cinese<sup>35</sup>.

#### 14. *La morfogenesi*

Nel 1950 Turing iniziò ad interessarsi sul serio di questioni biologiche. Da una parte discuteva con Young sui neuroni, il loro numero e la possibilità di stimolarne la crescita in direzione della formazione di circuiti dedicati. Dall'altra era affascinato dal problema della crescita

---

<sup>35</sup> J.R. Searle, "Minds, Brains and Programs", *Behavioral and Brain Sciences*, 3, 1980, pp. 417-58; trad. it. con integrazioni del 1990 in J.R. Searle, "La mente è un programma?", in *Mente e macchina*, Quaderno Le Scienze, 66, giugno 1992, pp. 5-10.

degli esseri viventi e della formazione di *pattern*. Aveva letto prima della guerra il libro di D'Arcy Wentworth Thompson (1860-1948), *Growth and Form* (1917). Già nelle *Natural Wonders* di Brewster si descriveva come da un uovo si formasse una creatura vivente, non con una trasformazione continua ma attraverso suddivisioni ripetute. Il problema generale era come la materia biologica potesse distribuirsi in *pattern* complicati dove ogni cellula andava a prendere un posto definito come se fosse in grado di comunicare con cellule distanti per coordinarsi: il modo come la materia prende forma era l'oggetto della *morfogenesi*.

I biologi usavano parole allusive e indefinite, come «forza vitale», o «campo morfogenetico» per ipotizzare una sorta di modello invisibile che dettava lo sviluppo armonioso. Il «campo» era accettato come esistente, senza conoscere la sua natura e il suo modo di operare, che si pensava dipendesse misteriosamente dalla chimica. Nel 1943 Erwin Schrödinger (1887-1961), nella conferenza *What is life*<sup>36</sup>, aveva espresso la convinzione che l'informazione genetica fosse memorizzata a livello molecolare. James D. Watson, Francis Crick e Rosalind Russell a Cambridge studiavano il problema della trasmissione dell'informazione. A Turing importava capire come l'informazione potesse essere tradotta in azione.

Turing decise di accettare l'ipotesi che il «campo morfogenetico» fosse definito da variazioni nelle concentrazioni chimiche di sostanze che chiamava morfogeni (e il termine è rimasto): una mistura di soluzioni chimiche, diffondendosi e reagendo le une con le altre, poteva stabilizzarsi in un flusso di onde chimiche: onde di concentrazione che si sarebbero indurite in tessuti coinvolgendo milioni di cellule e organizzandole in distribuzioni simmetriche su vasta scala.

Il problema fondamentale era quello della rottura della simmetria circolare della cellula, ed era esemplificato dalla gastrulazione: in una sfera perfetta improvvisamente si determina una scanalatura, che individua una testa e una coda. Ma se la simmetria della sfera è perfetta e le equazioni chimiche pure simmetriche, sembra che in questo atto decisivo sia creata informazione. Polanyi sosteneva nei suoi scritti la necessità del ricorso a una forza spirituale.

Turing pensò che se a livello chimico prevale la simmetria, a livello molecolare i movimenti individuali potevano non essere perfettamente simmetrici, sicché si poteva determinare la prevalenza di una dire-

---

<sup>36</sup> E. Schrödinger, *What is Life?*, Cambridge Univ. Press, 1952.

zione, come nel caso della cristallizzazione dei liquidi. La soluzione gli era suggerita dalle proprietà di biforcazione delle soluzioni delle equazioni differenziali: quando il parametro di un sistema passa attraverso un determinato valore critico si ha un cambiamento qualitativo nel comportamento, laddove uno stato prima stabile diventa instabile.

La situazione è simile a quella che si presenta in riferimento agli oscillatori elettrici. Capire come un oscillatore si mantiene una volta che ha iniziato è facile, ma a una prima considerazione non è ovvio come l'oscillazione abbia inizio. La spiegazione è che esistono perturbazioni casuali sempre presenti nel circuito. Ogni perturbazione la cui frequenza è la frequenza naturale dell'oscillatore tenderà a metterlo in moto.

Turing costruì nel suo ufficio un sistema di circuiti che usava per mostrare come venissero gradualmente a porsi in risonanza.

Il processo di inizio di un *pattern* di oscillazioni poteva essere descritto come la rottura di un equilibrio instabile. Nel caso delle cellule sferiche, si doveva mostrare che in qualche modo, attraverso una variazione della temperatura, o in presenza di una catalisi, l'equilibrio della bilancia chimica improvvisamente diventava instabile.

Per analizzare questo punto di crisi iniziale matematicamente, si dovevano introdurre forti semplificazioni:

[...] descriveremo un modello matematico della crescita dell'embrione. Il modello sarà un semplificazione e una idealizzazione, e quindi una falsificazione. Si spera che gli elementi mantenuti per la discussione siano quelli di maggiore importanza allo stato presente della conoscenza.

Con tutte le semplificazioni, le equazioni per un brodo di sole quattro soluzioni chimiche interagenti risultano ancora intrattabili, perché non lineari; non si potevano usare i metodi familiari della teoria elettromagnetica, che descrivono il sistema come somma di tante parti separate. Ma il momento critico del germoglio [*budding*], quando il sistema instabile si cristallizza in una forma, poteva essere trattato come un fenomeno lineare, e le equazioni linearizzate.

Turing considerò il caso dell'idra (menzionato da Brewster), un piccolo verme di acqua dolce di forma tubolare che poteva far rinascere una testa e una coda da qualunque sua sezione. Usando un modello con due sostanze chimiche in reazione e diffusione lungo l'anello, Turing diede un'analisi teorica di tutte le possibilità al momento del *budding*, e mostrò che sotto certe condizioni i reagenti si sarebbero

stabilizzati in onde di concentrazione definendo un numero di lobi sull'anello che probabilmente avrebbero formato la base per la forma dei tentacoli.

L'analisi mostrava anche la possibilità che le onde si concentrassero in masse asimmetriche, che ricordavano le strisce e le macchie irregolari del mantello di certi animali<sup>37</sup>.

Un altro obiettivo nello studio della forma che Turing intendeva affrontare riguardava la presenza dei numeri di Fibonacci nella posizione a spirale delle pigne degli abeti, nella distribuzione delle foglie nelle piante più comuni, dei semi nei girasoli; per il momento lo scopo era troppo ambizioso, sarebbe stata necessaria l'analisi di una superficie a due dimensioni, un problema molto più difficile.

Turing nel corso del 1951 scrisse un lavoro che apparirà nel 1952<sup>38</sup>. L'oscillatore chimico non lineare individuato da Turing verrà a essere conosciuto come reazione di Bolousov-Zhabitsky. Turing intanto non era più interessato ai calcolatori, se non per il loro possibile uso di calcolo, sempre per gli zeri della funzione di Riemann e per le sue indagini chimiche; sul Ferranti Mark 1 fece i calcoli relativi alle equazioni non lineari del modello di reazione-diffusione, forse la prima utilizzazione di questo tipo. Gli ingegneri erano già impegnati sulle nuove versioni della macchina, ma le nuove generazioni non sapevano nulla del ruolo giocato da Turing<sup>39</sup>. Si stava realizzando la sua visione di un centro di calcolo dove venivano portati problemi, risolti dai signori e dai servi; si stava costruendo una libreria di programmi; iniziavano le applicazioni commerciali e si parlava dell'introduzione dell'insegnamento nella scuola.

Turing si dedicava a questioni matematiche tradizionali; una sintesi del lavoro sugli zeri della funzione di Riemann apparve nel 1953; i calcoli che Turing sviluppava in base 32 non furono soddisfacenti, ma il metodo era buono e fu applicato nel 1955 da D.H. Lehmer per verificare che i primi 25.000 zeri cadono sulla linea critica. Nell'aprile del 1951 Turing ottenne un altro risultato sul problema della parola per i gruppi, apprezzato dallo specialista J.H.C. Whitehead ma non pubbli-

---

<sup>37</sup> Nel numero di marzo 2012 di *Nature Genetics* è apparsa la conferma della individuazione di due morfogeni, eccitatore e inibitore, previsti nel suo modello.

<sup>38</sup> "The Chemical Basis of Morphogenesis", *Phil Trans. Roy. Soc., series B*, 237, 1952, pp. 37-72.

<sup>39</sup> Nel volume *Faster Than Thought*, a cui Turing contribuì con un capitolo sui giochi (si veda oltre), e dedicato allo sviluppo dei calcolatori in Gran Bretagna, non c'era alcuna menzione del contributo pionieristico di Turing.

cato; partecipava a trasmissioni alla BBC, come quella del maggio 1951 sulle macchine e il pensiero.

Il 15 marzo del 1951 fu eletto *Fellow* della Royal Society, su presentazione di Newman e di Russell. In una risposta alle diverse congratulazioni ricevute, scrisse: «Spero che la motivazione non sia “distintosi per lavoro su problemi insolubili”».

Ma la sua attenzione era concentrata sulla morfogenesi. Raccoglieva e catalogava tutte le forme di fiori e piante. L'argomento che voleva affrontare era quello della fillotassi, la disposizione delle foglie nelle piante.

Siccome i suoi conoscenti appartenevano tutti al mondo della matematica o dei calcolatori, questa parte della vita di Turing è poco nota, mancano ricordi e confidenze, ma non c'è da dubitare che fosse preso da grande entusiasmo; si aspettava di lasciare una traccia come quella del 1936, e la delusione per la tiepida accoglienza immediata fu forte.

### 15. *Il processo*

Nel dicembre del 1951, sotto Natale, Turing conobbe Arnold Murray, un disoccupato che viveva di espedienti e piccoli furti. Gli incontri proseguirono nel gennaio; il 23 gennaio 1952 Turing trovò la casa svaligiata; presentò denuncia, e avendo conservato un bicchiere usato dai ladri, per le impronte digitali, la polizia fu in grado di risalire a un personaggio noto, che confessò e dichiarò di essere stato introdotto nell'appartamento da Murray. Turing, caduto in contraddizione nel descrivere il conoscente, alla fine confessò di «avere una relazione» [*affair*] con questa persona.

Gli investigatori furono sorpresi dalla disponibilità di Turing ad ammettere la natura del rapporto, nonostante così venisse a ricadere sotto la legge che vietava rapporti omosessuali anche tra adulti consenzienti<sup>40</sup>. Seguì inevitabilmente il processo, e Turing fu condannato per *Gross Indecency*; aveva la possibilità di scegliere tra un periodo di carcere e sottoporsi a un trattamento organo-terapico. Questo trattamento, altrimenti detto castrazione chimica, consisteva nella somministrazione di ormoni femminili (in precedenza erano stati sperimentati con scarso successo ormoni maschili). Turing per poter continuare a lavorare scelse questa possibilità, nonostante avesse come conseguen-

---

<sup>40</sup> La legge fu abolita nel 1967.

za l'impotenza (secondo alcuni temporanea) e altri effetti sui caratteri sessuali secondari (si lamenterà che gli sta crescendo il seno).

A seguito della condanna, Turing diventò persona inaffidabile, pericolosa per lo stato, visti i segreti militari di cui era a conoscenza; perse tra l'altro il diritto a recarsi negli Stati Uniti. Fu sottoposto a continua sorveglianza e se anche i suoi movimenti non furono impediti, lo furono quelli dei suoi conoscenti; per esempio fu respinto alla frontiera un amico norvegese che veniva a trovarlo (Turing aveva pensato di cercare relazioni fuori dalla Gran Bretagna, e scelto come primo paese la Norvegia<sup>41</sup>, imparando anche la lingua abbastanza bene da leggere le fiabe di Andersen).

Nei giorni del processo Turing partecipava a un seminario al dipartimento di Chimica dove dovette difendersi dalle critiche mosse al suo lavoro da Ilya Prigogine (1917-2003). Questi in seguito, non ricordando più questo episodio, avrebbe dichiarato: «Lo sviluppo della termodinamica irreversibile dei sistemi aperti da parte della scuola di Bruxelles aveva portato all'inizio degli anni Cinquanta allo studio dei processi non lineari. [...] Fu solo allora che ci accorgemmo di un notevole articolo di A.M. Turing (1952) che aveva di fatto costruito un modello chimico che esibiva instabilità. Il lavoro ci era prima sfuggito perché trattava l'argomento più specifico della formazione di *pattern* morfogenetici».

Il giudizio storico sul lavoro "Chemical Basis of Morphogenesis" è così riassunto dal curatore del volume relativo nelle opere complete, P.T. Saunders: esso contiene due idee fondamentali; la prima è che un *pattern* può formarsi grazie alla instabilità delle soluzioni costanti di semplici e plausibili equazioni differenziali; la seconda che esso è completamente determinato dalle equazioni e dalla forma della regione, senza necessità di ipotizzare altri processi, come l'attivazione di geni in diverse cellule; il modello funziona sia che le regioni siano divise in cellule sia che siano continue. «Qualunque sia il destino del modello di reazione-diffusione, questi principi sono destinati a restare fondamentali nella modellizzazione biologica».

## 16. *La fine*

Nel gennaio del 1952 Turing prese parte a una trasmissione della BBC sul tema «Si può dire che le macchine calcolatrici automatiche

---

<sup>41</sup> In seguito viaggiò anche in Grecia.

pensano?», dialogando con Jefferson e con il filosofo della scienza R. B. Braithwaite, moderatore Newman<sup>42</sup>. La discussione è divertente per un'uscita di Braithwaite che chiede a Turing se davvero egli pensi che in futuro saranno le macchine a consultare per noi l'orario ferroviario. La parte interessante riguarda la convinzione espressa da Turing della possibilità di meccanizzare la formazione di analogie, ispirandosi al probabile funzionamento del cervello al riguardo.

Quel che restava della vita di Turing si sfilacciava ora in vari rivoli; il più serio riguardava sempre la morfogenesi, in relazione alla quale Turing lasciò alcuni manoscritti (ora nelle opere complete), uno in collaborazione con un botanico, Claude W. Wardlaw: "A Diffusion Reaction Theory of Morphogenesis in Plants", uno con un giovane studente, Bernard Richards: "Morphogen Theory of Phyllotaxis", e infine un suo "Outline of the Development of the Daisy". Nel secondo di questi lavori è data una descrizione geometrica della distribuzione delle foglie, in particolare quella a spirale: Turing usava una rappresentazione reticolare, svolgendo il cilindro (del ramo) sul piano, a differenza delle sezioni trasversali usate dai botanici, e riusciva a dare una spiegazione semplice della comparsa dei numeri di Fibonacci; nella seconda parte provava ad applicare la sua teoria della reazione-diffusione.

Non riuscì tuttavia a spiegare come promesso la forma delle pigne degli abeti. Riprese a interessarsi di meccanica quantistica, promettendosi di costruire un calcolo degli spinori (restano poche pagine incomplete). Scrisse una proposta per "The Reform of Mathematical Notation".

Due ultimi scritti da segnalare riguardavano la teoria dei giochi e i problemi indecidibili. Il primo apparve come capitolo di un libro<sup>43</sup>, e per quel che riguarda gli scacchi Turing descriveva una ricerca che conciliava due tendenze possibili ed entrambe in seguito perseguite: da una parte una che non dipende dal processo di pensiero umano, dall'altra quella che considera come i giocatori stessi valutano le mosse. Turing tuttavia non arrivò a considerare che i grandi giocatori più che calcolare nello spazio delle mosse giudicano le distribuzioni dei pezzi sulla scacchiera.

L'altro scritto è dedicato all'argomento dei problemi indecidibili, ed è mirato a introdurre al risultato di P.S. Novikov (1901-75) del 1952 sulla indecidibilità del problema della parola per i gruppi.

---

<sup>42</sup> Si veda in questo volume, pp. 00-00.



<sup>43</sup> B.V. Bowden (ed.), *Faster Than Thought*, Pitman, 1953, cap. 25, pp. 286-310.

Frequentò uno psicoanalista junghiano, Franz Greenbaum, con cui strinse amicizia, che gli raccomandò di registrare i sogni per l'interpretazione, da cui con sorpresa di Turing emerse una ostilità latente per la madre. I quaderni riempiti degli appunti sconvolsero il fratello quando gli furono mostrati dallo psicoanalista. Dopo il processo Turing scrisse l'abbozzo di un racconto ispirato alla sua disavventura, poche pagine nelle quali inserì alcune osservazioni acute di autoanalisi.

Il protagonista del racconto, Alec Pryce, ha legato il suo nome a una invenzione chiamata «boa di Pryce», e Alec «arrossiva di piacere quando sentiva usata questa frase». Il suo lavoro riguardava i viaggi interplanetari, gli erano sempre piaciuti gli argomenti eccentrici, ma benché di solito si lasciasse andare parlando con i giornalisti o sul Terzo Programma, quando scriveva per lettori tecnicamente esperti il suo lavoro era sempre serio e fondato, o almeno lo era stato in gioventù. A Natale, Alec girava guardando le vetrine finché non gli veniva un'ispirazione per i regali da fare, una sorta di «allegoria del suo metodo di lavoro, l'attesa dell'ispirazione». Viene descritto il suo modo trasandato di vestire, una specie di uniforme studentesca «adatta alla sua età mentale», che lo faceva sentire più giovane. Gli uomini che non erano considerati complici sessuali erano per lui un sostituto del padre, a cui doveva dimostrare la sua forza intellettuale.

Nel febbraio del 1954 Turing fece un nuovo testamento. L'8 giugno fu trovato morto vicino al letto dalla donna di servizio, con saliva intorno alle labbra; il patologo dichiarò evidenti i segni di avvelenamento da cianuro; in casa c'era un barattolo di cianuro di potassio, e uno di soluzione di cianuro. A fianco al letto si trovò una mezza mela, morsicata più volte; la mela non fu mai analizzata, sicché non si può affermare con sicurezza che fosse stata immersa nella soluzione di cianuro. Turing era affascinato dalla storia di Biancaneve. Qualcuno pensa che abbia disposto la scena in modo che la madre potesse credere a un incidente di lavoro, come di fatto successe. Altri, con scarsi elementi fattuali, ma con un argomento plausibile, hanno avanzato il sospetto che si sia trattato di un suicidio di stato.

### *Lecture consigliate*

La biografia di Turing di Andrew Hodges, che resterà a lungo, forse per sempre insuperata, è: A. Hodges, *Alan Turing, The Enigma of Intelligence*, Hutchinson, London, 1983, riedita nel 2012 (Vintage, Random



House, London) con una nuova prefazione; trad. it. *Storia di un enigma. Vita di Alan Turing, 1912-1954*, Bollati Boringhieri, Torino, 1991.

Una breve presentazione della figura in Turing, sempre scritta da A. Hodges, è in *Turing*, Phoenix, London, 1997; trad. it. *Turing. Un filosofo della natura*, Sansoni, Firenze, 1998.

Esiste anche una biografia scritta dalla madre Sara Stoney Turing, *Alan M. Turing*, Heffers, Cambridge, 1959, ristampata nel 2012 da Cambridge Univ. Press con una prefazione di Martin Davis, e una postfazione del fratello maggiore John F. Turing, una persona *square* assolutamente incapace, come la madre, di capire e accettare le difficoltà psicologiche e le eccentricità di Alan; madre e fratello insieme forniscono uno spaccato interessante di una famiglia della borghesia inglese.

Sconsigliabili sono il romanzo di David Leavitt, *The Man Who Knew Too Much*, W.W. Norton, New York, 2006; trad. it. *L'uomo che sapeva troppo. Alan Turing e l'invenzione del computer*, Codice, Torino, 2007, che presenta Turing come socialmente inetto e sessualmente frustrato, e la recente biografia di B.J. Copeland, *Turing*, Oxford Univ. Press, 2012, al contrario reticente sugli aspetti privati della personalità di Turing.

Andrew Hodges cura il sito <http://www.turing.org.uk/turing/index.html>. Si veda in particolare:

<http://www.turing.org.uk/bio/index.html>

<http://www.turing.org.uk/sources/biblio.html>

Un altro sito curato da B.J. Copeland è <http://www.turingarchive.org> dove sono digitalizzati documenti del King's College di Cambridge.

Le opere di Turing sono pubblicate in quattro volumi di *Collected Works*, da North Holland (Elsevier Science B. V.), Amsterdam:

*Pure Mathematics*, (J.L. Britton ed.), 1992

*Mechanical Intelligence* (D.C. Ince ed.), 1992

*Morphogenesis*, (P.T. Saunders ed.), 1992

*Mathematical Logic* (R.O. Gandy & C.E.M. Yates eds.), 2001; quest'ultimo, ritardato rispetto agli altri dalla scomparsa del curatore R.O. Gandy, logico, unico studente di dottorato di Turing e suo amico, contiene in compenso anche alcuni scritti non di logica (uno di crittografia in particolare) che nel frattempo sono stati liberati dal segreto militare.

Nella prospettiva del centenario sono stati ristampati con minuziosi e utili commenti l'articolo del 1936 e la tesi di dottorato del 1939:

C. Petzold, *The Annotated Turing*, Wiley, Indianapolis, 2008

A.W. Appel (ed.), *Alan Turing's Systems of Logic*, Princeton Univ. Press, Princeton, 2012.

Una corposa antologia, con ampi lavori di presentazione, e numerosi contributi che illustrano tutto il lavoro di Turing è offerta da:

S.B. Cooper e J. van Leeuwen (eds.), *Alan Turing: His Work and Impact*, Elsevier Science, 2013, pp. 944, che arricchisce la precedente raccolta

B. J. Copeland (ed.), *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life*: plus *The Secrets of Enigma*, Clarendon Press, Oxford, 2004, pp. 613.

In italiano una scelta degli scritti del volume *Mechanical Intelligence* è tradotta in A.M. Turing, *Intelligenza meccanica*, a cura di G. Lolli, Bollati Boringhieri, Torino, 1994.

Il volume contiene:

“Proposal for the Development in the Mathematics Division of an Automatic Computing Engine (ACE)”, rapporto all’Executive Committee del National Physical Laboratory del 1945, pubblicato per la prima volta in B.E. Carpenter e R.N. Doran (eds.), *A.M. Turing’s ACE Report of 1946 and Other Papers*, MIT Press, Cambridge MA, 1986, pp. 20-105

*Lecture at the London Mathematical Society*, del 20 febbraio 1947

“Intelligent Machinery” (1948), rapporto interno del National Physical Laboratory, pubblicato per la prima volta in B. Meltzer e D. Michie (eds.), *Machine Intelligence 5*, Edinburgh Univ. Press, Edinburgh, 1969, pp. 3-23

“Computing Machinery and Intelligence”, *Mind*, 59, 1950, pp. 433-60.

L’intervento radiofonico del 1951 e il dibattito alla BBC del 14 gennaio 1952, con la partecipazione di Turing, **proposti** in questo volume alle pp. 127-42 sono stati pubblicati in italiano **da** A.M. Turing, “Si può dire che i calcolatori automatici pensano?”, *Sistemi Intelligenti*, 10, 1998, pp. 27-40.

## Secondo solo a Newton nel Pantheon di Cambridge...

---

Alan M. Turing (1912-54) è stato forse l'ultimo degli scienziati universali del ventesimo secolo, prima della istituzionalizzazione della *big science*. Nasce nello stesso anno in cui muore Henri Poincaré (1854-1912), un altro matematico universale, e la coincidenza ha forti valenze simboliche: la matematica è stata profondamente cambiata in seguito al lavoro di Turing, in una misura che ancora non è possibile valutare. Ma Turing ha influenzato in modo decisivo anche alcuni dei mutamenti culturali più importanti della fine del ventesimo secolo e del periodo che stiamo vivendo, la civiltà del virtuale e delle tecnologie dell'informazione.

In questo intervento vogliamo discutere perché, nonostante l'ampiezza e profondità dei suoi contributi, Turing sia conosciuto in modo superficiale e soprattutto come mai le sue idee siano generalmente volgarizzate e deformate nella divulgazione.

Il pensiero di Turing è sempre squisitamente scientifico, anche quando parla di argomenti popolari, e risulta troppo difficile da cogliere, seguire e recepire nel difficile equilibrio tra immaginazione, ipotesi e verifica sperimentale; per questo motivo viene ridotto a formule semplificate; laddove sfiora la filosofia, è interpretato in termini di problematiche tradizionali piuttosto rozze, proprio quelle che egli voleva sostituire con una visione scientifica.

Turing ha dato l'impronta alla teoria della programmazione e a tutte le ricerche che l'Intelligenza Artificiale ha perseguito negli ultimi sessanta anni (oltre ad aver costruito di fatto il primo calcolatore elettronico digitale, e aver posto le basi teoriche della teoria della calcolabilità). Ma Turing non era un alieno che trasmettesse conoscenze superiori extraterrestri, la ragione della sua lungimiranza è che nessuno aveva chiaro come lui il significato e la portata della macchina universale, che egli aveva concepito e definito e che gli permetteva di stabilire collegamenti tra campi apparentemente lontani.

Si pensi che ancora a metà degli anni Cinquanta si potevano sentire considerazioni scettiche del tipo seguente, da parte di ricercatori che hanno dato importanti contributi allo sviluppo dei calcolatori:

Se dovesse risultare che la logica di base di una macchina progettata per la soluzione numerica delle equazioni differenziali coincide con la logica di una macchina destinata alla contabilità di un magazzino, direi che si tratta della coincidenza più sorprendente in cui mi sono mai imbattuto<sup>1</sup>.

Invece fin dall'inizio Turing non aveva dubbi:

Questa particolare proprietà dei calcolatori digitali, che possono simulare qualsiasi macchina discreta, si esprime dicendo che essi sono macchine *universali*. L'esistenza di macchine che hanno questa proprietà ha l'importante conseguenza che, se si trascurano considerazioni di velocità, diventa superfluo progettare nuove macchine diverse per diversi processi di calcolo. Essi possono essere tutti realizzati da un calcolatore digitale, opportunamente programmato per ogni caso speciale<sup>2</sup>.


La precoce scomparsa, il fatto che Turing non abbia scritto molto, e che alcuni dei suoi rapporti siano stati a lungo secretati per l'interesse militare (e talvolta quindi le sue idee riscoperte e riproposte da altri indipendentemente), hanno inizialmente impedito il suo apprezzamento e che la sua fama si diffondesse. Una svolta è iniziata con la pubblicazione della biografia di Andrew Hodges nel 1983. Lentamente incomincia ad avere il riconoscimento che merita. È stato detto che se avesse continuato, sarebbe potuto diventare secondo solo a Newton nel Pantheon di Cambridge<sup>3</sup>.

Invece di soffermarci sull'elenco dei contributi scientifici di Turing<sup>4</sup>, tuttavia, vogliamo parlare del suo stile e del suo metodo.


Quello che è stato chiamato lo stile di Turing consiste nel riformulare i problemi in modo che l'onere della prova ricada sull'oppositore scettico; i suoi argomenti allora hanno la caratteristica che sembra che non ammettano confutazione. Gli viene rimproverato talvolta di risolvere non il problema originale, ma un altro, rilievo al quale Turing forse assentirebbe, nella convinzione che la sua versione sia quella cor-

---

<sup>1</sup> H. Aiken, "The Future of Automatic Computing Machinery", *Elektronische Rechenanlage und Informationsverarbeitung*, 33, 1956. Aiken ha costruito l'Harvard Mark I, da una collaborazione di Harvard e IBM.

<sup>2</sup> Turing 1950. sciogliere sigla 

<sup>3</sup> S. Toulmin, *The New York Review of Books*, 19 gennaio, 1984, p. 3.

<sup>4</sup> Per cui rinviamo alla *Nota biografica e scientifica*, in questo volume, pp. 9-50. 

retta, o per lo meno quella produttiva. Nel caso dell'intelligenza delle macchine, per esempio, Turing afferma:

La questione originale «Possono pensare le macchine» credo che sia troppo priva di significato per meritare una discussione. [...] La sostituirò con un'altra, che è strettamente collegata e che è espressa in parole relativamente non ambigue.

La riformulazione del problema non è una mossa retorica, una specie di argomento *ad hominem*, ma discende dall'interesse genuino di Turing. Sa che sembra esserci una contraddizione in termini nel parlare di una macchina con intelligenza, se si pensa a «espressioni di luoghi comuni (*catch phrases*) come “agire come una macchina”». Ma una ragione positiva per credere nella possibilità di costruire macchine (chiamate pensanti) consiste nel fatto che «è possibile fare imitare alle macchine piccole parti dell'uomo».

Già la macchina universale, pur provata in modo inconfutabile con la scrittura delle sue istruzioni e una dimostrazione delle sue caratteristiche, trova la giustificazione più convincente per il lettore in quello che Turing chiama un appello all'intuizione, ma che più esattamente è il fatto che Turing ricava i comportamenti fondamentali della macchina da un'analisi dell'operatore umano. Fino ad allora, *computer* in inglese indicava il calcolatore umano, e infatti Turing usa di preferenza il neologismo *computor*, oppure *computing machine* o simili. Questo spiega perché la descrizione della macchina sia intrisa di riferimenti mentalistici: «il “simbolo esaminato” è il solo di cui la macchina è, per così dire, “direttamente consapevole”. [Ma alterando gli stati] la macchina può effettivamente ricordare alcuni dei simboli che ha “visto” in precedenza».

Anche per le macchine calcolatrici reali, rispetto alla possibilità di ascrivere ad esse forme di intelligenza, l'impostazione di Turing conferma che per lui la propensione a riconoscere un comportamento come intelligente non dipende tanto dal comportamento stesso ma dalla nostra educazione e dalla nostra disposizione mentale.

La misura nella quale riteniamo che qualcosa si comporti in modo intelligente è determinata tanto dal nostro stato mentale e dalla nostra educazione quanto dalle proprietà dell'oggetto in considerazione. Se siamo in grado di predire il suo funzionamento o se ci sembra che dietro ad esso ci sia poca pianificazione, siamo poco tentati da immaginare un'intelligenza.

Le *catch phrases* non tengono conto di possibili usi più sofisticati del termine, al conferimento alle macchine di altre «parti dell'uomo» a cui Turing sta pensando: per esempio se una macchina deve mostrare intelligenza, dobbiamo «disporre che essa dia occasionalmente risposte sbagliate» (in occasione di domande indecidibili).

In particolare Turing ammette di essere portato a usare un vocabolario mentalistico per la sorpresa che prova di fronte a comportamenti inaspettati delle macchine; questi dipendono dal potenziamento imprevisto rispetto «ai processi che sono programmate a fare», e a cui continuano a essere vincolate, dovuto alla possibilità delle macchine di modificare parte delle proprie istruzioni. Paragona la situazione a quella in cui «uno scolaro ha imparato molto dal maestro, ma ci ha messo tanto di suo. Quando questo succede sento di essere obbligato a considerare la macchina come espressione di intelligenza».

Nel 1948 Turing descrive un esperimento che ha compiuto. Ha scritto un programma (una *paper machine*) per giocare, non male, a scacchi. Si immaginino tre persone A, B e C, dove A e C sono giocatori di scacchi scadenti, B è un operatore che aziona la macchina e gioca secondo le sue istruzioni. In due stanze separate, con metodi di comunicazione tra di esse, sono in una C e nell'altra o A o B: «C può trovare molto difficile decidere contro chi sta giocando».

Nel 1950 l'esperimento evolve nel gioco dell'imitazione, così descritto sommariamente: il gioco si gioca con tre persone, un uomo, una donna e un interrogante, che attraverso domande e risposte scritte, senza vedere, deve capire chi è l'uomo e chi è la donna. Turing assume questo gioco come guida nella riflessione sull'intelligenza, e si domanda: cosa succede se l'uomo è sostituito da una macchina? «L'interrogante darà risposte errate altrettanto spesso di quando il gioco è giocato tra un uomo e una donna? Queste questioni sostituiscono l'originale "Possono pensare le macchine?"».

Credo che fra circa cinquanta anni sarà possibile programmare i calcolatori, con una memoria di circa  $10^9$ , in modo da far loro giocare il gioco dell'imitazione così bene che un interrogante medio non avrà più del 70% di probabilità di fare l'identificazione giusta dopo cinque minuti di domande.

Si noti che la previsione potrebbe non essere considerata molto ottimistica. Se l'interrogante medio ha il 70% di probabilità di fare l'identificazione giusta in cinque minuti, vuol dire che la probabilità per la macchina di ingannare l'interrogante è inferiore al 30%. Ciò nonostante «io credo che alla fine del secolo l'uso delle parole e le credenze

generali educate saranno così cambiate che sarà possibile parlare di macchine pensanti senza aspettarsi di essere contraddetti».

Nella letteratura posteriore si parla del gioco come di un «test di Turing», per cui le macchine che lo superano sarebbero intelligenti. Turing ha usato solo una volta il termine «test» per descrivere il suo gioco<sup>5</sup>. Ma come si vede Turing non propone il gioco come un criterio discriminante o definitorio, ma solo come descrizione delle circostanze che potrebbero giustificare un nostro modo di parlare.

Il superamento del test, se vogliamo chiamare così una serie di giocate, non è un criterio necessario per l'attribuzione di intelligenza, né sufficiente forse; non esiste neanche un chiaro criterio di superamento del test, ma solo una valutazione statistica, con il riconoscimento di giocate più o meno buone, nel senso di difficoltà di riconoscimento per l'interrogante paragonabili o no al gioco con partecipanti umani. Sulla base della definizione del gioco, non ha senso pensare a una sola giocata (non è fissato per esempio un tempo di durata) e se di test si vuole parlare sembra più un test sull'intelligenza dell'interrogante.

Si sprecano invece nella filosofia della mente le discussioni se il test sia condizione sufficiente o solo necessaria per l'attribuzione di intelligenza alle macchine, e tutte in generale sono rivolte a confutare la significatività del test; intanto tuttavia la previsione di Turing sul modo di esprimersi delle persone istruite è ormai confermata: sui giornali si legge di cellulari che capiscono il sarcasmo, e di programmi che vincono ai giochi televisivi contro umani.

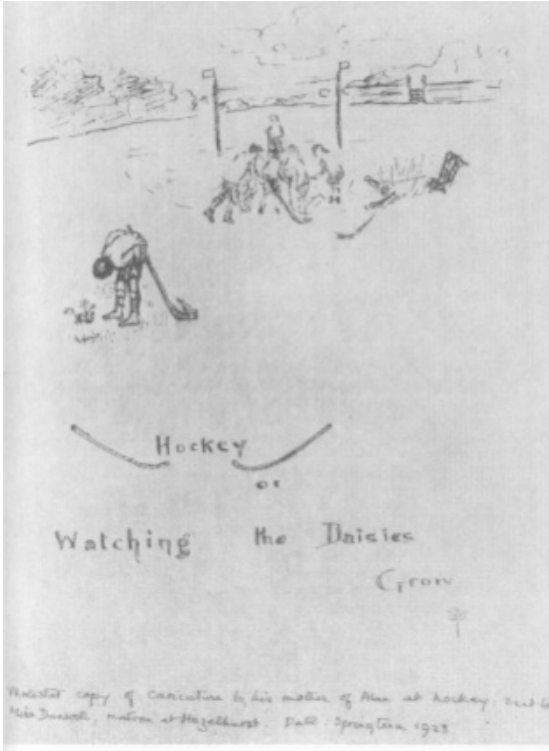
L'originalità e la capacità di spiazzamento delle discussioni di Turing sembrano la naturale conseguenza del suo guardare ai problemi con gli occhi del bambino, non appesantito dalle tradizioni di pensiero che hanno trattato gli stessi problemi, dalle impostazioni correnti, da posizioni filosofiche, né dagli strumenti consolidati, che peraltro possiede ma che nel caso si inventa se quelli disponibili non servono (così per esempio ha riscoperto da solo il teorema del limite centrale in probabilità). Le abitudini del pensiero sono sconvolte, ma la pigrizia fa reinterpretare le sue risposte come posizioni filosofiche, come sopra per la concezione della natura della mente, o per il meccanicismo.

La disposizione a guardare le cose «con gli occhi del bambino» è ben illustrata dal disegno dalla madre quando Turing era piccolo (fig. 1): rappresenta un gruppo di bambini che giocano a hockey in un prato e ce ne è uno che non partecipa alla mischia, sta un po' in disparte – il

---

<sup>5</sup> In un dibattito radiofonico alla BBC del gennaio 1952.

giovane Alan – a guardare delle margherite, che egli amava e che sono tra l'altro l'ultimo argomento su cui lavorava prima di morire.



1. Disegno di Sara Turing

Il riferimento ai bambini e all'educazione e all'apprendimento dei bambini è costante nella sua discussione delle macchine, della possibilità di far loro mostrare qualche forma di intelligenza e della loro educazione. Nella maturità Turing si convince che: «le potenzialità dell'intelligenza umana possono essere realizzate solo se è fornita un'educazione adatta»; Turing descrive vari metodi di insegnamento per le macchine non organizzate ispirandosi anche nella terminologia al processo educativo.

Turing compie esperimenti con macchine non organizzate (tutto sulla carta, ma prevede con ragione che le macchine non organizzate potranno essere simulate dalle macchine organizzate), sottoposte a due tipi definiti di interferenza, uno stimolo di piacere e uno stimolo di dolore, ottenuti aggiungendo una memoria su cui le interferenze



possono occorrere nella forma di scrittura dall'esterno; gli riesce di trasformare una macchina non organizzata in una macchina universale; ma non ritiene soddisfacente il risultato, perché non corrisponde a quello realistico dell'educazione dei bambini: la macchina è stata fatta girare a lungo con continue applicazioni di dolore, troppo numerose.

Nello stesso tempo, per quel che riguarda le macchine esistenti, Turing coltiva l'idea di un apprendimento basato, attraverso la comunicazione, su un'educazione più simile a quella della scuola che non il sistema piacere-dolore.

Man mano che va avanti, Turing parla sempre più delle sue macchine con lo stesso affetto che se parlasse davvero di bambini. Una comunicazione diretta potrebbe essere realizzata con qualche forma di linguaggio simbolico; ci potrebbe essere nella macchina anche un sistema completo di logica. Le comunicazioni esterne in forma dichiarativa potrebbero essere classificate come ordini, e il sistema potrebbe avere una premessa secondo cui tutto quello che dice l'insegnante è vero. Ci potrebbero essere anche imperativi espressi nel sistema, non facenti parte delle regole del sistema, che dicono di non fare certe cose a meno che non siano state menzionate dall'insegnante, per *default* – altro argomento che diventerà un classico, quello del *default*, insieme a quello delle euristiche da introdurre per governare l'applicazione delle regole logiche, in modo da fare diventare buoni ragionatori.

Un altro esempio del modo di procedere di Turing è fornito dal suo interesse per il cervello.

Nel 1943, mentre lavora al controspionaggio a Bletchley Park, Turing confessa al collaboratore Donald Bayley la sua ambizione di «voler costruire un cervello». Da allora l'impegno di Turing è teso a raccogliere tutti gli elementi che possono servirgli. Per esempio lo troviamo spesso impegnato, anche su sollecitazioni altrui (o per rispondere a obiezioni pregiudiziali sulle dimensioni necessarie per una macchina pensante), in calcoli sul numero di neuroni, sul numero di dendriti e connessioni, sulla crescita dei neuroni. In una lettera a J.Z. Young dell'8 febbraio 1951, nel confessare che sta perseguendo gli studi sull'embriologia per la soddisfazione che gli dà l'ottenere ivi abbastanza facilmente risultati, spiega come l'argomento non sia senza connessioni con l'altro tema che gli sta a cuore: «la struttura del cervello deve essere tale da poter essere ottenuta dal meccanismo embriologico genetico, e spero che la teoria su cui sto ora lavorando [la matematica della morfogenesi] possa fare luce su quali restrizioni questo implichi veramente. Quello che mi dici sulla crescita dei neuroni sotto stimolo è molto interessante a questo riguardo. Suggerisce i mezzi con cui si potrebbe far crescere

i neuroni in modo da formare un particolare circuito, e non solo da essere collocati in una particolare posizione».

E tuttavia (nella stessa lettera) «temo di essere lontano dallo stadio in cui mi sento portato a porre questioni anatomiche. Secondo la mia idea di come procedere questo non succederà fino a una fase molto più avanzata quando avrò una ben definita teoria su come vengono fatte le cose».

In assenza di una teoria globale, che non è però neanche cercata nel primo stadio, una teoria locale viene articolata nel seguente modo: ipotizzare modalità di funzionamento del cervello, riconoscerne il carattere meccanico e quindi sperimentarle programmandole.

Questo modo di procedere è illustrato dalla breve discussione dell'analogia. Nel dibattito radiofonico sopra citato, R.B. Braithwaite sostiene che una macchina non può riconoscere similarità quando non c'è nulla nel suo programma che dica quali sono le similarità che ci si aspetta riconosca, mentre può benissimo riconoscere analogie quando queste siano state già formalizzate in un modello matematico. Turing si dice invece convinto di poter fare una macchina che riconosca un'analogia; e considera il caso della doppia negazione, quei «non-non» che proprio non vogliono entrare in testa, finché non viene proposta l'immagine dell'attraversamento della strada, e allora qualcosa scatta. *This remark just clinched it.* Egli immagina allora cosa può avvenire nel cervello; una sorta di economia permessa quando due idee hanno la stessa trama di connessioni logiche, e il cervello può usarne una stessa più volte per memorizzare la struttura. Finché si parla di «non» e «non-non» sembra che le parole non arrivino alla parte giusta del cervello; con l'immagine dell'attraversamento, ci arrivano, ma per una strada diversa da quella diretta. Turing evita il problema delle somiglianze, spostando l'attenzione sulla struttura logica dell'argomento.

Secondo Turing l'analogia sarebbe un'imposizione dell'architettura del cervello, dovuta alla necessità di economizzare. Aveva già detto in una lettera a Jack – I. John Good – del 18 settembre 1948 (segno che da tempo ci stava pensando) che «non vedo tanto il cervello “in cerca di analogie” quanto piuttosto che le analogie gli siano imposte dalle sue stesse limitazioni».

Se ora si potesse dare una spiegazione puramente meccanica di questo processo, lo si potrebbe far fare a una macchina. Ma si noti che il carattere meccanico non è provato dal programma; il programma è la conseguenza finale della comprensione del funzionamento del cervello (illustrata dalla sua ipotesi sull'analogia). La posizione di Turing non è il funzionalismo che poi gli è stato attribuito.

Con «funzionalismo alla Turing» si intende in filosofia della mente la tesi che, detto in modo semplicistico, il cervello è una macchina digitale universale, oppure che questa è un modello adeguato del cervello, o più correttamente che se si riesce a programmare un calcolatore in modo che esso mostri una specifica competenza mentale, allora questa competenza è spiegata. Il funzionalismo alla Turing è anche identificato con la tesi forte dell'Intelligenza Artificiale, tesi che è stata inventata da John R. Searle che l'ha collegata al test di Turing: questo sarebbe per l'Intelligenza Artificiale un criterio scientifico per stabilire il successo o fallimento dell'impresa di creare menti scrivendo programmi, impresa che Searle si è proposto di confutare. Il criterio affermerebbe: se un calcolatore riesce a comportarsi in modo tale che un esperto non sia in grado di distinguere il suo comportamento da quello di un essere umano che possieda una certa capacità cognitiva – per esempio la capacità di fare le addizioni o di capire la lingua cinese – allora anche il calcolatore possiede questa capacità. L'Intelligenza Artificiale forte, secondo Searle, si pone come scopo quello di scrivere programmi per determinate capacità, con l'idea di realizzare così non solo modelli della mente, ma vere e proprie menti<sup>6</sup>.

Searle ha costruito la sua fortuna su una ventennale discussione del test di Turing, argomentando che se una macchina fatta di lattine di birra fosse in grado di condurre una conversazione in cinese, non per questo diremmo che conosce il cinese. Turing risponderebbe – crediamo – che se una macchina del genere fosse costruita, essa proverebbe che chi ha realizzato l'elenco delle regole del cinese ha fatto un buon lavoro. Si noti che Turing parlava di un interrogante medio, Searle di un esperto.

Turing non appare minimamente interessato alla prospettiva descritta da Searle, anzi la sua è letteralmente rovesciata. Il suo motto è: quando capiremo, potremo scrivere un programma, e non: se scriviamo un programma capiremo.

Dal contesto della discussione sopra riportata si evince che con «carattere meccanico» di una modalità di funzionamento del cervello Turing intende piuttosto una spiegazione fisica in senso lato, o una comprensione scientifica, perché sta discutendo con un neurologo e considera possibili spiegazioni in termini di attività neuronale. Quan-

---

<sup>6</sup> J.R. Searle, "Minds, Brains and Programs", *Behavioral and Brain Sciences*, 3, 1980, pp. 417-58; trad. it. con integrazioni del 1990 in J.R. Searle, "La mente è un programma?", in *Mente e macchina*, Quaderno Le Scienze, 66, giugno 1992, pp. 5-10.

do il neurologo lo avverte che non c'è allo stato attuale una spiegazione in termini di cellule e connessioni di fibre nel cervello, Turing non si lascia demoralizzare e si dice convinto che già lo si possa fare, solo che se lo facesse poi si direbbe che quello non è vero pensiero (osserva ironicamente: il pensiero sembra essere solo ciò che non capiamo come avviene). Turing non si adagia nel riduzionismo biologico e nell'attesa passiva, ritiene invece che si possa individuare un livello di analisi formale che presuppone l'instaurarsi di connessioni stabili, o circuiti, nei neuroni. La biologia gli serve solo come conferma della possibilità fisica delle ipotesi elaborate su questo presupposto.

Se si vuole enucleare una 'tesi di Turing', questa può essere formulata, nel modo più fedele al suo spirito, come la certezza che tutto ciò che è spiegabile in base alle leggi scientifiche è riproducibile su un calcolatore.

La tesi che tutte le leggi scientifiche abbiano una natura computazionale ha un'enorme conferma sperimentale<sup>7</sup>. Si consideri il fatto curioso che Roger Penrose, recente campione della non computabilità del pensiero, dà ragione a Turing, in un suo modo contorto. L'argomento di Penrose, è infatti il seguente: siccome certi fenomeni non sono computabili, essi non rientrano nelle attuali leggi fisiche, e quindi c'è bisogno per spiegarli di trovare leggi fisiche di una natura diversa da quella attuale, in una nuova elaborazione della teoria quantistica<sup>8</sup>. A prescindere dal successo o meno della proposta di Penrose, il **contrapposito** della sua premessa non è altro che la tesi di Turing, ed è indipendente dal fatto che esistano fenomeni non computabili: Turing avrebbe talmente ragione che per capire e spiegare un fenomeno non computazionale bisogna cambiare la natura delle leggi fisiche.

La fiducia nella scienza dura si ritrova nel contributo di Turing alla morfogenesi, troppo poco noto al di fuori della cerchia dei biologi matematici. In questo caso si può dire che Turing rifiutò di ascoltare e subire se stesso e di lasciarsi influenzare dalle competenze e dal modo di affrontare i problemi che aveva perfezionato nella precedente fase

---

<sup>7</sup> Gli unici risultati che gettano qualche dubbio sulla tesi sono quelli di Marian Pour-El su alcune soluzioni non calcolabili (in funzione dei dati iniziali) di equazioni differenziali, ma la loro interpretazione è dubbia. Per l'ampia evidenza positiva relativa alle equazioni classiche della fisica matematica invece si veda M. Pour-El e J.I. Richards, *Computability in Analysis and Physics*, Springer, Berlin, 1989.

<sup>8</sup> R. Penrose, *The Emperor's New Mind*, Oxford Univ. Press, Oxford, 1989; trad. it. *La nuova mente dell'imperatore*, Adelphi, Milano, 1990.

della sua carriera; con l'esperienza che aveva avuto con le macchine, si sarebbe potuto pensare che egli portasse nello studio della forma l'idea dello sviluppo di un organismo secondo un programma. La scoperta del codice della doppia elica è di due anni dopo, si può dire che era nell'aria, e si inseriva come conferma nella sintesi di darwinismo e mendelismo che era appena stata compiuta in quegli anni. Daniel Dennett ha cercato di utilizzare la nozione di algoritmo per presentare l'idea di evoluzione, credendo forse di fare quello che pensa avrebbe dovuto fare Turing<sup>9</sup>. Ma l'intenzione di Turing nel dedicarsi alla morfogenesi era tutt'altra, al contrario, era stata, come ha confessato a Robin Gandy, quella di sconfiggere l'argomento dell'orologiaio («the argument from design»).

Il vecchio argomento di William Paley era stato confutato naturalmente da Darwin, ma Turing lo vedeva risorgere nella volontà di spiegare tutto con l'evoluzione: ogni variazione si deve collegare con i vantaggi evolutivi e la selezione naturale sostituisce il creatore. Invece per Turing la forma organica doveva essere spiegata come la forma inorganica, con la fisica e la chimica. La forma delle foglie non doveva essere ascritta ai vantaggi che portava, un risultato di adattamento, ma dipendere dalla chimica e fisica della sua crescita. Anche D'Arcy Thompson nel 1917 aveva dato indicazioni simili, ma Turing costruisce un modello matematico.

Il modello utilizza matematica classica, equazioni differenziali, e la discussione che lo accompagna, delle sue ipotesi e dei suoi limiti, della linearizzazione, è un classico della modellistica matematica; in questo lavoro Turing introduce i concetti di diffusione-reazione e di equilibrio di Turing, che sono un contributo perenne alla biologia matematica.

Nelle discussioni preliminari di un problema, al posto di esaminare la letteratura esistente, Turing preferisce costruire analogie, con esempi tratti dalla vita quotidiana. Questa propensione si ritrova in ogni settore delle sue ricerche. Quando imposta i primi linguaggi di programmazione, per descrivere l'iterazione condizionata usa l'immagine dell'aeroplano che gira sull'aeroporto in attesa di avere il permesso di atterraggio; le istruzioni del linguaggio che diventerà l'*assembler* le chiama in *popular form*; quando discute dell'instabilità negli studi di morfogenesi concepisce un'immagine molto convincente sulla

---

<sup>9</sup> D.C. Dennett, *Darwin's Dangerous Idea*, Simon&Schuster, New York, 1991; trad. it. *L'idea pericolosa di Darwin. L'evoluzione e i significati della vita*, Bollati Boringhieri, Torino, 1997.

possibilità che l'equilibrio instabile si rompa per piccole interferenze casuali: «Immaginiamo una corda appesa poco sopra il suo baricentro, che cade dritta come un filo a piombo, e supponiamo che un topo incominci ad arrampicarsi lungo la corda: inizialmente vi sono solo vibrazioni, ma man mano che il topo sale succede che il filo incomincia a oscillare».

Lo stile letterario di Turing non è particolarmente elegante, il lessico non è ricco, la sintassi forzata; ma i suoi scritti sono illuminati dalle immagini sopra ricordate, e riescono sempre a esprimere in modo piano anche le questioni più difficili.

Nel 2009 il primo ministro inglese Gordon Brown ha pubblicamente chiesto scusa a Turing (e sia pure di sfuggita alle altre migliaia di omosessuali) per il trattamento indegno subito in base alla legge del tempo<sup>10</sup>. Nella formulazione delle scuse, due elementi disturbano, e mostrano come sia ancora inadeguata la comprensione della figura di Turing. Da una parte Mr Brown insiste soprattutto sulla dovuta riconoscenza per il contributo da lui dato alla fine anticipata della guerra, e alle migliaia di vite così risparmiate, ignorandone la grandezza scientifica. Dall'altra, collegata, Mr Brown non ha capito che è a noi tutti, all'umanità che la Gran Bretagna deve chiedere scusa per averci sottratto un simile pensatore nel pieno della maturità e della creatività scientifica<sup>11</sup>.

GABRIELE LOLLI

---

<sup>10</sup> <http://blog.jgc.org/2011/07/complete-text-of-gordon-browns-apology.html>.

<sup>11</sup> Per quel che riguarda il suicidio di Turing, a seguito della condanna alla castrazione chimica, resta aperto il sospetto che dopo la condanna per omosessualità egli, in possesso di quelli che erano ritenuti segreti militari, fosse ritenuto una persona pericolosa per lo Stato, in quanto ricattabile.

# Perenni indecisioni

---

## *L'homme machine*

Benché il progetto di una completa automatizzazione o meccanizzazione del pensiero umano sia relativamente recente, già nell'antichità erano state poste le basi del concetto di *algoritmo* che lo sottende. Il nome deriva da Muhammad ibn Musa al-Khwarizmi, un matematico arabo della prima metà del secolo IX «originario di Khwarizm» (l'odierna Khiva, in Asia Centrale), il cui libro *Hisab al-jabr w'al-muqabala*, «Calcoli mediante costruzione e riduzione», diede anche il nome all'algebra. I primi esempi di algoritmi sono invece molto più antichi, e risalgono ai primordi della matematica. Sia la formula algebrica babilonese per la risoluzione dell'equazione di secondo grado, sia le costruzioni geometriche greche con riga e compasso, posseggono infatti quelle proprietà di calcolabilità e costruibilità presenti nel titolo del libro di al-Khwarizmi, e caratteristiche della nozione di algoritmo.

La storia più recente della matematica fornisce anche i primi esempi di risultati negativi, del tipo di quello che troveremo per la logica predicativa: l'impossibilità di risolvere equazioni di grado superiore al quarto mediante radicali, e di quadrare il cerchio mediante riga e compasso. Anche se, in realtà, è la dimostrazione dell'irrazionalità di  $\sqrt{2}$  la madre di tutti i risultati di impossibilità.

Passando dalla matematica al pensiero puro, il trattamento sistematico delle definizioni nei *Topici* e dei sillogismi negli *Analitici Primi* è un primo abbozzo di un metodo di decisione, esplicitamente dichiarato in questi ultimi (45b): «tutti i sillogismi possono svilupparsi in questo modo, e soltanto in questo».

Il primo a credere coscientemente che tutto il pensiero si potesse in qualche modo ridurre ad algoritmi fu però Raimondo Lullo. Nel 1274 la sua opera principale, l'*Ars Magna*, propose un metodo 'logico' basato sull'individuazione di una serie di termini semplici, dalla cui combinazione dovevano risultare tutte le verità attingibili dall'intelletto umano.

I dettagli del progetto erano ovviamente insensati: ad esempio, i termini semplici risultavano essere nove predicati assoluti, nove predicati relativi, nove questioni, nove soggetti, nove virtù e nove vizi. Altrettanto (se non più) insensata era l'idea che la realizzazione del progetto avrebbe fornito un aiuto per la conversione degli infedeli: i quali, com'era logico, lo lapidarono nel 1315.

Le idee di Lullo invece non morirono e furono riprese entusiasticamente nel Rinascimento, pur finendo altrettanto tragicamente. Basti ricordare il caso del suo seguace Giordano Bruno, autore di varie opere lulliane, che fu arso vivo nel 1600: questa volta dai fedeli, per *par condicio*.

La concezione algoritmica del pensiero fu riproposta in maniera meno caricaturale nel 1655 da Thomas Hobbes, nella *Computatione sive logica*, «Il calcolo, ovvero la logica» (1, 2):

Per ragionamento intendo il calcolo, che consiste nel riunire insieme più cose per farne una somma o nel sottrarre una cosa dall'altra per conoscerne il resto. Ragionare è la stessa cosa che addizionare e sottrarre, e ogni ragionamento si riduce a queste due sole operazioni mentali.

Lullo e Hobbes furono dichiaratamente gli ispiratori del *De arte combinatoria*, in cui Leibniz prefigurò nel 1666 il metodo di aritmetizzazione sviluppato da Gödel nel 1931<sup>1</sup>. Ma per la discussione sulla natura algoritmica del pensiero è più importante uno scritto minore<sup>2</sup>, nel quale Leibniz avanzò alcune idee che troveranno la loro naturale applicazione nel corso di questo articolo.

In particolare, egli intuì che un *calculus ratiocinator*, «calcolo del ragionamento», permetterebbe di ridurre la complessità del pensiero alla semplicità delle quattro operazioni:

Quando sorgeranno delle controversie, non ci sarà maggior bisogno di discussione fra due filosofi di quanto ce ne siano fra due calcolatori. Sarà sufficiente, infatti, che essi prendano la penna in mano, si siedano a tavolino, e si dicano reciprocamente (chiamando, se vogliono, a testimone un amico): *Calculemus*.

---

<sup>1</sup> K. Gödel, "Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I", *Monatshefte für Mathematik und Physik*, 38, 1931, pp. 173-98.

<sup>2</sup> G. Leibniz, "De scientia universalis seu calculo philosophico", in *Philosophische Schriften*, ed. C.I. Gerhardt, vol. 7, Berlin 1875-1890, pp. 198-203.



Leibniz era però conscio dei limiti del pensiero umano, e li espresse nella famosa distinzione tra *verità di ragione* e *verità di fatto*:

La differenza è la stessa che intercorre fra i numeri razionali e quelli irrazionali. Come nei numeri irrazionali la scomposizione procede all'infinito, allo stesso modo le verità di fatto richiedono un'analisi infinita, che solo Dio può effettuare.

Le verità di ragione sono ovviamente quelle della logica e della matematica, e le verità di fatto quelle della scienza. Per arrivare a conoscerle, Leibniz riteneva che nel primo caso fosse sufficiente un'*ars judicandi*, «arte della verifica», mentre nel secondo fosse necessaria un'*ars inveniendi*, «arte della scoperta». La distinzione è fondamentale, anche se vedremo in seguito che in realtà il confine che separa le due arti è più arretrato di quanto Leibniz pensasse.

La sua intuizione che la logica sillogistica cadesse sotto il dominio dell'*ars judicandi* era però corretta, così come andavano nella giusta direzione i suoi abbozzi di calcoli per renderla algoritmica, basati su operazioni proposizionali o insiemistiche. Attraverso le opere di Eulero, Gergonne e Boole i calcoli abbozzati da Leibniz divennero poi una realtà, e oggi l'algebra booleana costituisce l'algoritmo paradigmatico per la decisione delle «leggi del pensiero» proposizionali e sillogistiche.

Con questi precedenti, non stupisce che nel 1928 Hilbert e Ackermann<sup>3</sup> proponessero il cosiddetto *Entscheidungsproblem*, «problema della decisione», per la nuova logica: stabilire, cioè, l'esistenza o meno di un algoritmo per determinare la validità delle formule predicative.

Se tale algoritmo esistesse, per risolvere positivamente il problema basterebbe esibirne uno, come si fa facilmente per le formule proposizionali e sillogistiche. Poiché invece non esiste, per risolvere negativamente il problema bisognerà scartare tutti gli infiniti algoritmi possibili (passati, presenti e futuri): questo richiede una preventiva caratterizzazione degli algoritmi, alla quale ora ci dedichiamo.

### *La machine homme*

Accanto alla storia alla quale abbiamo appena accennato, riguar-

---

<sup>3</sup> D. Hilbert, W. Ackermann, *Grundzüge der theoretischen Logik*, Springer Verlag, Berlin, 1928.

dante la ricerca e la scoperta di algoritmi per vari aspetti del pensiero umano, ne esiste una complementare riguardante la progettazione e la costruzione di «macchine» alle quali delegare l'implementazione di questi algoritmi. Le due storie costituiscono i binari paralleli del 'soffice' *software* e del 'duro' *hardware*, sui quali corre oggi il treno dell'informatica.

Come già nel caso degli algoritmi, i primi ausili artificiali per il calcolo risalgono all'antichità. L'abaco è il più noto e fu usato in varie forme, a partire dai Greci e fino a tempi recenti, ma ne esistevano molti altri: dalle cordicelle annodate già citate nel *Libro di Ezechiele* (XL, 3), e la cui versione più nota sono i *quipu* incaici, al sistema di bastoncini che rimase in vigore in Cina per più di duemila anni.

Tralasciando 'prototipi' quali il meccanismo di Antikythera, la prima vera 'macchina calcolatrice' fu inventata da Wilhelm Schickard, che in una lettera del 20 settembre 1623 a Keplero riportava:

Ciò che lei ha fatto in maniera logica, io ho cercato di fare in maniera meccanica. Ho costruito una macchina consistente di undici ruote dentate complete e sei incomplete che può calcolare. Lei scoppierebbe a ridere se vedesse come si muove da una colonna di decine alla successiva, o prende a prestito da esse durante la sottrazione.

Nel 1642 anche Blaise Pascal costruì una macchina a ruote dentate per fare somme e sottrazioni, la cosiddetta *pascalina*, e riuscì a elaborarne una cinquantina di versioni prima di deviare il suo talento scientifico sulla teologia. Nei successivi tre secoli la costruzione di macchine in grado di fare un numero sempre maggiore di operazioni aritmetiche prosperò, a cominciare dalla macchina a rulli introdotta verso il 1671 da Leibniz per fare i prodotti.

La discontinuità evolutiva che costituì il vero salto di qualità di questo sviluppo fu il cosiddetto *calcolatore universale*, abbozzato in versione a vapore da Babbage<sup>4</sup> nel 1837, e progettato in versione elettr(on)ica da Turing<sup>5</sup> nel 1936.

La scoperta fondamentale di Babbage e Turing fu che, non appena una macchina raggiunge la massa critica che le permette di decodifi-

---

<sup>4</sup> C. Babbage, "On the mathematical powers of the Calculating Engine", in *The origins of digital computers*, ed. B. Randell, Springer Verlag, Berlin, 1973, pp. 19-54.

<sup>5</sup> A. Turing, "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, 42, 1936, pp. 230-65.

care istruzioni codificate numericamente e simularle passo passo, essa diventa in grado di eseguire non soltanto i compiti per i quali è stata costruita, ma qualunque compito codificabile da un insieme finito di istruzioni. In termini più moderni, essa cessa di essere una macchina specificamente dedicata e diventa universalmente programmabile.

Le tecniche per la progettazione di un calcolatore universale non sono altro che quelle dell'aritmetizzazione, già usate da Kurt Gödel nel 1931 per la dimostrazione del suo teorema. Più precisamente:

- i *dati* di entrata e di uscita si codificano come se fossero dei numeri
- le *istruzioni* come se fossero regole di un sistema
- i *programmi* come se fossero sistemi di regole
- e i *calcoli* come se fossero dimostrazioni.

In tal modo si può tradurre in maniera aritmetica la proprietà « $C$  è un calcolo del risultato  $r$  a partire dai dati  $d$  e dal programma  $p$ ».

Possiamo allora concludere che *sotto certe ipotesi, un calcolatore aritmetico è universale*. Le ipotesi riguardano la capacità di calcolare la proprietà precedente, e quindi di cercare (e trovare, se c'è) il risultato di un calcolo a partire da qualunque dato e qualunque programma: di qui la qualifica di «universalità». Se poi si conviene di identificare la nozione di «algoritmo» con quella di *operazione programmabile*, allora il calcolatore risulta anche in grado di simulare qualunque algoritmo.

Gli argomenti a favore della definizione di algoritmo appena data sono sostanzialmente tre. Anzitutto, il fatto che alla nozione di computer universale si sia arrivati mediante un'analisi del tipo di processi coinvolti nelle tipiche operazioni di calcolo umano. Inoltre, la sorprendente equivalenza di tutte le altre definizioni della nozione di algoritmo proposte nel corso degli anni (tra le quali il  $\lambda$  calcolo, al quale accenneremo in seguito). E infine, la constatazione che tutti gli algoritmi noti risultano effettivamente programmabili.

Il calcolatore universale risolve in maniera definitiva il problema del calcolo numerico, essendo appunto in grado di calcolare tutto ciò che è algoritmicamente calcolabile. Per quanto riguarda invece l'automazione del pensiero, già nell'*Ars Magna* Lullo aveva proposto un meccanismo a ruote concentriche sulle quali stavano scritti i termini semplici. Girando le ruote, si potevano ottenere automaticamente tutte le combinazioni che corrispondevano alle «verità» del sistema.

La prima seria automatizzazione di una parte del pensiero fu permessa dagli algoritmi per la logica proposizionale e sillogistica forniti dall'algebra booleana, che risultarono di facile implementazione: la prima realizzazione meccanica fu prodotta da William Jevons nel 1870, e la prima elettrica da Allan Marquand nel 1885.

Viceversa, nel 1938 Shannon<sup>6</sup> mostrò che l'algebra booleana fornisce il paradigma per lo studio dei circuiti elettrici. Rappresentando infatti la presenza o l'assenza di corrente elettrica mediante i valori booleani 1 e 0, corrispondenti a «vero» e «falso», un circuito diventa semplicemente la realizzazione fisica di una tavola di verità. Il teorema di completezza vero-funzionale della logica proposizionale mostra allora che ogni circuito può essere sintetizzato mediante una combinazione di «porte logiche» corrispondenti ai connettivi.

Analogamente, nel 1943 McCulloch e Pitts<sup>7</sup> mostrarono che l'algebra booleana fornisce un paradigma simile per lo studio dei circuiti neuronali, in cui questa volta 1 e 0 rappresentano la presenza o l'assenza di uno stimolo nervoso, e i connettivi particolari collegamenti sinaptici in grado di sintetizzare qualunque circuito neuronale. L'applicazione pratica più importante di questi sviluppi fu certamente la costruzione, a partire dal 1945, del moderno *computer* da parte di von Neumann<sup>8</sup>, sulla base del progetto di Turing. Le istruzioni di un programma, che stabiliscono che in certe condizioni un calcolatore deve eseguire una certa operazione, sono infatti facilmente esprimibili mediante formule proposizionali, e dunque sintetizzabili mediante circuiti elettrici. Sintetizzando il programma di un calcolatore universale si ottiene appunto un computer, cioè una macchina programmabile in grado di eseguire qualunque algoritmo.

L'analogia tra circuiti elettrici e neuronali permette di pensare metaforicamente al cervello come a un *computer biologico* e al computer come a un *cervello elettronico*, ma spesso queste metafore sono state intese, o fraintese, in un senso più o meno letterale. Da un lato, il *funzionalismo* di Putnam<sup>9</sup> ha infatti proposto alle scienze cognitive una teoria del cervello come *hardware* e della mente come *software*. Dall'altro lato, l'*Intelligenza Artificiale* di Turing<sup>10</sup> ha proposto all'informatica una pratica di simulazione dell'intero pensiero umano da parte del computer.

<sup>6</sup> C. Shannon, "A symbolic analysis of relay and switching circuits", *Transactions of the American Institute of Electrical Engineers*, 57, 1938, pp. 713-23.

<sup>7</sup> W. McCulloch e W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, 5, 1943, pp. 115-33.

<sup>8</sup> J. von Neumann, "First draft of a report on the edvac", in *From eniac to univac*, ed. N.B. Stern, Digital Press, Bedford, Mass., 1981, pp. 177-246.

<sup>9</sup> H. Putnam, "On the nature of mental states", in *The nature of mind*, ed. D.M. Rosenthal, Oxford University Press, Oxford, 1967, pp. 197-203.

<sup>10</sup> A. Turing, "Computing machinery and intelligence", *Mind*, 59, 1950, pp. 433-60.

Oggi il funzionalismo è tornato a essere una semplice metafora, essendo stato abbandonato dallo stesso Putnam, ma l'Intelligenza Artificiale continua a essere presa seriamente, grazie anche ad alcune parziali ma appariscenti realizzazioni. Anzitutto i *sistemi esperti*, che condensano in programmi le capacità di giudizio degli specialisti in vari campi, dalla diagnostica medica alla consulenza giuridica. E poi i *proof checkers*, «verificatori di dimostrazioni», e i *theorem provers*, «dimostratori di teoremi», che realizzano versioni informatiche delle leibniziane *ars judicandi* e *ars inventiendi*.

A ispirare scetticismo sull'Intelligenza Artificiale è invece l'osservazione che il progetto si ispira sostanzialmente a un doppio e improbabile riduzionismo: da un lato, dell'emisfero destro dell'intuizione all'emisfero sinistro della ragione, e dall'altro lato, dell'elettrochimica del cervello all'elettronica dei circuiti. Poiché i successi nella simulazione del pensiero meccanico e gli insuccessi nella simulazione del pensiero creativo (o anche solo della sensorialità e del senso comune) erano entrambi prevedibili, non vale dunque la pena di esaltarsi per i primi o deprimersi per i secondi.

### *Fermi tutti*

Per trovare cose che i computer non possono fare non è comunque necessario scomodare le difficoltà di realizzazione del progetto dell'Intelligenza Artificiale: basta notare l'analogia fra programmi e sistemi formali, e immaginare che i teoremi di limitatezza degli uni si possano riprodurre in qualche forma per gli altri.

Già nel suo lavoro originario del 1936 Turing ebbe l'idea di applicare l'argomento di Russell, relativo agli insiemi e alla negazione, ai programmi aritmetici e a un'appropriata versione della negazione: ad esempio, all'operazione  $\sim$  che ferma un programma che non si fermava, e fa girare indefinitamente un programma che invece si fermava. L'obiettivo era di riprodurre la contraddizione del paradosso di Russell, dimostrando così che l'operazione  $\sim$  non è programmabile.

Il paradosso in questione è il più famoso della logica moderna, e fu comunicato da Russell a Frege in una storica lettera del 16 giugno 1902<sup>11</sup>:

---

<sup>11</sup> B. Russell, "Letter to Frege", in *From Frege to Gödel*, ed. J. van Heijenoort, Harvard University Press, Harvard, 1967, pp. 124-5.

Caro collega,

da un anno e mezzo sono a conoscenza dei suoi *Principi dell'aritmetica*, ma solo ora ho potuto trovare il tempo per lo studio dettagliato che intendevo fare del suo lavoro. Mi trovo completamente d'accordo con lei su tutto l'essenziale. [...]

C'è solo un punto in cui ho incontrato una difficoltà. Lei dice che una funzione può fungere da elemento indeterminato. Anch'io lo credevo, ma ora la cosa mi sembra dubbia a causa della seguente contraddizione. Sia  $R^{12}$  il predicato: essere un predicato che non può essere predicato di se stesso. Può  $R$  essere predicato di se stesso? Da ciascuna risposta segue il suo contrario. Dobbiamo dunque concludere che  $R$  non è un predicato. Analogamente, non c'è una classe di tutte le classi che non appartengono a se stesse. Ne concludo che, in certe circostanze, una collezione definibile non forma una classe.

In altre parole, se

$$R = \{x : \neg (x \in x)\},$$

allora

$$x \in R \Leftrightarrow \neg (x \in x),$$

e

$$R \in R \Leftrightarrow \neg (R \in R).$$

Come già aveva osservato Gödel nel caso del suo teorema, nel caso dei programmi  $p$  e dell'operazione  $\sim$  non si può procedere per pura analogia con l'argomento di Russell, ponendo semplicemente

$$T(p) = \sim (p(p)),$$

in modo da ottenere

$$T(T) = \sim (T(T)),$$

perché scrivere  $T(p)$  e  $p(p)$  non ha senso: in entrambi i casi stiamo infatti lavorando con programmi aritmetici, i quali si possono applicare

---

<sup>12</sup> Russell usava  $w$  al posto di  $R$ .

a numeri, ma non ad altri *programmi*. In altre parole, non è possibile dimenticare la distinzione fra uso e menzione per i programmi.

Si può però associare a ogni programma un numero, chiamando  $\bar{p}$  il posto in cui il programma  $p$  sta in un qualunque ordinamento di tutti i programmi con un argomento<sup>13</sup>: si distingue così il programma  $p$  dal suo numero  $\bar{p}$ . E mentre non si poteva scrivere  $p(p)$ , perché un programma non si può applicare a un altro programma, in particolare a se stesso, si può scrivere  $p(\bar{p})$ , perché un programma si può applicare a un qualunque numero, in particolare al suo.

A questo punto possiamo immediatamente riformulare la definizione precedente in modo sensato, cioè

$$T(\bar{p}) = \sim (p(\bar{p})),$$

e ottenere

$$T(\bar{T}) = \sim (T(\bar{T})).$$

Poiché  $T(\bar{p})$  si ferma se e solo se il programma  $p$ , applicato al suo numero, non si ferma, allora  $T(\bar{T})$  si ferma se e solo se il programma  $T$ , applicato al suo numero, non si ferma. Ma il programma  $T$ , applicato al suo numero, è  $T(\bar{T})$  stesso. Dunque,  $T(\bar{T})$  si ferma se e solo se non si ferma.

La contraddizione mostra, come annunciato, che l'operazione  $\sim$  non è programmabile. Ma poiché l'unica difficoltà per la sua programmazione sta nel decidere se un dato programma si ferma o no su un dato input, abbiamo così dimostrato il **teorema di Turing**: *non esiste nessun programma che sia in grado di decidere il problema della fermata di un dato programma su un dato input.*

Il teorema si può riformulare dicendo che *un calcolatore universale è incompleto*, nel senso che non può decidere tutti i problemi riguardanti il proprio funzionamento.

Più in particolare, benché un calcolatore universale possa simulare passo per passo qualunque programma su qualunque argomento, localmente, non può decidere *a priori* se tali simulazioni porteranno o no a un risultato, globalmente.

---

<sup>13</sup> Ad esempio, in base alla lunghezza (prima i più corti, cioè con meno simboli) se di lunghezza diversa, e in ordine lessicografico se della stessa lunghezza.

*Anche la Chiesa dice la sua*

A questo punto abbiamo gli strumenti necessari per risolvere negativamente l'*Entscheidungsproblem* di Hilbert e Ackermann. Il teorema di Turing ha infatti, come conseguenza immediata, il **teorema di Church**<sup>14</sup>: *la logica predicativa è indecidibile*, nel senso che non esiste nessun programma in grado di determinare se una formula è valida oppure no. Un tale programma potrebbe infatti decidere, in particolare, le istanze del problema della fermata, perché è possibile tradurre in modo naturale queste istanze sotto forma di formule logiche.

Le tecniche per la traduzione sono una versione logica dell'aritmetizzazione, che permette già di tradurre il problema della fermata del programma  $p$  sui dati  $d$  mediante la formula aritmetica «esistono  $C$  e  $r$  tali che  $C$  è un calcolo del risultato  $r$  a partire dai dati  $d$  e dal programma  $p$ ». Più precisamente:

- L' $n$ -esima cifra binaria di un *dato* si codifica come se fosse il valore di verità dell' $n$ -esima lettera proposizionale di una lista associata al dato in questione.
- Un'*istruzione*, come se fosse un'implicazione relativa alle lettere che codificano le cifre dei dati
- Un *programma*, come se fosse una congiunzione di istruzioni
- Una *configurazione*, come se fosse una congiunzione di dati
- E un *calcolo*, come se fosse una congiunzione di configurazioni di cui la prima è quella di entrata, ciascuna delle altre è ottenuta dalla precedente mediante un'istruzione, e l'ultima è quella di uscita.

Tecnicismi a parte, il motivo intuitivo dell'indecidibilità della logica predicativa è semplice: le regole logiche per la verità del quantificatore universale e per la falsità del quantificatore esistenziale richiedono la considerazione di tutti i termini del linguaggio, e l'alternanza di quantificatori ne può produrre infiniti. Nei casi invece in cui i termini sono soltanto un numero finito, gli alberi semantici rimangono anch'essi finiti e le relative formule si possono decidere. Questo succede, ad esempio, quando non ci sono funzioni e si vuole determinare la:

- *validità o soddisfacibilità di formule con soli predicati unari*
- *validità di formule del tipo  $\forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_m$*
- *soddisfacibilità di formule del tipo  $\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m$ .*

---

<sup>14</sup> A. Church, "A note on the Entscheidungsproblem", *Journal of Symbolic Logic*, 1, 1936, pp. 40-1.



In particolare, come ha notato Löwenheim nel 1915<sup>15</sup>, *la logica predicativa monadica è decidibile*. E poiché la logica sillogistica è una parte della logica monadica, anch'essa è decidibile: il confine fra decidibilità e indecidibilità segna dunque lo spartiacque fra la semplicità della logica greca e la complessità di quella moderna.

In generale, invece, *la logica predicativa è semidecidibile*. Di una formula valida, infatti, possiamo sempre determinare la validità, perché l'albero semantico ha tutti i rami contraddittori ed è finito. Di una formula non valida, invece, non sempre possiamo determinare la non validità, perché qualche ramo non contraddittorio dell'albero semantico può essere infinito. In altre parole, è possibile determinare tutte le risposte positive ma non tutte quelle negative.

Nei termini usati da Leibniz, mentre la validità di una formula predicativa è sempre una *verità di ragione*, che si può umanamente verificare mediante un procedimento finito, la non validità di una formula predicativa è a volte una *verità di fatto*, che solo Dio può verificare mediante un procedimento infinito. Il che significa, in particolare, che mentre per la verifica della validità di una formula proposizionale o monadica è sufficiente l'*ars judicandi*, per la verifica della validità di una formula predicativa arbitraria è necessaria l'*ars inveniendi*.

### *Un piatto al Curry\**

Benché la matematica moderna sia fondata sulla teoria degli insiemi, questa mal si addice alla programmazione e all'informatica, che trattano di funzioni. Naturalmente è possibile ridurre le funzioni ai loro grafi, che sono insiemi (di coppie argomento-valore), ma la cosa è obliqua e perversa. Più ovvio sarebbe cercare di dare alle funzioni una fondazione autonoma, e nel 1933 Church<sup>16</sup> ha proposto come soluzione una versione funzionale della teoria ingenua degli insiemi detta *lambda calcolo*.

Il principio di estensionalità si traduce in questo contesto dicendo che *due funzioni sono uguali quando hanno gli stessi valori per gli stessi argomenti*:

---

<sup>15</sup> L. Löwenheim, "Über Möglichkeiten im Relativkalkül", *Mathematische Annalen*, 76, 1915, pp. 447-70.

<sup>16</sup> A. Church, "A set of postulates for the foundation of logic (second paper)", *Annals of Mathematics*, 34, 1933, pp. 839-64.

$$f = g \Leftrightarrow \forall x (f(x) = g(x)).$$

Il principio di comprensione si traduce invece dicendo che *a ciascuna descrizione  $f(x)$  di valori corrisponde una funzione*, necessariamente unica per il principio di estensionalità, che si indica con

$$\lambda x. f(x).$$

La lettera  $\lambda$ , che dà il nome alla teoria, non ha nessun particolare significato: poiché nei *Principia Mathematica* Russell e Whitehead avevano usato la notazione  $\hat{x} . P(x)$  per la comprensione insiemistica, Church dapprima usò  $\lambda x . f(x)$  per la comprensione funzionale, e poi notò la somiglianza di  $\lambda$  con il lambda greco (maiuscolo).

Proseguendo nella traduzione, la relazione di appartenenza di un elemento a un insieme, definita nei *Principia Mathematica* dall'equivalenza

$$a \in \{x : P(x)\} \Leftrightarrow P(a),$$

diventa la relazione di applicazione a un argomento di una funzione, definita dall'uguaglianza

$$(\lambda x. f(x))(a) = f(a).$$

Quanto al paradosso di Russell, che aveva decretato la fine della teoria ingenua degli insiemi, si può immaginare che la sua traduzione decreterà la fine della teoria ingenua delle funzioni. Ma questa traduzione incontra un'ovvia difficoltà, legata al ruolo della negazione: essa è infatti un operatore logico, non insiemistico, e non è chiaro a quale operatore funzionale possa corrispondere.

Lasciando per un momento sospesa la questione, e chiamando  $n$  l'ipotetica versione funzionale della negazione, possiamo tradurre la definizione dell'insieme  $R$

$$x \in R \Leftrightarrow \neg(x \in x),$$

nella definizione di una funzione  $C$

$$C(x) = n(x(x)).$$

La paradossale equivalenza insiemistica

$$R \in R \Leftrightarrow \neg(R \in R),$$

diventa allora l'uguaglianza funzionale

$$C(C) = n(C(C)).$$

Ma questa, lungi dall'essere paradossale, esprime soltanto il fatto che la funzione  $n$  lascia invariato l'argomento  $C(C)$ . O, come si dice, che  $C(C)$  è un punto fisso di  $n$ . E poiché non abbiamo fatto alcuna ipotesi sulla funzione  $n$ , abbiamo in realtà dimostrato il **teorema di Curry**<sup>17</sup>: *ogni funzione del lambda calcolo ha un punto fisso*.

La cattiva notizia data dal teorema è che, poiché la negazione non ha punti fissi, non ne esiste nessuna versione all'interno del lambda calcolo: la logica non può dunque esservi rappresentata. L'aritmetica, invece, sì: come avevano infatti notato Peano<sup>18</sup> nel 1891 e Wittgenstein nel 1921 (*Tractatus*, 6.021), «il numero è l'esponente di un'operazione». Più precisamente, si può identificare il numero  $n$  con l'operazione

$$\underbrace{\lambda f. \lambda x. f \cdot \dots \cdot f(x)}_{n \text{ volte}} = \lambda f. \lambda x. f^{(n)}(x)$$

che, data una funzione  $f$  e un argomento  $x$ , itera la funzione  $n$  volte sull'argomento. E Turing<sup>19</sup> ha dimostrato nel 1937 che le funzioni definibili nel lambda calcolo rispetto a questa nozione di numero sono esattamente quelle numeriche programmabili nel suo senso: esse costituiscono la classe delle *funzioni ricorsive*, che la **tesi di Church e Turing** propone di identificare con la classe delle *funzioni calcolabili* in senso intuitivo.

La buona notizia data dal teorema è invece che ogni equazione funzionale ha almeno una soluzione nel lambda calcolo, esattamente come ogni equazione polinomiale ha almeno una soluzione nei numeri complessi. In altre parole, il teorema di Curry costituisce l'analogo

<sup>17</sup> H. Curry, "The inconsistency of certain formal logics", *Journal of Symbolic Logic*, 7, 1942, pp. 115-7.

<sup>18</sup> G. Peano, "Sul concetto di numero", *Rivista di Matematica*, 1, 1891, pp. 87-102 e 256-67.

<sup>19</sup> A. Turing, "Computability and  $\lambda$ -definability", *Journal of Symbolic Logic*, 2, 1937, pp. 153-63.

del teorema fondamentale dell'algebra, e fornisce un solido argomento interno a favore della tesi di Church e Turing appena enunciata, complementare all'argomento esterno della coincidenza di un gran numero di possibili formalizzazioni della nozione di funzione calcolabile, oltre alle due già citate.

Se da un lato l'incompatibilità del lambda calcolo con la logica ha segnato il fallimento della sua motivazione originaria, che era di fornire una fondazione funzionale della matematica analoga a quella insiemistica, dall'altro lato la sua compatibilità con la programmazione ne ha invece decretato il successo come naturale fondazione dell'informatica, e come paradigma dei linguaggi di programmazione funzionale esemplificati dal LISP<sup>20</sup>.

Naturalmente, come ci si può attendere dall'equivalenza tra funzioni definibili nel lambda calcolo e funzioni programmabili, un analogo del teorema di Curry vale anche per i programmi. Anzi, l'abbiamo già dimostrato senza neppure accorgercene: basta infatti considerare al posto della  $\sim$  una qualunque operazione programmabile, e ripetere la dimostrazione del teorema di Turing, per accorgersi che *ogni operazione programmabile ha un punto fisso*. E ottenere, tra parentesi, una giustificazione teorica della pratica della cosiddetta *programmazione ricorsiva*.

L'argomento del paradosso di Russell ha dunque rivelato ancora una volta (anzi, due) la sua fecondità, e una tabella permette di riassumere le metamorfosi che esso ha assunto nei teoremi di Turing e Curry, esibendone il perfetto parallelismo:

Russell 1902	Turing 1936	Curry 1942
$x \in R \Leftrightarrow \neg(x \in x)$	$T(p) = \sim(p(p))$	$C(x) = n(x(x))$
$R \in R \Leftrightarrow \neg(R \in R)$	$T(\bar{T}) = \sim(T(\bar{T}))$	$C(C) = n(C(C))$

PIERGIORGIO ODIFREDDI

---

<sup>20</sup> J. McCarthy, "Recursive functions of symbolic expressions and their computation by machine", *Communications of the Association for Computing Machinery*, 3, 1960, pp. 184-95.

# Alan Turing e il programma di ricerca dell'Intelligenza Artificiale

---

## 1. Dal test di Turing al programma di ricerca dell'IA

È riduttivo e per alcuni aspetti anche fuorviante incentrare sul cosiddetto «gioco dell'imitazione», noto anche come Test di Turing (TT), l'analisi dei contributi di Alan Turing alla nascita e allo sviluppo iniziale dell'Intelligenza Artificiale (IA)<sup>1</sup>. Anzitutto, la funzione intesa del TT sembra essere stata fin dall'inizio puramente divulgativa: Turing aveva l'obiettivo di raggiungere un ampio pubblico di persone colte, alle quali trasmettere e illustrare la possibilità tecnologica di sviluppare macchine capaci di elaborare strutture simboliche e di manifestare comportamenti intelligenti<sup>2</sup>. Inoltre, le regole proposte per la conduzione e il superamento del TT risultano essere molto vaghe, senza peraltro soddisfare i requisiti di intersoggettività per la valutazione dei risultati di un test empirico. È infine rilevante il dato di fatto che il TT non sia stato usato come *benchmark* o test empirico per i sistemi concretamente sviluppati dall'IA. L'obiettivo di superare il TT – qua-

---

<sup>1</sup> Come ormai è noto, Turing diede formulazioni del suo 'test' diverse tra loro sotto molti aspetti. Stando a quella più concisa, si tratta di riuscire a programmare una macchina (un calcolatore digitale *general purpose*) «che risponda alle domande in un modo tale che sarà estremamente difficile indovinare se le risposte sono state date da un uomo o da una macchina» (Turing 1951, trad. it. p. 24).

<sup>2</sup> «L'articolo del 1950 non si proponeva tanto di dare un contributo profondo alla filosofia, quanto di fare della propaganda. Turing riteneva che i tempi fossero maturi perché filosofi, matematici e scienziati prendessero sul serio l'idea che i calcolatori non sono solo macchine per calcolare, ma entità capaci di esibire comportamenti che dobbiamo considerare intelligenti. Cercò di diffondere questa convinzione e scrisse questo articolo – a differenza dei suoi articoli matematici – rapidamente e prendendoci gusto. Lo ricordo mentre mi leggeva dei brani – sempre con un sorriso sulle labbra, a volte trattenendo una risata. Alcuni commenti che ho avuto modo di leggere sovraccaricano questo articolo di significati rispetto alle intenzioni iniziali» (Gandy 1996, p. 125).

lunque cosa si intenda con ciò – è stato perseguito prevalentemente in manifestazioni socio-culturali, come il *Loebner Prize*<sup>3</sup>. Ciononostante, è piuttosto diffusa la strategia di incentrare proprio intorno al TT il discorso sul rapporto tra Turing e l'IA.

In questo lavoro seguiremo una diversa strategia per analizzare le relazioni tra Turing e l'IA, soprattutto in considerazione della sostanziale estraneità del TT allo svolgimento effettivo delle ricerche condotte nell'ambito dell'IA. Ecco in breve che cosa ci proponiamo di fare: partendo da una ricostruzione schematica dell'IA vista come un *programma di ricerca*, metteremo in evidenza i contributi di carattere modellistico, epistemologico, metodologico e anche tecnologico che Turing ha dato al suo sviluppo.

## 2. *L'IA come programma di ricerca*

L'analisi dell'IA come programma di ricerca che è proposta qui di seguito si colloca nel quadro di una famiglia di analisi metodologiche e di modelli epistemologici della scienza che sono stati sviluppati principalmente nel corso dell'ultimo cinquantennio. Una pietra miliare di questo percorso è costituito dalla pubblicazione di *The structure of scientific revolutions* di Thomas S. Kuhn, avvenuta nel 1962 (l'ultima edizione del cinquantenario è del 2012). Successivamente, Imre Lakatos e Larry Laudan hanno elaborato altri modelli epistemologici dei programmi di ricerca scientifica, che seguono per alcuni aspetti l'impostazione data da Kuhn, ma se ne distaccano notevolmente per altri aspetti (Lakatos 1970; Laudan 1977). Secondo questi autori, un programma di ricerca scientifico si sviluppa principalmente in funzione dei suoi successi e fallimenti, attraverso l'interazione di vari processi asincroni. I principali processi asincroni che si prendono in considerazione sono da un lato lo sviluppo di quella che noi chiameremo la *cornice* del programma di ricerca – cornice variamente analizzata e denominata dagli autori ricordati «paradigma», «nucleo duro» del programma o «tradizione di ricerca» – e dall'altro lato lo sviluppo del *problem solving* svolto dai ricercatori all'interno della cornice, di volta in volta descritto come attività di «risoluzione di rompicapo» o di

---

<sup>3</sup> Il Premio Loebner viene assegnato annualmente dal 1999 al programma per calcolatore che ottiene il punteggio maggiore in una competizione ispirata al TT (si veda <http://www.loebner.net/Prizef/loebner-prize.html>).

costruzione di una «cintura protettiva» del nucleo duro, che si concretizza principalmente nello sviluppo di modelli e teorie in base alle direttive generali incluse nella cornice.

Ora quali sono i processi asincroni più rilevanti per analizzare l'IA come programma di ricerca? Nel rispondere a questa domanda bisogna tenere conto preliminarmente del fatto che l'IA è un programma di ricerca che presenta una forte dipendenza dallo sviluppo della tecnologia informatica, nella sua duplice versione hardware e software. Questa dipendenza non si manifesta solo in quella componente dell'IA che si occupa di sviluppare modelli teorici e prototipi di dispositivi informatici o robotici da usare come strumenti che sostituiscono o coadiuvano gli esseri umani nello svolgimento di funzioni percettive e cognitive, così come nello svolgimento di azioni basate sul coordinamento senso-motorio mediato da processi cognitivi. Anche l'altra componente dell'IA, quella più schiettamente teorica, perché indirizzata allo studio di principi e processi che regolano il comportamento intelligente nei sistemi biologici e nelle macchine, dipende in modo cruciale dallo sviluppo delle tecnologie informatiche.

In considerazione di ciò, al processo di sviluppo della *tecnologia* viene qui assegnato un ruolo centrale nell'analisi dell'IA come programma di ricerca, un ruolo più rilevante di quello che è assegnato a questo processo dai modelli tradizionali dei programmi di ricerca elaborati da Kuhn, Lakatos e Laudan. Complessivamente, prenderemo in considerazione le interazioni tra il processo di sviluppo della *tecnologia*, il processo di sviluppo della *cornice* e il processo di costruzione di *modelli* informatici o robotici.

La ricostruzione che qui proponiamo dell'IA come programma di ricerca tiene conto, come vedremo più avanti, sia della componente teorica che della componente ingegneristica dell'IA. Sotto questo aspetto, la nostra impostazione si distacca dai modelli proposti da Kuhn, Lakatos, Laudan per i programmi di ricerca *scientifici*. Altre differenze emergeranno nel seguito, sia in funzione del ruolo centrale che viene assegnato al processo di sviluppo della tecnologia sia di caratteristiche specifiche che assumono nel contesto dell'IA gli altri due processi, di elaborazione della cornice e di elaborazione di modelli.

Per cominciare, ecco una breve descrizione di ciascuno dei tre processi.

*LA CORNICE.* Questo processo riguarda l'elaborazione iniziale, i successivi rimaneggiamenti ed eventualmente l'abbandono di un nucleo ristretto di assunzioni sulle entità e sui processi che costituiscono il dominio di indagine, sui metodi di studio da adottare e sugli obiettivi di fon-

do da conseguire. Chiamiamo *cornice* questo nucleo di assunzioni che viene elaborato e si evolve nel corso della vita del programma di ricerca dell'IA: esso influenza lo sviluppo di specifici modelli, teorie, programmi per calcolatore e macchinari, senza peraltro identificarsi con i particolari risultati e prodotti sviluppati all'interno del programma di ricerca.

Nel caso dell'IA, come vedremo con qualche dettaglio nel paragrafo seguente, la tesi che sia possibile ottenere comportamenti intelligenti anche in sistemi non biologici, e in particolare in alcune tipologie di macchine, è un elemento chiave e relativamente stabile della cornice. Questa tesi è neutrale rispetto a distinzioni più sottili tra vari modelli realizzati all'interno dell'IA, i quali prendono, per esempio, la forma di programmi per calcolatore, di reti neurali o di artefatti robotici. All'interno della cornice dell'IA possono dunque coesistere quelle che chiameremo *linee di ricerca*, che si differenziano e magari risultano anche in opposizione tra loro, ma tuttavia conservano stretti vincoli di parentela determinati da varie tesi e assunzioni condivise.

I vincoli di parentela tra diverse linee di ricerca sono riconducibili alla condivisione di un nucleo di assunzioni generali che fanno parte della cornice del programma di ricerca. Consideriamo, per esempio, la cosiddetta IA simbolica (generalmente riconducibile alla modellistica o alla simulazione di attività percettivo-cognitive mediante programmi informatici per l'elaborazione simbolica) e il connessionismo (che analizza o sviluppa sistemi in grado di svolgere tali attività facendo ricorso alle reti neurali artificiali). Queste linee di ricerca si distinguono tra loro per alcune posizioni contrastanti, spesso sottolineate in modo esasperato dagli stessi ricercatori. Ciononostante, le assunzioni dalle quali si evince l'opposizione tra IA simbolica e connessionismo non mettono in discussione gli obiettivi ultimi dell'IA, collocandosi perciò a un livello inferiore di generalità rispetto ad altre assunzioni, parimenti condivise dall'IA simbolica, dal connessionismo e da altre linee di ricerca dell'IA. Per esempio, la tesi di fondo che abbiamo appena richiamato – e cioè che sia possibile ottenere comportamenti intelligenti anche in sistemi non biologici e in particolare in alcune tipologie di macchine – sussume le tesi meno generali dell'IA simbolica e del connessionismo, le quali riguardano i modi più opportuni di arrivare a questo obiettivo generale. Da questa prospettiva, IA simbolica e connessionismo non confliggono in merito al *che cosa* ma piuttosto in merito al *come*, alle metodologie e alle tecniche più appropriate per conseguire gli obiettivi generali dell'IA. Per questo motivo, è conveniente considerare IA simbolica e connessionismo come linee di ricerca interne al programma di ricerca dell'IA.



Vi è una relazione dinamica tra le varie *linee di ricerca* interne al *programma di ricerca* dell'IA, le quali ereditano le assunzioni più generali della cornice, pur distinguendosi tra loro per assunzioni meno generali che determinano eventuali divergenze e conflitti reciproci. In particolare, è accaduto che i conflitti iniziali tra linee di ricerca abbiano infine dato luogo, attraverso processi di confluenza e *ibridazione*, ad altre linee di ricerca interne all'IA. Il caso appena menzionato del conflitto tra IA simbolica e connessionismo è esemplare in proposito: dopo un primo momento, il conflitto è stato successivamente ricomposto dando luogo ad alcune linee di ricerca dell'IA che avevano lo scopo di realizzare sistemi computazionali variamente ibridi simbolico-connessionisti (su questo torneremo nel paragrafo 4).

L'ibridazione tra linee di ricerca interne all'IA è una caratteristica diffusa del processo di rielaborazione o rimaneggiamento degli obiettivi e dei metodi dell'IA, che sembra avere avuto maggiore incidenza in questo programma di ricerca piuttosto che in altri, come la fisica, ed è forse paragonabile a quanto è avvenuto invece, sia pure in forme diverse e con diverso peso, nelle scienze biologiche (Mayr 1982) o psicologiche (Mecacci 1992).

Ma quali sono le tesi di fondo che caratterizzano la cornice generale del programma di ricerca dell'IA? La risposta che forniamo a questa domanda, come vedremo nel paragrafo seguente, si articola attraverso la formulazione di cinque tesi che colgono degli *impegni* di fondo o *commitment* ontologici e metodologici dei ricercatori dell'IA, ispirando fin dall'inizio il lavoro scientifico e tecnologico sulle macchine intelligenti. Ciò non significa che la cornice sia rimasta immutata nel tempo. Proprio per il loro carattere generale, i concetti chiave che appaiono nelle cinque tesi sono stati più volte reinterpretati e specificati all'interno di diverse linee di ricerca. In particolare, nel corso dello sviluppo storico del programma di ricerca dell'IA e nell'ambito delle varie linee di ricerca dell'IA, a ciascuna di queste tesi è stato dato un peso diverso e ciascuna è stata oggetto di nuove articolazioni in concomitanza con le fasi di progresso e di stagnazione alle quali è andato incontro il programma di ricerca<sup>4</sup>.

---

<sup>4</sup> È evidente che, in generale, l'individuazione di posizioni di fondo condivise e rielaborate nel tempo da ricercatori che perseguono approcci anche molto diversi tra loro comporta un processo di astrazione rispetto alle posizioni e alle tesi concretamente sviluppate da singoli ricercatori nell'ambito dell'IA. Ma è proprio questo distacco

*LA MODELLISTICA.* Il secondo processo si dispiega nell'attività di risoluzione di rompicapo o, come abbiamo detto, di *problem solving* svolta dai ricercatori dell'IA. Questa attività ha come obiettivo principale lo sviluppo, all'interno della cornice generale, di modelli di varie capacità percettive, di ragionamento, di controllo dell'azione e così via. L'articolazione interna di questo obiettivo nelle varie linee di ricerca dell'IA richiede una riflessione sui diversi significati attribuiti alla parola 'modello' nel corso della storia dell'IA. L'attività di *problem solving* si è più comunemente indirizzata allo sviluppo di *modelli costruttivi*, o prototipi di dispositivi ingegneristici in grado di manifestare determinate capacità percettive, di ragionamento o di controllo dell'azione. Ma altre linee di ricerca dell'IA hanno perseguito, con ambizioni teoriche di spiegazione o previsione scientifica, l'obiettivo di sviluppare modelli computazionali dei processi che consentono agli esseri umani o ad altri sistemi biologici di manifestare comportamenti intelligenti. È questo il caso, per esempio, dell'abbozzo di modello computazionale della visione umana elaborato da David Marr (Marr 1982). Questa pluralità di obiettivi modellistici riflette la distinzione sopra richiamata delle due componenti – teorica e ingegneristica – dell'IA, che ne caratterizza da sempre il suo programma di ricerca fino ad oggi (si pensi anche alla biorobotica attuale, nella duplice versione teorica e ingegneristica secondo la descrizione data per esempio da Lambrinos *et al.* 2000).

La valutazione epistemologica delle attività modellistiche svolte all'interno della cornice richiede che si individuino i risultati durevoli del programma di ricerca rispetto agli scopi delineati nella cornice generale o agli scopi di linee di ricerca particolari all'interno della cornice, che si valutino comparativamente i vari modelli prodotti dall'IA in riferimento a determinati comportamenti intelligenti e, sul lungo periodo, che si identifichino fasi di progresso, stagnazione o regresso del programma di ricerca. Tali criteri valutativi sono utilizzati per cercare di stimare la significatività dei risultati ottenuti *rispetto agli scopi che lo stesso programma di ricerca identifica*, soprattutto mediante la costruzione e il rimaneggiamento della sua cornice generale. Tale valutazione adotta, secondo la terminologia proposta da Laudan, criteri di *razionalità interna* allo stesso programma di ricerca, che bisogna distinguere da valutazioni basate su criteri di *razionalità esterna* (Laudan 1977). Queste ultime prendono in considerazione, tra gli indici di successo

---

che consente di attingere a un punto di osservazione più generale sulle dinamiche di lungo periodo dell'IA.

o progresso, l'utilità sociale o economica dei risultati ottenuti, o altri obiettivi che non sono strettamente inerenti al programma di ricerca.

Una valutazione basata su criteri di razionalità interna serve a prendere decisioni in merito a una rinnovata adesione alla cornice in funzione della significatività rispetto alla stessa cornice dei risultati ottenuti dai ricercatori grazie alla loro attività di *problem solving*, in merito a un rimaneggiamento della cornice alla luce di ricorrenti insuccessi nello sviluppo di modelli adeguati del comportamento intelligente, o anche in merito al definitivo abbandono di certe linee di ricerca alla luce di un'analisi comparativa dei risultati ottenuti rispetto ad altre linee di ricerca nel contesto della cornice. In questo contesto di razionalità interna al programma di ricerca risulta particolarmente evidente l'inadeguatezza del TT come strumento per rispondere a queste fondamentali esigenze metodologiche del programma di ricerca dell'IA.

I criteri di razionalità interna ed esterna confluiscono nei processi decisionali relativi al proseguimento, all'abbandono o al rimaneggiamento di un programma di ricerca. Una valutazione anche fortemente negativa in base a criteri di razionalità interna potrebbe infatti rivelarsi insufficiente a sancire l'abbandono di un programma di ricerca se i risultati conseguiti fossero sufficientemente significativi in base a criteri di razionalità esterna – per esempio nel caso di significative ricadute sociali di risultati poco interessanti rispetto agli obiettivi fissati nella cornice generale. I criteri di razionalità esterna hanno giocato un ruolo significativo nella storia del programma di ricerca dell'IA, come si evince anche dall'intreccio con lo sviluppo tecnologico, che ora passiamo a considerare.

*LA TECNOLOGIA.* Il terzo processo riguarda lo sviluppo, da parte dei ricercatori che lavorano al programma di ricerca dell'IA, di tecnologie hardware e software di supporto alle loro attività di *problem solving*. Molti pionieri dell'IA, come lo stesso Turing, hanno contribuito allo sviluppo di nuove tecnologie hardware o software per l'IA. Lo sviluppo delle tecnologie informatiche ha dato impulsi importanti al programma di ricerca dell'IA, consentendo talvolta di superare fasi di stagnazione, di introdurre *ex novo* oppure di riprendere con nuovo slancio alcune linee di ricerca interne all'IA. Dalla prospettiva dello sviluppo di tecnologie di supporto, un caso di studio esemplare riguarda il succedersi di fasi di progresso e di stagnazione nelle indagini sulle reti neurali e sul connessionismo tra gli anni Cinquanta e gli anni Ottanta del secolo scorso. Su questo caso di studio torneremo più avanti (v. paragrafo 4).

È opportuno tenere conto del flusso bidirezionale di nuove tecnologie informatiche o robotiche da altri settori di ricerca verso l'IA e,

nell'altro senso, dall'IA verso altri settori dell'informatica o della robotica. I modelli dell'IA non sempre forniscono soluzioni soddisfacenti a problemi modellistici sollevati dall'interno del programma di ricerca. Nondimeno, modelli che risultano carenti dal punto di vista della razionalità interna al programma di ricerca, cioè di criteri valutativi che premiano il conseguimento di obiettivi propri del programma di ricerca, possono portare allo sviluppo di tecnologie e strumenti informatici che si rivelano utili in altri settori dell'informatica e delle tecnologie dell'informazione. Anche questi risultati possono essere valutati positivamente, ma in base a criteri di razionalità che non sono strettamente inerenti al programma di ricerca dell'IA. Per esempio, varie tecniche e sistemi software per la rappresentazione e l'elaborazione della conoscenza, sviluppati all'interno dell'IA, sono stati esportati con successo verso settori piuttosto distanti dell'informatica e delle tecnologie dell'informazione. Esportazioni di successo forniscono motivi rilevanti, seppure basati solo su criteri di razionalità esterna, per la sopravvivenza della cornice e, più in generale, del programma di ricerca dell'IA.

Nel paragrafo seguente esamineremo più in dettaglio ciascuno di questi tre processi, prestando maggiore attenzione al primo, cioè alla cornice del programma di ricerca dell'IA, ed evidenziando il contributo che Turing ha dato alla sua rielaborazione. Ci soffermeremo più rapidamente sugli altri due processi, essendo più noto il ruolo da Turing svolto in ambito modellistico e tecnologico. Discuteremo poi nel paragrafo 4 il ruolo della modellistica e della tecnologia nello sviluppo dell'IA dopo Turing, attraverso l'esame di un caso di studio che riguarda due linee di ricerca dell'IA alle quali abbiamo accennato sopra, quella simbolica e quella subsimbolica o connessionista. Questo ci permetterà anche di esemplificare i processi di ibridazione tra linee di ricerca all'interno della cornice dell'IA. Vedremo infine nel paragrafo 5 come la nostra ricostruzione del programma di ricerca dell'IA presenti punti di accordo, ma anche significativi punti di disaccordo, con altre ricostruzioni razionali del progresso scientifico proposte dai filosofi della scienza sopra ricordati con riferimento ad altre discipline.

### 3. *Il contributo di Turing all'IA*

#### 3.1. *Turing e la cornice dell'IA*

Per cominciare, ecco le cinque tesi che contribuiscono a delineare caratteristiche centrali – diremmo le più trincerate – di quella che abbiamo chiamato la cornice dell'IA, presentate in una forma schematica

che intende riflettere lo stato della discussione intorno alle macchine intelligenti già al tempo di Turing<sup>5</sup>.

*Funzionalismo.* I processi alla base dell'adattatività, dell'apprendimento e dell'intelligenza possono essere studiati prescindendo dalle caratteristiche specifiche della struttura organica dei sistemi biologici. Sono infatti le caratteristiche dell'*organizzazione funzionale* dei sistemi che contano. Tali processi possono svilupparsi in supporti materiali diversi, anche in alcune tipologie di macchine, purché i supporti materiali prescelti siano in grado di istanziare l'organizzazione funzionale richiesta.

*Metodo sintetico.* Le ipotesi funzionaliste sui processi alla base dell'adattatività, dell'apprendimento e del comportamento intelligente possono essere controllate empiricamente costruendo artefatti dotati di quella stessa organizzazione funzionale e osservandone il comportamento in condizioni sperimentali opportune. Questa impostazione metodologica può mirare, attraverso la costruzione di artefatti, a fornire *prove di sufficienza* – cioè prove della possibilità tecnologica di sviluppare o replicare comportamenti adattativi e intelligenti in un artefatto (chiamiamo questa la forma 'debole' del metodo sintetico). Ma nel contesto di ricerche sul comportamento adattativo e intelligente di sistemi biologici, dalle prove di sufficienza si può arrivare fino allo sviluppo di modelli simulativi ed esplicativi dei processi soggiacenti al comportamento (chiamiamo questa la forma 'forte' del metodo sintetico).

*Uniformità del metodo.* L'applicazione del metodo sintetico – da adottarsi inizialmente per studiare i processi dell'adattamento, dell'apprendimento, del ragionamento, della deliberazione e della pianificazione – può essere estesa con successo anche allo studio di processi mentali che tradizionalmente sono considerati come processi che sfuggono all'analisi scientifica, ad esempio quelli che sostengono la creatività, l'intuizione, l'intenzionalità e la coscienza.

*Rappresentazionalismo.* La capacità di elaborazione simbolica e di rappresentazione del mondo mediante strutture simboliche è richiesta

---

<sup>5</sup> In questa formulazione schematica, le cinque tesi sono un'astrazione rispetto al concreto svolgimento del dibattito tra i ricercatori dell'IA sull'identificazione di un nucleo del programma di ricerca. Le cinque tesi costituiscono quasi un minimo comune denominatore delle diverse linee di ricerca formulate nel corso dello sviluppo dell'IA. A ben vedere, le loro origini possono essere rintracciate in un'era precedente l'affermarsi della stessa cibernetica, dunque ben prima dell'IA (Cordeschi 2002).

per conseguire un'ampia gamma di comportamenti adattativi e intelligenti.

*Agentività.* Sono applicabili fruttuosamente anche a vari tipi di macchine le ipotesi di razionalità e scopo che un osservatore adotta generalmente per spiegare e prevedere il comportamento di esseri umani e di altri sistemi biologici.

Sia pure brevemente, vogliamo (i) mostrare come il nucleo portante della cornice, esemplificato dalle cinque tesi, si era già sviluppato ai tempi di Turing, (ii) esporre quale sia stato il contributo di Turing alla sua rielaborazione, e (iii) illustrare come questo contributo sia stato in seguito ripreso e ulteriormente rielaborato dai ricercatori dell'IA. Seguendo questo percorso, vedremo anche come le varie linee di ricerca all'interno dell'IA hanno condiviso le cinque tesi della cornice, pur differenziandosi reciprocamente in funzione della diversa rilevanza conferita a ciascuna di esse o in funzione di diverse interpretazioni dei concetti chiave usati nella loro formulazione. Prendendo dunque in considerazione il dibattito intorno alle cinque tesi, potremo individuare con chiarezza punti di contatto o di divergenza tra le varie linee di ricerca dell'IA<sup>6</sup>.

Il contributo di Turing allo sviluppo della cornice si colloca in un contesto di proposte allora innovative per lo studio del comportamento intelligente. Così la tesi 1 del Funzionalismo è stata esplicitamente avanzata – rivendicando sia la centralità dell'organizzazione funzionale dei sistemi che della sua realizzabilità multipla in macchine – al-

---

<sup>6</sup> Si noti che le cinque tesi possono essere utili anche per chiarire le differenze tra quelle che abbiamo richiamato sopra (v. paragrafo 2) come le *diverse* componenti dell'IA, quella ingegneristica e quella teorica. In particolare, quest'ultima componente, nella sua variante modellistica o simulativa (originata dalla *Information Processing Psychology* di Allen Newell e Herbert Simon, poi confluita nella Scienza Cognitiva), è interessata a quella che abbiamo chiamato la forma 'forte' della tesi 2 del Metodo sintetico. La variante non simulativa dell'IA teorica, invece, interessata piuttosto alla formulazione di principi generali dell'intelligenza comuni a organismi e macchine (quella originata, tra le altre, dalle ricerche di John McCarthy sulla rappresentazione della conoscenza), aderisce alla versione che abbiamo detto 'debole' della tesi 2 del Metodo sintetico, ovvero alla semplice prova di sufficienza. Anche la componente ingegneristica dell'IA è interessata solo a una versione 'debole' della tesi 2 ma, a differenza dell'IA teorica non simulativa, è anche neutrale rispetto alla tesi 3 dell'Uniformità del metodo.

lorché è avvenuta quella che possiamo chiamare «la scoperta dell'artificiale», e cioè nel momento in cui si è pensato alle macchine come ad artefatti in grado di adattarsi alle condizioni mutevoli dell'ambiente, di apprendere in base all'esperienza e di rielaborare in modo flessibile le strategie per il conseguimento di determinati obiettivi (Cordeschi 2002). Solo alcune tipologie di macchine parvero soddisfare questi requisiti, diventando così confrontabili sotto questi aspetti con i sistemi biologici. Come si esprimeva lo psicologo Clark Hull più di un quarto di secolo prima della nascita ufficiale dell'IA, a proposito di quello che egli chiamerà l'«approccio del robot» (ovvero lo studio del comportamento degli organismi viventi attraverso la costruzione di artefatti, che costituisce il nucleo del «metodo sintetico»), «l'obiettivo è la costruzione di artefatti [...] che svolgano le funzioni essenziali dell'apprendimento. [...] Apprendimento e pensiero non sono considerati da noi come funzioni esclusive di un organismo più di quanto non lo sia il volo» (Hull e Baernstein 1929, p. 14).

Richiamandosi all'approccio di Hull, Thomas Ross (che William Ross Ashby e William Grey Walter consideravano un precursore della robotica cibernetica) elaborava questa idea nei termini di quello che possiamo chiamare il *funzionalismo delle macchine*, per il quale «vari meccanismi alternativi, molto diversi nelle caratteristiche superficiali e nelle forme dell'energia utilizzata, possono produrre lo stesso risultato finale» (Ross 1938, p. 185). È un'idea, questa, che si ritrova più tardi nella tesi di Norbert Wiener e di altri cibernetici, per i quali i «nuovi» automi (quelli, in breve, dotati di sistemi di controllo basati sulla retroazione negativa) «possono essere trattati in un'unica teoria insieme con i meccanismi della fisiologia» (Wiener 1948, trad. it. p. 71).

Un ampliamento straordinario della classe di automi studiata dai cibernetici si ebbe con l'introduzione dei primi calcolatori digitali *general purpose* costruiti dopo la fine della Seconda guerra mondiale. Sono le capacità di elaborazione simbolica di questi dispositivi che estendono radicalmente le capacità di adattarsi, apprendere e mostrare comportamenti diretti a uno scopo delle macchine cibernetiche, e cioè di macchine dotate di un sistema di controllo in grado di modificarne il comportamento attuale in modo da ridurre l'errore rispetto al comportamento atteso (macchine a retroazione negativa, come abbiamo ricordato). Turing esplora i confini di questo nuovo orizzonte con l'ausilio di un formidabile cannocchiale teorico, quello delle macchine universali, modelli astratti dei calcolatori *general purpose*, che gli consentono di guardare ben oltre i limiti tecnologici dei primi esemplari di calcolatore.

Come è noto, si deve a Turing la scoperta e l'analisi teorica delle macchine universali (Turing 1936-37; si veda il contributo di Gabriele Lolli in questo volume). Una macchina universale è in grado di interpretare ed eseguire un numero potenzialmente infinito di programmi diversi. E nel loro insieme, secondo un'ipotesi di fondo dell'informatica teorica nota come tesi di Church-Turing, questi programmi consentono di calcolare tutte le funzioni calcolabili mediante un algoritmo. Una macchina universale è un modello idealizzato dei calcolatori digitali *general purpose* concretamente costruibili: questi ultimi sono soggetti a limitazioni di memoria e di tempo di calcolo che non si applicano alle macchine universali, ma delle macchine universali conservano, a differenza di altri automi, la capacità di interpretare ed eseguire le istruzioni scritte in un linguaggio di programmazione universale (come, per esempio, il linguaggio C o il Java)<sup>7</sup>.

Turing sposa l'approccio funzionalista della tesi 1 in relazione ai calcolatori *general purpose*, quando egli afferma che le differenze fisiche tra i processi che si svolgono in questi sistemi (per esempio di carattere meccanico o elettrico) e quelli che si svolgono nel sistema nervoso non sono da considerarsi rilevanti: «dovremmo piuttosto interessarci delle analogie matematiche di funzionamento» (Turing 1950, trad. it. p. 173). Ma Turing procede a rielaborare profondamente la formulazione cibernetica della tesi 1, sostenendo che, ai fini della costruzione di macchine capaci di esibire comportamenti intelligenti, è opportuno concentrarsi sui calcolatori *general purpose*. A partire da questi, egli conclude, «è possibile costruire meccanismi per imitare qualsiasi parte limitata dell'uomo» (Turing 1948, trad. it. p. 102). Turing, inoltre, evidenzia le nuove fonti dell'adattatività dei calcolatori *general purpose*, sia in quanto macchine programmabili, capaci di interpretare ed eseguire un qualsiasi programma scritto in un linguaggio di programmazione universale, sia, come vedremo in seguito, in quanto macchine capaci di apprendere modificando i propri programmi in base a meccanismi di ricompensa e punizione.

Certo è l'IA simbolica la linea di ricerca che eredita la forma di funzionalismo propugnata da Turing, mettendo a frutto la programmabi-

---

<sup>7</sup> Diciamo universale un linguaggio di programmazione i cui programmi consentono di calcolare ogni funzione calcolabile da una macchina di Turing. Per un'analisi della tesi di Church-Turing e delle sue implicazioni per l'IA e la scienza cognitiva, si veda Tamburrini 2002. Sulle relazioni tra le nozioni di interpretazione, universalità, programmabilità e virtualità dei calcolatori, si veda Trautteur e Tamburrini 2007.



lità dei calcolatori e dando impulso agli studi sull'apprendimento automatico nei calcolatori<sup>8</sup>. Ma le modifiche e le specificazioni della tesi 1 nella storia dell'IA non si fermano certo a Turing e all'IA simbolica. La tesi 1 è stata modificata e arricchita in seguito, per esempio dalla cosiddetta «nuova IA» che, facendo propria la tesi della *comune organizzazione funzionale* organismi-macchine, ne ha esteso la portata ai meccanismi che consentono la vita e l'evoluzione dei sistemi organici. Di nuovo, anche in questo ambito il tipo di struttura materiale non è ritenuto *direttamente* rilevante alla comprensione del fenomeno indagato (può trattarsi di componenti organici o inorganici). Per esempio, come sostiene Christopher Langton, «la tesi *centrale* [della Vita Artificiale] è che un insieme opportunamente organizzato di componenti di base artificiali che svolgano lo stesso ruolo funzionale delle molecole biologiche nei sistemi viventi naturali fornisce un supporto adeguato per un processo che risulta «vivente» nello stesso senso in cui lo sono gli organismi naturali» (Langton 1989, p. 33).

È evidente che l'approccio funzionalista (tesi 1) è intrinsecamente connesso al metodo sintetico (tesi 2), sia per ciò che abbiamo chiamato prove di sufficienza sia per lo sviluppo di modelli del comportamento adattativo e intelligente dei sistemi biologici. Per quanto riguarda il primo aspetto, come aveva precisato Thomas Ross nel presentare quello che chiamava esplicitamente il «metodo sintetico», «per trovare le *condizioni di sufficienza* dell'apprendimento dovremmo cercare di costruire una macchina che apprende» (Ross 1938, p. 185, corsivo nostro). Per quanto riguarda il secondo aspetto, ancora Ross sosteneva la possibilità di «controllare (*to test*) le diverse ipotesi psicologiche sulla natura del pensiero costruendo macchine ispirate ai principi che implicano tali ipotesi, e confrontando il comportamento delle macchine con quello delle creature intelligenti» (Ross 1935, p. 185). In effetti, in questa proposta ritroviamo il nucleo della metodologia simulativa poi sviluppata da alcune linee di ricerca dell'IA nell'intero arco del suo sviluppo, la quale consiste, molto schematicamente, di tre fasi. Anzitutto si costruisce un artefatto (programma per calcolatore, sistema robotico) come istanza di un modello teorico dell'organizzazione funzionale di un sistema biologico. Successivamente si confrontano le prestazioni dell'artefatto con quelle dell'organismo in situazioni sperimentali ben determinate. Infine, si valutano i risultati

---

<sup>8</sup> Questi ultimi sfoceranno, all'inizio degli anni Ottanta del secolo scorso, nella nascita di quel settore di indagine noto come *machine learning*.

dell'esperimento al fine di corroborare o di evidenziare limitazioni del modello teorico di partenza.

Questo schema metodologico ispira un percorso interno al programma di ricerca dell'IA che porta dalla già ricordata *Information Processing Psychology* di Newell e Simon fino ad alcune linee di ricerca interne alla più recente robotica *behavior-based*, per la quale «il metodo sintetico propone teorie relative a un sistema [naturale] tentando di costruire un sistema artificiale che esibisca le stesse capacità di quello naturale. [...] Nel metodo sintetico il controllo (*test*) della teoria si basa sul confronto tra il sistema artificiale e la realtà» (Steels 1995, p. 92).

Lo schema metodologico riproposto da Luc Steels nell'ambito della robotica *behavior-based* porta ancora una volta in primo piano il complesso problema della natura dei vincoli (*constraint*) da imporre al programma o al sistema robotico perché questo possa aspirare a diventare un'adeguata piattaforma sperimentale per il controllo di un modello teorico dei *processi* che sottendono il comportamento intelligente di un qualche sistema biologico. Non possiamo esaminare qui nei dettagli tale problema che abbiamo approfondito altrove (Cordeschi 2002, cap. 7; Tamburrini e Datteri 2005; Datteri e Tamburrini 2007), se non per dire che i vincoli da imporre sull'artefatto possono essere di natura molto diversa, a seconda del genere di ipotesi e di modelli teorici che il ricercatore decide di prendere in considerazione e mettere alla prova con il metodo sintetico (vincoli psicologici come nel caso di Ross, vincoli derivanti da modelli strutturali o funzionali del sistema nervoso, vincoli comportamentali o anche vincoli evolutivi come nel caso della robotica *behavior-based* o di quella evolucionista)<sup>9</sup>.

Da parte sua Turing, nel considerare lo sviluppo di vari programmi simulativi (per il gioco degli scacchi, ad esempio), riteneva vincoli rilevanti varie forme di plasticità dei calcolatori, a cominciare dalla capacità di apprendere e alla programmabilità, per la quale «non è necessario progettare varie macchine differenti per compiere processi differenti di calcolo» (Turing 1950, trad. it. p. 175). Il requisito della eseguibilità di istruzioni di un linguaggio universale mutuato dalla macchina universale di Turing è stato considerato da Newell (1990) come il primo vincolo di un *Physical Symbol System* e successivamente di architetture

---

<sup>9</sup> Per esempio, il test di sufficienza è stato considerato solo il «passo iniziale» (Paige e Simon 1966) di una metodologia che riteneva indispensabili vincoli più esigenti al fine di assicurare la plausibilità o il grado di realismo dei modelli simulativi a un certo livello di «grana», per riprendere l'espressione di Pylyshyn 1984.

cognitive come SOAR, dove, accanto al vincolo dell'universalità, se ne considerano altri, diretti a garantire la plausibilità psicologica dei modelli simulativi.

La tesi 3 dell'Uniformità del metodo si inquadra all'interno dell'approccio *naturalistico* allo studio della mente, secondo il quale non esistono caratteristiche e processi mentali, come l'intuizione, la coscienza, l'intenzionalità e la creatività, che in linea di principio non possano essere spiegati ricorrendo alle ordinarie procedure della ricerca scientifica. Nello specifico contesto della cornice dell'IA, la richiesta di uniformità comporta la possibilità di estendere il metodo funzionalistico e sintetico dallo studio dei processi cognitivi del *problem solving*, della pianificazione o della percezione fino allo studio dei processi sui quali si basano l'intenzionalità, la creatività e la coscienza.

Al tempo delle dispute sollevate da biologi e psicologi vitalisti (che oggi ci appaiono inevitabilmente datate), Hull doveva preoccuparsi di dimostrare come il suo «approccio del robot» riuscisse a catturare, almeno in prospettiva, caratteristiche complesse della cognizione (degli animali sia umani che non umani) come l'*insight*, quando premetteva che il suo obiettivo era quello di «riuscire a dare una deduzione dell'*insight* in modo tale che un bravo ingegnere potrebbe realizzarlo in un meccanismo non vivente o inorganico» (Hull 1935, p. 231). Si tratta in generale di un obiettivo che era condiviso, in modo non dogmatico, e cioè fino a prova contraria, dai fondatori della cibernetica. Essi infatti osservavano: «Se paragoniamo ancora organismi viventi e macchine arriviamo a queste conclusioni: i metodi di studio per i due gruppi sono oggi simili. Se saranno sempre gli stessi oppure no dipenderà dalla presenza o no di una o più caratteristiche qualitativamente distintive e tipiche di un gruppo e non di un altro» (Rosenblueth, Wiener e Bigelow 1943, trad. it. p. 83).

Certe ipotetiche «caratteristiche qualitativamente distintive» dei sistemi biologici erano state indicate a proposito delle nuove macchine sotto forma di quelle «incapacità varie», come la creatività e la coscienza, di cui parlava Turing quando discuteva e cercava di confutare le «opinioni contrarie a proposito dell'argomento principale [circa l'intelligenza dei calcolatori]» nel suo articolo del 1950<sup>10</sup>. Turing sostie-

---

<sup>10</sup> Si tratta di «incapacità» che sono oggetto di ricerche oggi in primo piano: si pensi al tema della cosiddetta *machine consciousness*, sollevato solo in tempi più recenti all'interno del programma di ricerca dell'IA (Trautteur 1995), tema al quale è dedicata ora anche una rivista, lo *International Journal of Machine Consciousness*.

ne la tesi 3, in relazione alla creatività e alla coscienza, adottando una strategia di *difesa* piuttosto che di elaborazione di proposte costruttive per affrontare i problemi della creatività e della coscienza. Rifacendoci alla terminologia di Lakatos, potremmo collocare la strategia di Turing nell'ambito dell'euristica *negativa* piuttosto che dell'euristica *positiva* di un programma di ricerca scientifico. Infatti, senza portare argomenti a favore della tesi 3, egli cerca di minare le posizioni contrarie e gli argomenti offerti a loro sostegno. La conclusione di Turing faceva emergere il paradosso «delle sfoglie della cipolla»:

Considerando le funzioni mentali o cerebrali troviamo certe operazioni che possiamo spiegare in termini puramente meccanici. Questo, diciamo, non corrisponde alla mente come essa è in realtà: è una specie di sfoglia che dobbiamo togliere se vogliamo trovare la mente reale. Ma poi in ciò che rimane troviamo un'altra sfoglia da togliere, e così via. Procedendo in questo modo arriviamo alla mente 'reale' o solo a un'altra sfoglia? (Turing 1950, trad. it. p. 188).

Si consideri la questione della creatività delle macchine: potremmo magari concludere che l'ordinaria attività di *problem solving* è una caratteristica sondabile con gli strumenti dell'IA, e dunque spiegabile «in termini puramente meccanici» (ovvero automatizzabile), ma che dire dell'«altra sfoglia», e cioè dell'intuizione e del *problem solving* creativo? È in questo, nella creatività, che consiste la *vera* intelligenza, la «mente 'reale'», che come tale sfugge a quegli strumenti, o è pur sempre la creatività una forma, per quanto complessa, di *problem solving* riproducibile mediante opportuni programmi simulativi, come sosteneva Simon (1973) contro Popper, e dunque automatizzabile a sua volta? E in questo caso, qual è la successiva «sfoglia»?

Nella paradossale conclusione di Turing si potrebbe riassumere quella che Vittorio Somenzi ricordava come l'obiezione «sleale», menzionata ai tempi della prima IA da Paul Armer, la quale di volta in volta sposta la definizione di intelligenza «in modo che non rientrino più in essa i fenomeni dei quali abbiamo ottenuto una spiegazione soddisfacente o addirittura la ripetizione da parte di meccanismi da noi costruiti» (Somenzi 1968, p. 231). In questo senso, paradossalmente, «l'IA è tutto ciò che non è ancora stato fatto» (Hofstadter 1979, trad. it. p. 649). Turing aveva sviluppato una riflessione simile nel paragrafo finale del suo articolo del 1948, significativamente intitolato «L'intelligenza come concetto emotivo». Egli osservava che la classificazione pre-teorica di un comportamento come intelligente viene rivista una

volta che i meccanismi soggiacenti siano stati svelati, e che il comportamento stesso divenga prevedibile e spiegabile su questa base:

Quanto siamo disposti a considerare intelligente il comportamento di una qualche entità, dipende tanto dal nostro stato mentale e dalla nostra preparazione quanto dalle proprietà dell'oggetto in esame. Se siamo capaci di spiegare e predire il suo comportamento, o se ci sembra che esso sia poco pianificato, non siamo molto tentati di immaginare un'intelligenza in azione. Perciò è possibile che una persona consideri un oggetto intelligente e un'altra, che ha scoperto le regole del suo comportamento, no. (Turing 1948, trad. it. p. 119)<sup>11</sup>.

L'adesione alla tesi 3 da parte di generazioni successive di ricercatori che hanno operato nell'ambito del programma di ricerca dell'IA si è rivelata più problematica, scontrandosi nel tempo con difficoltà che hanno provocato in tanti delusioni di non poco conto, e che sono anche all'origine di un intricato e non lineare sviluppo dell'IA.

Passiamo ora alla tesi 4 del Rappresentazionalismo. La sua più popolare versione «ristretta», dopo la nascita dell'IA, identifica le rappresentazioni con strutture simboliche discrete, del tipo di quelle che un calcolatore digitale è in grado di manipolare. Qui il ruolo di Turing nella storia dell'IA è stato riconosciuto da sempre come centrale, fin dal suo articolo del 1936 (v. ad esempio Pylyshyn 1984, p. 49 sgg.). Quando, a partire dalla metà degli anni Ottanta del secolo scorso, questa versione che abbiamo detto ristretta delle rappresentazioni è stata criticata all'insegna dello slogan «intelligenza senza rappresentazione» (Brooks 1991), Turing è stato considerato l'alfiere e il campione negativo del rappresentazionalismo, e alla macchina di Turing come prototipo dei modelli computazionali del comportamento intelligente è stato addirittura contrapposto un meccanismo analogico per eccellenza, come il regolatore di Watt, quale prototipo di sistemi dinamici adatti a fornire modelli del comportamento intelligente (van Gelder 1995). Senza entrare qui nel merito di questa disputa (v. Cordeschi e Frixione 2007), che segna il formarsi di nuove linee di ricerca all'interno della

---

<sup>11</sup> Un modo per uscire dalla conclusione paradossale di Turing potrebbe essere quello di vedere esemplificata nella tesi 3 la strategia generale della costruzione dei modelli nella scienza, come la descrivevano ad esempio Rosenblueth e Wiener: una gerarchia di costrutti che va «dalle strutture teoriche estremamente astratte e relativamente semplici sino a quelle più concrete e complesse», più prossime cioè alla complessità reale del fenomeno originale (Rosenblueth e Wiener 1945, trad. it. p. 90).

IA, le quali si proponevano di superare, con un cambiamento di rotta, fasi di stagnazione tanto dell'IA simbolica quanto del connessionismo, vogliamo ricordare qui come la tesi del Rappresentazionalismo si sia di fatto sviluppata anche in relazione agli stati interni di un sistema fisico *diverso* da una macchina a stati discreti, e in particolare a dispositivi analogici come il regolatore di Watt e altri sistemi a retroazione negativa. Essa è stata sostenuta in questa forma, prima di Turing, anche dai pionieri della cibernetica.

Per Kenneth Craik, ad esempio, le rappresentazioni sono «tracce» o «repliche» fisiche degli stati del mondo esterno nel sistema nervoso dell'organismo, e come tali possono essere realizzate in opportuni artefatti. Erano questi le macchine con retroazione negativa e i dispositivi di calcolo analogici dei suoi tempi. Si trattava dunque di «una teoria simbolica del pensiero, in cui il sistema nervoso è visto come una macchina calcolatrice in grado di farsi un parallelo o un modello degli eventi esterni, e [...] che questo processo è la caratteristica fondamentale del pensiero e della spiegazione» (Craik 1943, p. 121). Con l'avvento della cibernetica, rappresentazioni simboliche di questo tipo hanno avuto un ruolo centrale nella teoria di Donald MacKay, per il quale è tramite un «meccanismo imitativo» simbolico che un agente, sia esso un organismo o un opportuno artefatto (sempre un dispositivo analogico), riesce ad interagire con il mondo in modo adattativo. Così l'atto della percezione è considerato come un atto di risposta interna dell'organismo a determinati schemi o *pattern* provenienti dall'esterno, o ancora, così si esprimeva MacKay, «come una risposta autodiretta di conformità (*matching response*) [...] che minimizza certi parametri di non equilibrio o discrepanza (*mismatch*) tra tali *pattern* e la rappresentazione interna». Questa procedura di confronto tra lo stato interno attuale e quello del mondo esterno è realizzabile anche in un opportuno artefatto, in cui essa funziona come «il rappresentante interno o il correlato simbolico del *pattern* percepito corrispondente» (MacKay 1952, p. 73). Nel caso del termostato (un esempio ricorrente nella cibernetica), la temperatura misurata a un dato momento è confrontata con quella richiesta al fine di minimizzare la discrepanza. MacKay riteneva che un siffatto «meccanismo imitativo» fosse alla base anche di attività più complesse descrivibili negli organismi e nelle macchine, dalla classificazione di stimoli alla formazione di concetti fino a diverse forme di coscienza (in coerenza con quella che abbiamo chiamato la tesi 3 dell'Uniformità del metodo).

Per finire, consideriamo brevemente la tesi 5 della Agentività. Ci si potrebbe chiedere perché usare nell'interazione con le «nuove» mac-

chine quel linguaggio degli scopi, dei desideri, delle convinzioni o credenze, delle intenzioni, e così via (in breve il linguaggio della psicologia ingenua o *folk psychology*) che usiamo comunemente, e senza porci troppi problemi, nell'interazione con i nostri simili e anche con gli animali non umani. Già in quest'ultimo caso, quello degli animali non umani, si potrebbero sollevare dubbi e problemi rispetto all'uso del linguaggio della *folk psychology*. Per esempio, all'inizio del secolo scorso il grande zoologo Herbert Jennings poneva questo interrogativo a proposito della descrizione del comportamento di animali inferiori: perché, per prevederne e controllarne il comportamento, attribuiamo comunemente stati mentali a un cane e non, quando la osserviamo al microscopio, a un'ameba, che pure è capace di un numero indubbiamente limitato ma comunque significativo di interazioni plastiche e adattative con l'ambiente esterno? Solo per una questione di dimensioni, rispondeva Jennings, denunciando così con molto anticipo quello che Daniel Dennett chiamava il «pregiudizio della scala spaziale» («se dovessimo guardare nel microscopio per osservare le giocose evoluzioni delle lontre, non saremmo più tanto sicuri del fatto che esse amano divertirsi», Dennett 1996, p. 73).

Per quanto riguarda il caso delle macchine, è noto come siano stati Rosenbluth, Wiener e Bigelow (1943) a sollevare esplicitamente il problema posto dalla tesi 5 nel loro articolo, in cui proponevano una rivalutazione della dignità scientifica del linguaggio teleologico per descrivere il comportamento di certe macchine (quelle con diversi tipi di retroazione negativa), ritenendo tale linguaggio, spogliato di ogni interpretazione vitalistica, utile e talvolta anche necessario in quel contesto. Anche in questo caso, come per le altre tesi che abbiamo esaminato, il problema era stato già sollevato non appena si era manifestata la concreta possibilità di macchine che interagissero in modo adattativo o plastico con l'ambiente esterno. Era infatti la relativa complessità delle prestazioni dei nuovi artefatti a suggerire e a far ritenere giustificata l'attribuzione di scopi e intenzioni per la previsione e il controllo del loro comportamento. Così, sempre sullo sfondo del già ricordato approccio del robot, il biofisico Nicolas Rashevsky osservava come una descrizione in «termini psicologici» di una macchina che apprende, quale una semplice macchina da lui progettata, fosse inevitabile per controllare e prevedere il suo comportamento, anche perché «i processi dell'apprendimento si verificano a un livello diverso da quello di una comune reazione muscolare. [...] È difficile negare che un osservatore non prevenuto definirebbe il comportamento di tale sistema “diretto a uno scopo”» (Rashevsky 1931, pp. 393 e 403).

In realtà questa tesi, per la quale una macchina che interagisca in modo plastico e adattativo con l'ambiente, e dunque con eventuali suoi interlocutori, non può che essere descritta con predicati psicologici, è implicita anche nel Test di Turing (TT). Si pensi al caso di un TT in una formulazione «ristretta», cioè limitato a un'unica prestazione, quello descritto da Turing (1948) nella forma di una macchina che gioca a scacchi con un giocatore umano. Sia quest'ultimo che l'esaminatore devono necessariamente ricorrere al linguaggio teleologico per controllarne e prevederne il comportamento; in breve essi devono trattare la macchina che gioca a scacchi come un *agente razionale*, non meno di come essi reciprocamente si trattano durante la partita. In particolare, il successo del test è dato proprio dalla impossibilità, per l'esaminatore, di distinguere il comportamento della macchina da quello del suo opponente umano: entrambi *devono* essere considerati come agenti razionali, il cui comportamento ha ad ogni passo finalità e scopi ben precisi, al di là della possibilità, sempre presente, che entrambi commettano errori. Errori teoricamente ineludibili che mettono in luce la «razionalità limitata» sia della macchina che del giocatore umano (Simon 1969). A sua volta la razionalità limitata, già particolarmente influente nella prima IA (si ricordi l'idea del calcolatore come macchina «euristica»), ha fornito una motivazione per introdurre il «livello della conoscenza» di Newell (1982), e cioè un livello di descrizione dell'agente razionale in quanto fornito di un repertorio limitato di conoscenze, convinzioni o credenze, strategie di risoluzione di problemi, obiettivi intermedi e finali e di altre caratteristiche ancora.

Indipendentemente dai significativi collegamenti concettuali con la razionalità limitata e il livello della conoscenza, va detto che il TT è stato respinto dalla componente modellistica o simulativa dell'IA, dalle sue origini all'interno della *Information Processing Psychology* fino alla Scienza Cognitiva. Il TT è stato ereditato e discusso da altri settori della prima IA nella formulazione «ristretta» sopra ricordata. È in quella forma che è stato inizialmente sperimentato nel premio Loebner (Cordeschi 1998). Dalla metà degli anni Novanta il TT è stato sperimentato anche in forme meno ristrette o più «generali» (cioè più prossime all'idea del «gioco dell'imitazione» di Turing 1950), fino alle straordinarie ma controverse prestazioni del calcolatore IBM Watson, che ha riaperto recentemente la discussione sul TT nella comunità dell'IA (Shah 2011).

### 3.2. *Turing e i modelli dell'IA*

Abbiamo ricordato come le ricerche di Turing contribuissero ad



aprire la strada a un ampliamento epocale della tipologia degli automi considerati agli esordi della cibernetica, portando in primo piano i calcolatori digitali *general purpose*. Il contributo specifico di Turing allo sviluppo di modelli realizzati sotto forma di programmi per calcolatore è ormai noto, come pure lo è il ruolo da lui svolto nell'avanzamento della tecnologia dei grandi calcolatori nel periodo pre e post bellico, anche attraverso le diverse esperienze che egli fece a contatto con i principali pionieri dell'epoca, primo tra tutti John von Neumann (Hodges 1983; Copeland 2004; Numerico 2005). Possiamo quindi limitarci a sottolineare qui brevemente il contributo di Turing alla modellistica collocandolo nel quadro della nostra ricostruzione del programma di ricerca dell'IA, per ricordare poi come Turing abbia anche considerato la possibilità di sviluppare altre macchine come modelli dell'intelligenza, macchine molto diverse dai calcolatori digitali.

Per quanto riguarda la natura dei modelli proposti da Turing, realizzati sotto forma di programmi per calcolatore, egli, nel paragrafo conclusivo e giustamente celebre dell'articolo del 1950, suggeriva come fosse possibile perseguire contemporaneamente due approcci diversi.

In primo luogo, egli riteneva che la ricerca sulla cosiddetta «intelligenza meccanica» avrebbe potuto concentrarsi sperimentalmente sul gioco degli scacchi e sulle capacità «astratte» dell'intelligenza necessarie in casi come questo, secondo quanto aveva già suggerito nel suo articolo del 1948, dove egli menzionava altri giochi, e anche la traduzione automatica, la crittografia e la matematica, come ambiti di una ricerca che poteva richiedere «poco contatto con il mondo esterno» (Turing 1948, trad. it. p. 104). In un lavoro successivo pubblicato nel 1953, *Digital computers applied to games*, egli esplorò proprio questa linea di ricerca. Qui Turing distingueva tra diverse capacità che dovevano essere implementate sui calcolatori programmati per giocare a scacchi, come la verifica delle mosse consentite, la decisione di tentare una sequenza vincente in un dato numero di mosse, o di giocare un gioco sufficientemente buono, magari apprendendo dall'esperienza. Donald Michie (cit. da Copeland 2004, p. 563) ricorda come Turing, che si era occupato della possibilità di programmare il gioco degli scacchi già dal 1941, avesse intuito l'importanza di quelle che poi si sono definite procedure *euristiche* per la scelta delle mosse. Come avremo modo di accennare nel paragrafo successivo, la programmazione euristica dedicata ai giochi e a diversi rompicapo costituirà una delle prime **linea** di ricerca dell'IA, se non la prima, tuttavia già effettivamente inaugurata dalla pionieristica programmazione del gioco della dama

sui primi grandi calcolatori IBM ad opera di Arthur Samuel, antecedente la nascita ufficiale dell'IA al convegno di Dartmouth nel 1956<sup>12</sup>.

In secondo luogo, Turing, sempre nel paragrafo conclusivo dell'articolo del 1950, osservava come un calcolatore avrebbe potuto utilmente interagire con il mondo esterno attraverso organi di senso, e per esempio apprendere così una lingua. Dal punto di vista della nostra ricostruzione del programma di ricerca dell'IA, queste iniziali formulazioni di Turing relativamente a diversi possibili modi di sperimentare la nascente tecnologia dei calcolatori costituiscono altrettanti incentivi a sviluppare *diverse* catene di modelli all'interno della cornice come l'abbiamo sopra descritta. Una prima possibilità, già praticabile in quel momento, era quella di puntare a modelli sempre più efficaci di attività cognitive come il ragionamento o la pianificazione o l'applicazione di euristiche, cosa esemplificata dal gioco degli scacchi; un'altra possibilità poteva essere quella di puntare in futuro a modelli sempre più efficaci dell'apprendimento del linguaggio naturale o della traduzione tra lingue, lasciando però che tali modelli interagissero con il mondo esterno, e magari «apprendessero» con un meccanismo simile a quello dell'evoluzione naturale (Turing 1950, trad. it. p. 189 sgg.). Ciascuna di queste possibilità ha dato luogo a linee di ricerca diverse all'interno del programma di ricerca della nascente IA. Infine, sappiamo ormai come Turing pensasse anche a macchine molto diverse dai calcolatori digitali come modelli dell'intelligenza, quelle che egli definiva «macchine non organizzate», ovvero reti addestrabili di unità connesse in modo casuale (Turing 1948). A queste macchine si è dato recentemente un grande rilievo, perché nelle intuizioni di Turing in proposito alcuni hanno riconosciuto i prodromi di altre linee di ricerca poi sviluppatesi all'interno dell'IA, quelle, alle quali abbiamo anche accennato, promosse dai sostenitori del connessionismo e dei sistemi dinamici (Copeland 2002, p. 403 sgg.).

In conclusione, Turing ha svolto un ruolo di primo piano tra i pionieri che lavorarono allo sviluppo di ciascuno dei tre processi da noi descritti come cruciali per il nascente programma di ricerca dell'IA, quelli della cornice, della modellistica e della tecnologia hardware e software. Anche altri pionieri dell'IA lo hanno fatto in varia misura,

---

<sup>12</sup> Si veda Cordeschi 2007. Il documento ispiratore del convegno, redatto nell'anno precedente da John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon, è stato pubblicato in trad. it., a cura di R. Cordeschi e M. Frixione, nella rivista *Sistemi Intelligenti*, 18, 2006, pp. 407-13.

come abbiamo già ricordato per quanto riguarda lo sviluppo della cornice e dei modelli. Quanto alla tecnologia software, non possiamo non menzionare almeno lo sviluppo, a partire dai primi anni Cinquanta del secolo scorso, delle reti neurali Snarc di Marvin Minsky e di quelle di Nathaniel Rochester e collaboratori, e di linguaggi di programmazione come lo IPL ad opera di Newell e Simon, in collaborazione con Clifford Shaw, o come il LISP ad opera di McCarthy (v. per dettagli Cordeschi 2002, cap. 6).

#### 4. *Ibridazioni e sviluppo tecnologico tra IA simbolica e subsimbolica*

A Turing possono dunque farsi risalire importanti intuizioni sull'intelligenza delle macchine che poi sono state sviluppate da linee di ricerca diverse all'interno dell'IA. Un esempio sul quale vogliamo fermarci brevemente è quello dell'IA simbolica e del connessionismo. Si tratta di un caso in cui emergono in modo evidente due aspetti ai quali abbiamo accennato, e che riteniamo essere stati molto influenti nello sviluppo del programma di ricerca dell'IA. Il primo è quello delle ibridazioni, l'altro quello dell'influenza della tecnologia dei calcolatori.

La linea di ricerca dell'IA simbolica si sviluppa in particolare da una specializzazione della tesi 4 del Rappresentazionalismo alle strutture simboliche discrete manipolabili mediante regole algoritmiche e programmi per calcolatore. Per cominciare, va detto che all'interno dell'IA simbolica i ricercatori hanno spesso radicalizzato le differenze tra i diversi modelli da loro stessi proposti, finendo per parlare di «paradigmi» contrapposti. Si pensi, per fare un esempio, al caso della risoluzione automatica di problemi, il settore nel quale si è cimentata l'IA fin dalle sue origini. Spesso la sua evoluzione è stata descritta in termini di coppie di «paradigmi» contrapposti. L'elenco di tali asseriti paradigmi include la ricerca euristica (basata sulla semplice esplorazione dell'albero di ricerca) vista in contrapposizione alla rappresentazione della conoscenza; i metodi deboli (o generali) come contrapposti ai metodi forti (o relativi a un dominio specifico); i *toy problems* o problemi semplificati come contrapposti ai problemi della vita reale. Vi sono state altre contrapposizioni ancora, come quelle, piuttosto note, tra logicisti e antilogicisti e tra dichiarativisti e proceduralisti.

Non possiamo entrare qui nei dettagli di queste singole dispute, che hanno a lungo diviso la comunità dell'IA, e che spesso si sono più dissolte che risolte (Cordeschi 2002, cap. 6). È tuttavia da sottolineare che il termine «paradigma» come veniva usato nei casi sopra ricordati non

può essere inteso in senso tecnico (ovvero, come vedremo nel paragrafo seguente, à la Kuhn). Più che di paradigmi, si tratta di *linee di ricerca interne all'IA, anzi in questo caso interne alla stessa IA simbolica*. Nel programma di ricerca dell'IA si assiste infatti allo sviluppo di linee di ricerca di diversa generalità, che possono anche dar luogo a ibridazioni che cercano di unire i punti di forza delle linee di ricerca interessate superandone i rispettivi limiti. Nel caso che esaminiamo, queste ibridazioni si sono realizzate non solo, lo vedremo nel seguito, tra IA simbolica e IA connessionista, ma anche tra linee di ricerca diverse *interne all'IA simbolica*, dunque a un livello di generalità minore delle prime (per maggiori dettagli, si veda Cordeschi 2006).

Un esempio di quest'ultimo tipo è l'esito della contrapposizione appena ricordata tra l'asserito paradigma della ricerca euristica attraverso metodi generali, cosiddetti «deboli», applicabili soprattutto a *toy problems*, e quello della ricerca basata sulla conoscenza, ricerca che in questo caso si ritiene possibile grazie al ricorso a metodi specializzati relativamente a un certo dominio, cosiddetti «forti», in grado di affrontare problemi della vita reale. Per esemplificare: nel primo caso abbiamo problemi come la logica o i diversi giochi e rompicapo (la torre di Hanoi e così via), affrontabili con procedure per scegliere le mosse che sono basate su indizi diversi. Si tratta di procedure dette euristiche, solitamente di tipo molto generale e dunque deboli (per esempio, la scomposizione di un problema in sottoproblemi più semplici). Nel secondo caso abbiamo problemi complessi, come fare una diagnosi medica, dove nella presa di decisione entrano in gioco, oltre alle aspettative disattese e agli imprevisti in cui ci si imbatte nella vita reale, soprattutto conoscenze specializzate anche se sempre incomplete, e metodi di ricerca euristica ad esse relativi, solitamente detti forti. Intorno alla fine degli anni Sessanta si ebbe il passaggio dalla risoluzione automatica di problemi attraverso l'albero di ricerca alla risoluzione automatica di problemi basata sulla conoscenza (che caratterizza i cosiddetti sistemi esperti). Ma l'una e l'altra linea di ricerca, dopo un'iniziale contrapposizione, hanno finito per «ibridarsi» (il termine è di Simon), arrivando a suggerire nuovi e più soddisfacenti modelli e anche nuove intuizioni teoriche.

C'è da sottolineare in questo contesto un fattore decisivo già ricordato: lo stato di avanzamento della tecnologia dei calcolatori. Come abbiamo detto, si tratta in realtà di un fattore che ha influenzato in modo decisivo la storia dell'intera IA, certe sue svolte, certi suoi successi e insuccessi, e anche certe sue illusioni. Nell'IA simbolica, un esempio è il caso che stiamo esaminando dello spostamento dell'enfasi dai modelli

basati sulla semplice ricerca euristica ai modelli basati sulla conoscenza, e dunque dallo studio di *toy problems* a quello di problemi della vita reale. Era Simon a ricordare come la scelta iniziale di concentrarsi sullo studio dei primi piuttosto che sui secondi si doveva anche allo stato della tecnologia: «Non è che non fossimo consapevoli dell'importanza giocata dalla conoscenza [nella risoluzione di problemi]. Piuttosto, non venivano considerati compiti nei quali la conoscenza era un fattore essenziale perché non si era in grado di costruire grandi basi di dati con i calcolatori dell'epoca» (si veda in Crevier 1973, p. 147). L'iniziale enfasi su compiti che richiedevano poca o pochissima conoscenza da parte del solutore, come i ricordati *toy problems*, sembra dunque essere stata una scelta pressoché obbligata per i pionieri della IA.

Ma la storia non finisce qui. Inizialmente, Edward Feigenbaum, considerato il padre dei sistemi basati sulla conoscenza o sistemi esperti, concluse che i metodi deboli erano inefficienti e insufficienti per costruire modelli efficaci della soluzione «esperta» di problemi del tipo della vita reale, metodi che dovevano essere dunque potenziati ricorrendo all'uso, da parte del programma, di conoscenza specifica (si pensi ancora al caso della diagnosi medica). La successiva ricerca mostrò tuttavia che le iniziali buone prestazioni dei sistemi basati su conoscenze specializzate, quali erano i sistemi esperti, non portavano ai risultati che ci si poteva aspettare. Negli anni Ottanta, lo stesso Feigenbaum riassume le difficoltà incontrate nel costruire sistemi esperti con prestazioni efficienti in un articolo scritto con Douglas Lenat: «Un limite dei sistemi esperti passati e presenti sta nella loro fragilità (*brittleness*). Essi funzionano efficientemente in base alla loro piattaforma elevata di conoscenze e competenze finché non raggiungono i limiti delle loro conoscenze, dopodiché essi precipitano rovinosamente a livelli di totale incompetenza. Certo, anche la gente incontra difficoltà del genere, ma la loro piattaforma è molto più estesa, e il loro declino meno rapido. *La caduta è in parte attenuata dal possesso di metodi più deboli e generali che sottostanno alla conoscenza specializzata*» (Lenat e Feigenbaum 1991, p. 196, corsivo nostro).

Le esperienze di Feigenbaum sui sistemi esperti e quelle di Lenat sui programmi per l'apprendimento e la scoperta di concetti scientifici in ambiti più o meno specialistici sembravano dunque portare a diverse successioni o catene di modelli (per riprendere la terminologia di Lakatos) che soffrivano di una stessa limitazione: entrambe non erano in grado di sfruttare quella che gli stessi autori chiamavano «la conoscenza basata su una realtà condivisa» (*consensus reality knowledge*), che negli esseri umani realizza la conoscenza basata sul senso comu-

ne, spesso di carattere analogico e impreciso, basata com'è proprio su metodi di ricerca di tipo generale o deboli, e che soccorre la presa di decisione in casi di incertezza.

Insomma, la ricerca sui sistemi basati sulla conoscenza, dopo aver conseguito alcuni successi nell'ambito della risoluzione automatica di problemi, era entrata a sua volta, negli anni Ottanta, in una fase di stagnazione, che non sembrava portare a nuovi progressi. La conoscenza basata sul senso comune costituì la vera bestia nera, o se si vuole la fonte di anomalie difficilmente eliminabili all'interno di quella linea di ricerca. Il successivo tentativo di Lenat dell'ambizioso programma CYC (da *enCYClopedia*) può essere qui considerato come il tentativo di uscirne, progettando un sistema con il quale si tenta di gestire efficientemente la conoscenza del senso comune in domini sempre più vasti. Non completato nella sua formulazione originale, CYC ha finito per incrociarsi con la ricerca attuale sulle ontologie per il web (Cordeschi 2006).

L'IA simbolica è una linea di ricerca che a un certo punto è entrata in competizione con una linea di ricerca alternativa, ma ugualmente ispirata dalle cinque tesi della cornice: quella delle reti neurali, subsimbolica o connessionista, secondo la quale ogni tipo di comportamento intelligente può emergere dall'attività di elaborazione parallela dell'informazione da parte di unità di calcolo originariamente concepite come modelli semplificati del funzionamento dei neuroni biologici. Per distinguerli dalle rappresentazioni simboliche dell'IA simbolica, questi pattern di attivazione che evolvono nel tempo vengono talvolta chiamati rappresentazioni subsimboliche. Dal punto di vista che qui proponiamo, l'IA simbolica e quella subsimbolica costituiscono *due linee di ricerca concorrenti, all'interno della cornice più generale formata dalle cinque tesi*. Questo ha consentito lo sviluppo di modelli molto diversi tra loro, che hanno poi dato luogo anche a ibridazioni in forma neuro-simbolica. Proviamo a vedere come sono andate le cose con qualche dettaglio.

Un convegno importante nella storia della IA dopo Dartmouth è stato quello svoltosi a Teddington nel 1958. Forse è in questa occasione che si materializzò per la prima volta il contrasto tra le linee di ricerca dell'IA simbolica, sostenuta da Minsky, e quella subsimbolica, sostenuta da Frank Rosenblatt. Ciascuno di essi portò argomenti che tendevano a mostrare i vantaggi del proprio approccio e i limiti di quello concorrente. Secondo Minsky, i sostenitori delle reti neurali non erano in grado di proporre modelli di attività dell'intelligenza per così dire alta, come il ragionamento basato su concetti astratti. Rosenblatt, dal

canto suo, presentando in quell'occasione il suo Perceptron, sostenne che i modelli basati su programmi per calcolatore non permettevano di costruire modelli biologicamente plausibili, in particolare per quanto riguardava le capacità associative e l'apprendimento (Cordeschi 2002, cap. 6).

A partire dagli anni Sessanta, la ricerca sulle reti neurali entrò in una fase che possiamo dire di stagnazione, almeno nel confronto con i primi successi dell'IA simbolica (quelli ai quali abbiamo accennato nell'ambito della risoluzione automatica di problemi). Il celebre libro pubblicato nel 1969 da Minsky con Seymour Papert sui limiti dei Perceptron costituì un altro argomento per il ridimensionamento della linea di ricerca sulle reti neurali. Una fase di ripresa di quest'ultima risale ai primi anni Ottanta, con l'introduzione di nuovi e più potenti modelli di reti neurali<sup>13</sup>.

Retrospectivamente, si può dire che si è trattato di un'euforia durata un periodo molto breve. Tra gli anni Ottanta e Novanta, i sistemi connessionisti mostrarono ben presto limiti difficilmente superabili, sostanzialmente confermando la diagnosi che Minsky aveva fatto a Teddington. I sistemi connessionisti, come ai vecchi tempi del Perceptron di Rosenblatt, davano il meglio di sé in compiti di riconoscimento di forme, di associazione e in certe forme di apprendimento, ma generalmente si mostravano inadeguati in compiti che richiedevano capacità di ragionamento e di pianificazione. Ancora una volta, sistemi «ibridi» sono stati proposti per rispondere a queste problematiche, superando i limiti complementari di IA simbolica e subsimbolica, combinando i punti di forza di entrambi e reinterpretando in modo opportuno le cinque tesi della cornice (Cordeschi 2002, cap. 6).

Per concludere, va sottolineato che anche nel caso delle reti neurali vale quanto abbiamo detto sopra a proposito della ricerca euristica nel dominio dei *toy problems*, e cioè come lo stato della tecnologia dei calcolatori abbia influito nell'orientare la ricerca effettiva (le reti neurali si simulano tramite software per calcolatore). Su questo problema ha insistito proprio un connessionista, James McClelland, osservando che «la potenza di calcolo di cui si disponeva negli anni Sessanta era completamente insufficiente [per lo sviluppo di reti neurali]». Anche qui le

---

<sup>13</sup> La ripresa si deve ad almeno due lavori: quello di John Hopfield sulle memorie associative e quello sull'algoritmo di retropropagazione (*backpropagation*) di David Rumelhart e collaboratori, che consentiva di superare alcuni limiti dei Perceptron denunciati da Minsky e Papert nel loro libro.

limitazioni della tecnologia dei calcolatori hanno influito nell'orientare inizialmente i ricercatori verso una linea di ricerca (quella simbolica) piuttosto che verso un'altra (quella subsimbolica o connessionista), al punto che sempre McClelland smentiva esplicitamente uno dei luoghi comuni della polemica connessionista, quello che gettava sul libro di Minsky e Papert la principale responsabilità della stagnazione e della regressione della ricerca sulle reti neurali negli anni Sessanta (Crevier 1993, p. 309).

##### 5. *Il programma di ricerca scientifico dell'IA e la filosofia della scienza*

L'impostazione data all'analisi dei contributi di Turing nel contesto del programma di ricerca dell'IA presenta, come abbiamo sottolineato più sopra (v. paragrafo 2), evidenti punti di convergenza con il modello della scienza normale di Kuhn, il modello dei programmi di ricerca scientifici di Lakatos e il modello delle tradizioni di ricerca di Laudan. Si possono tuttavia cogliere significative differenze con ciascuno di questi modelli. Vediamo perché, iniziando dalla scienza normale di Kuhn.

Nei periodi di scienza normale, secondo Kuhn i ricercatori svolgono la propria attività di *problem solving* o di risoluzione di rompicapo in base a un paradigma. Come ha osservato Laudan, tuttavia, un paradigma kuhniano ha una struttura eccessivamente rigida, che ne impedisce l'evoluzione nel tempo in funzione delle anomalie che esso genera. Inoltre, un paradigma riceve una formulazione solo implicita, nella misura in cui esso viene identificato per mezzo di casi «esemplari», e cioè mediante l'individuazione di qualche applicazione di concetti e teorie matematiche nell'ambito della fisica o di altre scienze empiriche, che deve fungere da modello di riferimento per la soluzione di ulteriori rompicapo.

La ricostruzione dell'IA come programma di ricerca che abbiamo qui abbozzato si discosta dai paradigmi kuhniani per entrambe le caratteristiche. Infatti, la formulazione della cornice all'interno della quale i ricercatori svolgono le loro attività scientifiche e tecnologiche non avviene solo in modo implicito mediante il riferimento a casi esemplari. La cornice è formata da tesi che, pur contemplando varianti non sempre equivalenti tra loro, costituiscono un nucleo di assunzioni esplicite sugli enti e sui processi da indagare e sui modi più opportuni per sviluppare le indagini scientifiche e tecnologiche. E abbiamo anche visto che la cornice, al contrario di un paradigma, può essere modifi-



cata. Di fatto essa è stata modificata incisivamente da Turing, subendo in seguito ulteriori modifiche alle quali abbiamo potuto solo accennare per grandi tratti. In questi passaggi, tuttavia, non si ravvisa quel cambiamento gestaltico che Kuhn identifica con un mutamento di paradigma. Si tratta piuttosto di un processo graduale di rimaneggiamento della cornice, ottenuto dando a ciascuna delle cinque tesi un peso diverso o specificando diverse accezioni dei concetti che occorrono in esse, come quelli di macchina, di rappresentazione, di uso simulativo delle macchine, e così via. Inoltre, come abbiamo visto nel paragrafo precedente, ricostruire lo sviluppo dell'IA attraverso la lente dei cambiamenti gestaltici, ovvero dei mutamenti di paradigmi contrapposti, rende difficilmente spiegabili i non pochi processi di ibridazione che si verificano in IA, sia tra linee di ricerca diverse di IA sia all'interno di una stessa sua linea di ricerca.

Consideriamo ora i programmi di ricerca scientifici di Lakatos. La cornice del programma di ricerca dell'IA ha elementi significativi in comune con quello che Lakatos chiama il nucleo di un programma di ricerca scientifico (PRS). Infatti, il nucleo di un PRS è una collezione di principi euristici per la costruzione di teorie e modelli scientifici. Il nucleo, che è costituito da enunciati che riguardano la natura e il comportamento di varie entità e processi, può includere (o fare riferimento a) un corpo di leggi scientifiche, insieme a varie tesi ontologiche ed epistemologiche sul dominio di indagine. Per esempio, il nucleo del PRS noto come fisica cartesiana può essere riassunto con l'asserzione che «l'universo è un enorme meccanismo (e sistema di vortici) nel quale l'urto è l'unica causa del movimento» (Lakatos 1970, p. 57). Si tratta di un enunciato metafisico, secondo i criteri di classificazione tra tipi di enunciati proposti da Popper e ripresi da Lakatos. Una tale asserzione, infatti, non può essere contraddetta da nessuna asserzione singolare, e dunque non risulta falsificabile mediante esperimento (Lakatos 1970, trad. it. pp. 47 e 33). Da questa prospettiva, devono essere classificate come metafisiche anche la tesi 1 del Funzionalismo o la tesi 3 dell'Uniformità del metodo, non essendovi esperimenti in grado di falsificarle direttamente.

Secondo Lakatos, il nucleo si conserva sostanzialmente invariato nel corso dello sviluppo di un PRS, fino al suo eventuale abbandono a vantaggio di un altro PRS. Il nucleo del PRS della fisica cartesiana è stato abbandonato per il nucleo di un PRS rivale di maggiore successo, quello newtoniano, che ammette l'azione a distanza e include sia la legge di gravitazione di Newton sia le tre leggi della dinamica (Lakatos 1970, trad. it. p. 58). Come abbiamo già rilevato a proposito dei para-

digmi kuhniani, l'invarianza del nucleo nel corso dello svolgimento di un PRS non trova riscontro nell'impostazione che abbiamo dato all'analisi dell'IA come programma di ricerca. Infatti non solo singole linee di ricerca dell'IA ma, come il nucleo di un PRS, perfino la cornice del programma di ricerca dell'IA può essere abbandonata in situazioni di persistente stagnazione delle ricerche e in presenza di programmi di ricerca alternativi di maggiore successo.

Le differenze tra i PRS di Lakatos e l'impalcatura introdotta nei paragrafi precedenti si estendono alle *relazioni tra i modelli* prodotti nel quadro del programma di ricerca dell'IA. Lakatos si limita a consentire la costruzione, all'interno di un PRS, di catene di modelli, all'interno di ciascuna delle quali ogni nuovo modello ha un contenuto empirico maggiore di modelli precedentemente introdotti<sup>14</sup>. Nel programma di ricerca dell'IA, non è stato quasi mai possibile confrontare il contenuto empirico dei singoli modelli, cosicché le relazioni tra modelli non ricadono, in generale, nel caso previsto da Lakatos. In realtà, salvo alcune eccezioni (esemplari il caso di Newell e Simon nello studio di funzioni cognitive 'alte' come il *problem solving* o quello di David Marr nella visione artificiale), la comunità dell'IA, negli anni che seguirono la morte di Turing, non è stata sempre sensibile alla questione metodologica di elaborare criteri, magari diversi dai criteri particolarmente restrittivi proposti da Lakatos, che permettessero di confrontare i risultati della ricerca nel settore, consentendo di riconoscere i progressi compiuti con lo sviluppo di nuovi modelli, programmi per calcolatore o sistemi robotici.

Vi sono relazioni più strette tra la nostra ricostruzione del programma di ricerca dell'IA e le tradizioni di ricerca di Laudan. Gli elementi costitutivi della cornice, così come le tradizioni di ricerca di Laudan, sono assunzioni relative al dominio di indagine e ai metodi da adottare per studiare le entità e i processi nel dominio di indagine<sup>15</sup>. E la cornice dell'IA evolve nel tempo, come si vede dalle varie interpretazioni e

---

<sup>14</sup> In questo contesto Lakatos riprende l'idea popperiana del confronto tra diverse teorie basato sul rispettivo contenuto empirico, mettendo in luce la complessità delle decisioni metodologiche richieste per istituire il confronto nel quadro della sua metodologia dei programmi di ricerca scientifici. Si veda per esempio la nota 82 a p. 111 della trad. it. di Lakatos 1970.

<sup>15</sup> Secondo Laudan, una tradizione di ricerca è formata da un insieme di assunzioni sulle entità e i processi all'interno di un dominio di indagine, e sui metodi più appropriati per costruire teorie e affrontare i problemi di quel dominio. Si veda, in

precisazioni delle cinque tesi, proprio come una tradizione di ricerca, ma al contrario dei paradigmi di Kuhn o dei nuclei dei PRS di Lakatos. Vi sono tuttavia altre differenze rilevanti tra la cornice e le tradizioni di ricerca di Laudan. Ci limitiamo qui a metterne in evidenza tre per quanto riguarda l'IA: l'importanza dello sviluppo tecnologico per l'avanzamento del suo programma di ricerca, il ruolo centrale del processo di ibridazione tra linee di ricerca interne alla sua cornice, il ruolo centrale di ciò che chiamiamo *razionalità esterna* nella valutazione dei suoi risultati<sup>16</sup>.

Sulle prime due ci siamo già soffermati in precedenza. Lo sviluppo della tecnologia è poco considerato nello schema di Laudan<sup>17</sup>, mentre da noi è stato individuato come particolarmente influente sul programma di ricerca dell'IA, sia per avviare una linea di ricerca all'interno all'IA sia per ridarle slancio. Analogamente, i limiti contingenti di una tecnologia hanno talvolta determinato la stagnazione o l'abbandono di alcune linee di ricerca interne all'IA. Abbiamo fatto l'esempio delle reti neurali degli anni Cinquanta; un altro esempio è la robotica cibernetica, alla quale si ispirano i successivi sviluppi della robotica *behavior-based* degli anni Ottanta del secolo scorso: essa arrivò a una situazione di stallo sia per la mancanza di una miniaturizzazione efficace dei componenti sia per l'assenza di modelli architetturali adeguati (Brooks 1995, p. 38).

Per quanto riguarda i processi di ibridazione, è opportuno evidenziare che le dinamiche del progresso scientifico considerate da Laudan comprendono processi di *amalgamazione* tra tradizioni di ricerca<sup>18</sup>. Ma Laudan non si sofferma su processi di amalgamazione interni a una tradizione di ricerca. Qui abbiamo invece sottolineato l'ampia portata dei processi di ibridazione tra linee di ricerca interne all'IA,

---

particolare, il paragrafo intitolato "La natura delle tradizioni di ricerca" nel capitolo 3 di Laudan 1977.

<sup>16</sup> Vi sono altre differenze, che segnaliamo ma non affrontiamo perché meno rilevanti per i problemi metodologici esaminati in questo paragrafo. Per esempio, Laudan classifica la cibernetica (ma implicitamente anche l'IA per motivi analoghi) come una tradizione di ricerca 'non-standard' per assenza di impegni o *commitment* ontologici.

<sup>17</sup> Nell'ambito della storia della fisica, settore privilegiato di analisi per Laudan, è stato Peter Galison a sottolineare la relativa autonomia e l'influenza dei processi di sviluppo degli strumenti scientifici sulla ricerca teorica e sperimentale (Galison 1997).

<sup>18</sup> Si veda il paragrafo intitolato "L'integrazione delle tradizioni di ricerca" nel capitolo 3 di Laudan 1977.

anche per il loro impatto sull'interpretazione e il rimaneggiamento della cornice, come abbiamo visto sopra nell'esempio dell'IA simbolica e subsimbolica<sup>19</sup>.

Veniamo ora alla questione della razionalità esterna. Galison sottolinea elementi di disunità all'interno della fisica, evidenziando i confini tra differenti «sottoculture» e comunità, come quelle che si occupano dello sviluppo della strumentazione, quelle che si concentrano sull'impostazione e sulla conduzione degli esperimenti, e quelle che si concentrano sull'elaborazione di teorie e modelli. Queste varie sottoculture condividono alcune attività, pur divergendo tra loro per altre attività, interessi e obiettivi di ricerca. Le varie sottoculture godono perciò di una parziale indipendenza reciproca. Tra di esse, tuttavia, lo scambio è stato intenso e reciprocamente fruttuoso. Per descrivere siffatte interazioni, Galison ricorre all'analogia con le *trading zone*, zone franche dedicate al commercio tra comunità geograficamente limitrofe. Anche l'IA è stata molto attiva nello scambio tecnologico con altre comunità che ad essa risultano limitrofe sia scientificamente che tecnologicamente. Come è stato già messo in evidenza in relazione alla distinzione tra razionalità interna ed esterna al programma di ricerca, le tecnologie software sviluppate all'interno dell'IA non hanno sempre portato ai risultati attesi nella costruzione di modelli soddisfacenti di aspetti del comportamento intelligente. E tuttavia le stesse tecnologie, insoddisfacenti dal punto di vista della razionalità interna, si sono rivelate essere utili per altri settori dell'informatica e delle tecnologie dell'informazione. Alcune tecnologie software sviluppate nell'ambito dell'IA vengono utilizzate come componenti di *smartphone*, di motori di ricerca, di giochi informatici, di sistemi multimediali, di sistemi per il commercio in rete o per il monitoraggio e controllo del consumo energetico. In generale, le tecnologie dell'IA consentono di dotare sistemi informatici complessi di capacità come la percezione, il ragionamento, l'apprendimento e la pianificazione.

In tutti questi casi, i risultati conseguiti soddisfano criteri deboli di razionalità *esterna* per la sopravvivenza della cornice: i risultati sono infatti giustificati in relazione agli obiettivi di ricerca e sviluppo di co-

---

<sup>19</sup> Ricordiamo anche che lo sviluppo della cornice ha elementi in comune con l'individuazione di temi di fondo per la ricerca scientifica che Gerald Holton ha chiamato *themata*, identificandoli con concezioni stabili e pervasive nelle comunità scientifiche, che non possono ridursi alle pratiche sperimentali e teoriche del lavoro scientifico (Holton 1988).

munità limitrofe, piuttosto che in relazione al conseguimento di obiettivi che sono caratteristici del programma di ricerca dell'IA.

La distinzione tra razionalità esterna ed interna nella giustificazione dei risultati ottenuti dal programma di ricerca dell'IA non coincide con la distinzione proposta da Laudan tra le motivazioni razionali e non razionali che inducono a perseguire un programma di ricerca o a privilegiare selettivamente alcuni problemi e attività di *problem solving* tra quelli che una tradizione di ricerca deve affrontare. Come motivazioni non razionali Laudan elenca le pressioni economiche, politiche, morali o sociali. Le applicazioni fruttuose dell'IA in settori limitrofi non dipendono necessariamente da motivazioni o pressioni economiche, politiche, morali o sociali. E tuttavia l'impatto di quelle che Laudan chiama motivazioni non razionali è stato molto forte anche sul programma di ricerca dell'IA. Basta ricordare a tale riguardo l'impulso economico che l'apparato militare-industriale ha dato allo sviluppo delle ricerche della cibernetica e dell'IA (Cordeschi e Tamburrini 2006; Datteri e Tamburrini 2012).

## 6. Conclusioni

La ricostruzione del programma di ricerca dell'IA che abbiamo proposto, nella sua duplice versione ingegneristica e teorica (quest'ultima, a sua volta, nella duplice tendenza simulativa e non simulativa), si è articolata attraverso la descrizione di una cornice, di una modellistica e di una specifica tecnologia, quella dei calcolatori. Abbiamo individuato cinque tesi, relativamente stabili, che caratterizzano la cornice, e abbiamo descritto brevemente come all'interno di tale cornice si siano avvicendate diverse linee di ricerca, i cui risultati si sono realizzati in altrettante catene di modelli, per la costruzione dei quali si è sempre rivelato cruciale il livello di sviluppo della tecnologia dei calcolatori. Tali catene di modelli si sono spesso sviluppate con esiti diversi, dando luogo a fasi di progresso o a fasi di stagnazione, quando non di regressione, delle rispettive linee di ricerca. Abbiamo visto come vari tentativi di superamento delle fasi di stagnazione o regressione sono stati attuati attraverso processi di ibridazione tra linee di ricerca. Ma abbiamo visto anche come le ibridazioni possono aver luogo all'interno di una stessa linea di ricerca, quando questa si evolve o si articola a sua volta in linee di ricerca ancora più specifiche. Abbiamo fatto nel primo caso l'esempio dell'IA simbolica e di quella connessionista; nel secondo caso, abbiamo ricordato alcune confluenze interne alla linea

di ricerca dell'IA simbolica, che si sono sviluppate a partire da contrapposizioni iniziali (*toy problems* vs. *real life problems*, dichiarativismo vs. proceduralismo, e così via).

In base alla nostra ricostruzione, risulta inappropriato applicare rigidamente allo sviluppo dell'IA modelli interpretativi della razionalità scientifica come quelli di volta in volta proposti per altre discipline da diversi filosofi della scienza, a partire almeno da Kuhn. Nella nostra ricostruzione, la cornice dell'IA si sviluppa sulla base di assunzioni metodologiche e ontologiche già avanzate dalla cibernetica, e condivise dalle sporadiche e tuttavia vitali esperienze precibernetiche, sviluppatesi all'insegna di quella che Hull definiva ai suoi tempi la «tendenza meccanicistica della psicologia». Questa ha contribuito non poco, come abbiamo visto, all'elaborazione delle cinque tesi che caratterizzano la cornice, ma anche, e scontando i limiti della tecnologia assai primitiva delle macchine dell'epoca, alla prima modellistica del metodo sintetico.

La nostra ricostruzione aveva come obiettivo principale quello di rendere esplicito il ruolo svolto da Turing nella nascita e nell'evoluzione dell'IA. Abbiamo visto che non si potrebbe pensare alla cornice come oggi la conosciamo, con i relativi sviluppi della modellistica (nella versione debole come in quella forte del metodo sintetico), senza il contributo di Turing, che pure si colloca agli esordi degli sviluppi rapidi e tumultuosi della tecnologia hardware e software dei calcolatori. In particolare, il suo contributo ha portato a una svolta decisiva nell'elaborazione delle cinque tesi, caratterizzando una linea di ricerca che sarebbe poi diventata a lungo prevalente in IA, quella dell'IA simbolica. E tuttavia, al di là di quest'ultima, che solitamente vale a Turing il riconoscimento di precursore dell'IA, egli ha prospettato possibilità diverse, che hanno caratterizzato diverse linee di ricerca successive, incluse quelle cosiddette *situata* e *embodied*. Turing ha individuato anche alcuni problemi di fondo, sotto forma di obiezioni alla sua tesi dell'intelligenza meccanica, che riguardano la creatività, la coscienza e altre caratteristiche del mentale che risultano elusive per ogni linea di ricerca dell'IA, *embodied* o meno.

Abbiamo inoltre sottolineato il contributo di Turing all'identificazione di alcune difficoltà epistemologiche di fondo per l'IA, che nascono da concezioni pre-teoriche del comportamento intelligente, e in particolare dalle sue presunte caratteristiche di imprevedibilità e di non conformità a regole. Turing osserva che siffatte concezioni dell'intelligenza si ripercuotono sul problema di valutare il progresso del programma: ogni volta che l'IA consegue un qualche risultato esplicativo o previsionale, in quel preciso momento il comportamento spiegato o

previsto viene derubricato, in quanto comportamento soggetto a regole e pertanto non più intelligente secondo l'appena richiamata concezione pre-teorica. Se da un lato l'IA, come ogni programma di ricerca scientifica, è alla ricerca di criteri di razionalità interna per valutare i propri risultati e il destino della sua cornice generale, da un altro lato la sua stessa strategia esplicativa e previsionale trasforma ogni successo in qualcosa che può soddisfare solo i criteri della razionalità esterna.

È chiaro che il problema di fornire criteri di razionalità interna adeguati per valutare comparativamente i risultati ottenuti dai ricercatori e i loro progressi nell'attività di *problem solving*, individuando elementi di continuità e cumulatività tra vari modelli o dispositivi informatici e robotici, non si scontra solo con le concezioni pre-teoriche dell'intelligenza richiamate da Turing. Il programma di ricerca dell'IA è stato afflitto da persistenti difficoltà di identificare misure di progresso e stagnazione. Si tratta di difficoltà note nella ricerca scientifica e di difficile superamento, come testimonia anche il caso della psicologia (Mecacci 1992). Sul problema dei criteri di valutazione del progresso, e cioè della razionalità interna al programma di ricerca dell'IA, si è sviluppato ed è tuttora in corso un ampio dibattito, che fin dall'inizio ha coinvolto soprattutto alcuni ricercatori di quella che abbiamo chiamato la componente teorica dell'IA, come Allen Newell, Herbert Simon e David Marr<sup>20</sup>.

Infine ricordiamo il collegamento cruciale tra il lavoro che Turing ha svolto nell'ambito della logica e dell'informatica teorica da un lato e i suoi contributi allo sviluppo del programma di ricerca dell'IA dall'altro lato. Le proposte di Turing in merito alla cornice del programma di ricerca dell'IA si basano essenzialmente sulla sua scoperta delle macchine universali. Nel confronto con la precedente tradizione cibernetica, la reinterpretazione che Turing dà delle cinque tesi mette al centro l'elaborazione simbolica e il comportamento plastico derivante dalla capacità che una macchina universale ha di interpretare ed eseguire un'infinità potenziale di programmi. Ancora dopo la morte di Turing nel 1954, queste novità non erano state pienamente recepite nella riflessione filosofica sulle macchine e sull'interazione uomo-macchina. Nel 1956 Günther Anders, nel suo *L'uomo è antiquato*, continuava a sostenere che «caratteristica essenziale della macchina è che essa esegue un unico lavoro specializzato, si esaurisce in questa sua presta-

---

<sup>20</sup> Per una discussione introduttiva sui criteri di razionalità interna discussi in IA rimandiamo il lettore interessato a Cordeschi e Tamburrini 2001, Tamburrini 1997.

zione e trova in essa e soltanto in essa la giustificazione della sua esistenza» (Anders 1956, trad. it. p. 74). Turing ha identificato in questa concezione superata uno dei motivi di resistenza psicologica all'idea di una intelligenza meccanica: «le macchine usate fino a tempi recenti (possiamo dire fino al 1940)» avevano caratteristiche molto limitate; «questo fatto ha incoraggiato la convinzione che le macchine fossero necessariamente limitate a compiti estremamente diretti, forse perfino a quelli solo ripetitivi» (Turing 1948, trad. it. p. 89). La scoperta delle macchine universali impone l'abbandono di questa concezione antiquata delle macchine, ne svela le profonde carenze e schiude le porte alla nuova epoca delle macchine intelligenti.

ROBERTO CORDESCHI, GUGLIELMO TAMBURRINI



*Riferimenti bibliografici*

- Anders, G. (1956), *Die Antiquiertheit des Menschen. Band I: Über die Seele im Zeitalter der zweiten industriellen Revolution*, C.H. Beck, München; trad. it. *L'uomo è antiquato*, vol. I, Bollati Boringhieri, Torino, 2003.
- Brooks, R.A. (1991), "Intelligence without representation", *Artificial Intelligence*, 47, pp. 139-59.
- Copeland, B.J. (a cura di) (2004), *The Essential Turing*, Clarendon Press, Oxford.
- Cordeschi, R. (1998), "'Pensiero meccanico' e giochi dell'imitazione", *Sistemi Intelligenti*, 10, pp. 44-52.
- Cordeschi, R. (2002), *The Discovery of the Artificial: Behavior, Mind and Machines Before and Beyond Cybernetics*, Kluwer, Dordrecht.
- Cordeschi, R. (2006), "Searching in a maze, in search of knowledge", *Lecture Notes in Computer Science*, 4155, Springer, Berlin-Heidelberg, pp. 1-23.
- Cordeschi R. (2007), "AI turns fifty: Revisiting its origins", *Applied Artificial Intelligence*, 21, pp. 259-79.
- Cordeschi, R. e Frixione, M. (2007), "Computationalism under attack", in M. Marraffa, M. De Caro e F. Ferretti (a cura di), *Cartographies of the Mind: Philosophy and Psychology in Intersection*, Springer, Berlin-Heidelberg, pp. 37-49.
- Cordeschi, R. e Tamburrini, G. (2001), "L'intelligenza artificiale: la storia e le idee", in E. Burattini e R. Cordeschi (a cura di), *Intelligenza Artificiale. Manuale per le discipline della comunicazione*, Carocci, Roma, pp. 15-44.
- Cordeschi, R. e Tamburrini, G. (2006), "Intelligent machines and warfare: Historical debates and epistemologically motivated concerns", in L. Magnani e R. Dossena (a cura di), *Computing, Philosophy and Cognition*, College Publications, London, pp. 1-20.
- Craik, K.J.W. (1943), *The Nature of Psychology*, a cura di S.L. Sherwood, Cambridge University Press, Cambridge.
- Crevier, D. (1993), *AI. The Tumultuous History of the Search for Artificial Intelligence*, Basic Books, New York.
- Datteri, E. e Tamburrini, G. (2007), "Biorobotic experiments for the discovery of biological mechanisms", *Philosophy of Science*, 74, pp. 409-30.
- Datteri, E. e Tamburrini, G. (2012), "Robotic weapons and democratic decision-making", in E. Hilgendorf e J.-Ph. Günther (a cura di), *Robotik und Gesetzgebung*, Nomos Verlagsgesellschaft, Baden Baden, **in corso di stampa**.
- Dennett, D.C. (1996), *Kinds of Minds*, Basic Books, New York.
- Galison, P. (1997), *Image and Logic. A material Culture of Microphysics*, University of Chicago Press, Chicago, Ill.
- Gandy, R.O. (1996), "Human versus mechanical intelligence", in P. Millican

- e A. Clark (a cura di), *Machines and Thought*, Clarendon Press, Oxford, pp. 125-36.
- Hodges, A. (1983), *Alan Turing: The Enigma*, Simon and Schuster, New York.
- Hofstadter, D.R. (1979), *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York; trad. it. a cura di G. Trautteur, *Gödel, Escher, Bach: un'eterna ghirlanda brillante*, Adelphi, Milano, 1984.
- Holton, G. (1973), *Thematic Origins of Scientific Thought: Kepler to Einstein*, Harvard University Press, Cambridge, MA.
- Hull, C.L. (1935), "The mechanism of the assembly of behavior segments in novel combinations suitable for problem solution", *Psychological Review*, 42, pp. 219-45.
- Hull, C.L. e Baernstein, H.D. (1929), "A mechanical parallel to the conditioned reflex", *Science*, 70, pp. 14-5.
- Kuhn, Th.S. (2012), *The structure of scientific revolutions*, 50<sup>th</sup> Anniversary Edition, The University of Chicago Press, Chicago, Ill. (1 ed. 1962); trad. it. *La struttura delle rivoluzioni scientifiche*, Einaudi, Torino, 1969.
- Lakatos, I. (1970), "Falsification and the methodology of scientific research programmes", in J. Worrall e G. Currie (a cura di), *Philosophical Papers*, vol. I, Cambridge University Press, 1978; trad. it. in I. Lakatos, *La metodologia dei programmi di ricerca scientifici*, a cura di M. Motterlini, Il Saggiatore, Milano, 1996. 1970 o 1978???
- Laudan, L. (1977), *Progress and its Problems. Toward a Theory of Scientific Growth*, University of California Press, Berkeley; trad. it. *Il progresso scientifico*, Armando, Roma, 1979.
- Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., Wehner, R. (2000), "A mobile robot employing insect strategies for navigation", *Robotics and Autonomous Systems*, 30, pp. 39-64.
- Langton, C.G. (a cura di) (1989), *Artificial Life*, Addison-Wesley, Reading, MA.
- Lenat, D.B. e Feigenbaum, E.A. (1991), "On the thresholds of knowledge", *Artificial Intelligence*, 47, pp. 185-250.
- MacKay, D.M. (1952), "Mentality in machines", *Proceedings of the Aristotelian Society*, Supplements, 26, pp. 61-86.
- Marr, D. (1982), *Vision*, Freeman, San Francisco.
- Mayr, E. (1982), *The Growth of Biological Thought: Diversity, Evolution, and Inheritance*, Belknap Press, Cambridge, MA; trad. it. *Storia del pensiero biologico: diversità, evoluzione, eredità*, Bollati Boringhieri, Torino, 1990.
- Mecacci, L. (1992), *Storia della psicologia del Novecento*, Laterza, Roma.
- Newell, A. (1982), "The knowledge level", *Artificial Intelligence*, 18, pp. 87-

- Newell, A. (1990), *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.
- Numerico, T. (2005), *Alan Turing e l'intelligenza delle macchine*, Franco Angeli, Milano.
- Paige, J.M. e Simon, H.A. (1966), "Cognitive processes in solving algebra word problems", in B. Kleinmuntz (a cura di), *Problem Solving*, Wiley, New York.
- Pylyshyn, Z.W. (1984), *Computation and Cognition. Toward a Foundation for Cognitive Science*, MIT Press, Cambridge, MA.
- Rashevsky, N. (1931), "Possible brain mechanisms and their physical models", *Journal of General Psychology*, 5, pp. 368-406.
- Rosenblueth, A. e Wiener, N. (1945), "The role of models in science", *Philosophy of Science*, 12, pp. 316-21; trad. it. *Il ruolo dei modelli nella scienza*, in Somenzi e Cordeschi 1994.
- Rosenblueth, A., Wiener, N., Bigelow, J. (1943), "Behavior, purpose and teleology", *Philosophy of Science*, 10, pp. 18-24; trad. it. *Comportamento, scopo e teleologia*, in Somenzi e Cordeschi 1994.
- Ross, T. (1935), "Machines that think. A further statement", *Psychological Review*, 42, pp. 387-93.
- Ross, T. (1938), "The synthesis of intelligence. Its implications", *Psychological Review*, 45, pp. 185-9.
- Shah, U. (2011), *Turing's misunderstood imitation game and IBM Watson's success*, AISB 2011 Convention, York, U.K.
- Simon, H.A. (1973), "Does scientific discovery have a logic?", *Philosophy of Science*, 40, pp. 471-80.
- Simon, H.A. (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, MA; trad. it. della II ed. 1981, *Le scienze dell'artificiale*, Il Mulino, Bologna, 1988.
- Somenzi, V. (1968), "Aspetti filosofici del problema della intelligenza artificiale", in *L'automazione elettronica e le sue implicazioni scientifiche, tecniche e sociali*, Atti del convegno, Roma, Accademia nazionale dei Lincei, Quaderno 110, pp. 227-32.
- Somenzi, V. e Cordeschi, R. (a cura di) (1994), *La filosofia degli automi. Origini dell'Intelligenza Artificiale*, Bollati Boringhieri, Torino.
- Steels, L. (1995), "Building agents out of autonomous systems", in L. Steels and R. Brooks (a cura di), *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, Erlbaum, Hillsdale, N.J., pp. 83-121.
- Tamburrini, G. (1997), "Mechanistic theories in cognitive science: The import of Turing's thesis", in M.L. Dalla Chiara *et al.* (a cura di), *Logic and Scientific Methods*, Kluwer, Dordrecht, pp. 239-57.
- Tamburrini, G. (2002), *I matematici e le macchine intelligenti*, Bruno Mondadori, Milano.

- Tamburrini, G. e Datteri, E. (2005), "Machine experiments and theoretical modelling: from cybernetic methodology to neuro-robotics", *Minds and Machines*, 15, pp. 335-58.
- Trautteur, G. (a cura di) (1995), *Consciousness: Distinction and Reflection*, Bibliopolis, Napoli.
- Trautteur, G. e Tamburrini, G. (2007), "A note on discreteness and virtuality in analog computing", *Theoretical Computer Science*, 371, pp. 106-14.
- Turing, A.M. (1936-37), "On computable numbers, with an application to the *Entscheidungsproblem*", in *Proceeding of the London Mathematical Society* (2<sup>nd</sup> serie), 42, pp. 230-65; 43, p. 544.
- Turing, A.M. (1948), "Intelligent machinery", in D.C. Ince (a cura di) *Mechanical Intelligence*, vol. III di *Collected Works of A.M. Turing*, North-Holland, Amsterdam, 1992, pp. 107-31; trad. it. in Turing A.M., *Intelligenza meccanica*, a cura di G. Lolli, Bollati Boringhieri, Torino, 1994.
- Turing, A.M. (1950), "Computing machinery and intelligence", *Mind*, 59, pp. 433-60; trad. it. *Macchine calcolatrici e intelligenza*, in Somenzi e Cordeschi 1994.
- Turing, A.M. (1951), *Can digital computers think?*, in Copeland 2004; trad. it. "Si può dire che i calcolatori automatici pensano?", *Sistemi Intelligenti*, 10, 1998, pp. 21-6.
- Turing, A. M. (1953), *Digital computers applied to games*, in D.C. Ince (a cura di) *Mechanical Intelligence*, vol. III di *Collected Works of A.M. Turing*, North-Holland, Amsterdam, 1992, pp. 161-85.
- van Gelder, T. (1995), "What might cognition be, if not computation", *Journal of Philosophy*, 91, pp. 345-81.
- Wiener, N. (1948), *Cybernetics, or Control and Communication in the Animal and the Machine*, MIT Press, Cambridge, MA; trad. it. della II ed. 1961, *Cibernetica: il controllo e la comunicazione nell'animale e nella macchina*, Il Saggiatore, Milano 1968.



## APPENDICE



# Si può dire che i calcolatori automatici pensano?

---

*Braithwaite*<sup>1</sup>: Oggi siamo qui per discutere se si possa dire che le macchine calcolatrici pensano in un qualche senso proprio del termine. Di solito si considera talmente il pensiero come una peculiarità dell'uomo, e forse di altri animali superiori, che la questione può sembrare troppo assurda per meritare una discussione. Ma, naturalmente, tutto dipende da cosa si debba includere nel concetto di pensiero. La parola è usata per coprire una quantità notevole di attività diverse. Lei, Jefferson, come fisiologo, quali direbbe che siano gli elementi più importanti coinvolti nel pensare?

*Jefferson*<sup>2</sup>: Non credo che ci sia bisogno di sprecare molto tempo sulla definizione di pensiero dal momento che sarebbe difficile andare oltre frasi di uso comune, come avere idee nella mente, meditare, riflettere, deliberare, risolvere problemi o fare supposizioni. I filologi dicono che la parola «uomo» è derivata da un termine sanscrito che significa «pensare», probabilmente nel senso di giudicare tra un'idea e un'altra.

Convengo che non potremmo più usare la parola pensare in un senso che la restringe all'uomo. Nessuno negherebbe che molti animali pensano, benché in un modo molto limitato. Manca loro l'intuito. Per esempio, un cane impara che è sbagliato salire su cuscini o sedie con le zampe inzaccherate, ma lo impara soltanto in quanto condotta che non paga. Non ha nessuna idea della ragione vera, cioè che facendolo rovina la tappezzeria.

La persona media forse si accontenterebbe di definire il pensiero in termini molto generali come considerare idee nella mente, avere

---

Dibattito radiofonico registrato il 10 gennaio 1952, trasmesso dal Terzo Programma della BBC lunedì 14 gennaio 1952 (9.35-10.20) e, in replica, mercoledì 23 gennaio (6.20-7.5) [ndt].

<sup>1</sup> R.B. Braithwaite era fellow del King's College, a Cambridge [ndt].

<sup>2</sup> M.H. Jefferson, fellow della Royal Society, era professore di neurochirurgia all'Università di Manchester. Si tratta dello stesso Jefferson che Turing menziona nella sezione n. 4 di *Computing Machinery and Consciousness* (Turing 1950, p. 445) [ndt].

nozioni nella testa, avere la mente occupata da un problema e così via. Ma è giusto aggiungere che le nostre menti perdono la gran parte del tempo dietro a questioni di nessuna importanza. Si può dire alla fine che pensare sia il risultato generale di avere un sistema nervoso sufficientemente complesso. Quelli molto semplici non procurano alla creatura nessun problema che non sia risolto da un semplice meccanismo riflesso. Pensare allora diventa l'insieme delle cose che accadono nel cervello, cose che spesso terminano in un'azione, ma non necessariamente. Potrei dire che è la somma complessiva di ciò che fa il cervello di un uomo o di un animale. Turing, cosa ne pensi? Hai una definizione meccanica?

*Turing*<sup>3</sup>: Non voglio dare una definizione di pensare, ma se dovessi darla sarei probabilmente incapace di dire al riguardo qualcosa di più del fatto che era una sorta di ronzio che ha avuto luogo nella mia testa. Ma in realtà non vedo affatto la necessità di accordarci su una definizione. La cosa più importante è tentare di tirare una linea di demarcazione tra le proprietà del cervello, o di un uomo, che noi vogliamo discutere, e quelle che non vogliamo discutere. Per fare un esempio limite, non ci interessa il fatto che il cervello ha la consistenza del *porridge* freddo. Non vogliamo dire «Questa macchina è molto dura, perciò non è un cervello, e perciò non può pensare».

Mi piacerebbe suggerire un tipo particolare di *test* che si potrebbe applicare ad una macchina. Potreste chiamarlo un *test* per vedere se la macchina pensa, ma sarebbe meglio evitare di dare per dimostrata, con questo modo di esprimersi, la questione e dire che le macchine che lo superano sono (poniamo) macchine da «dieci». L'idea del *test* è che la macchina deve provare a fingere di essere un uomo, rispondendo alle domande che le vengono poste, e lo supera solo se la simulazione è ragionevolmente convincente. Dev'essere ingannata dalla finzione una parte consistente di una commissione, che non dovrebbe essere pratica di macchine. Non è consentito vedere la macchina – sarebbe troppo facile. Pertanto la macchina viene tenuta in una stanza distante e si permette alla commissione di farle domande che le sono comunicate: la macchina invia a sua volta una risposta dattiloscritta.

*Braithwaite*: Le domande devono essere somme aritmetiche, oppure potrei chiederle cosa ha mangiato a colazione?

---

<sup>3</sup> Al tempo della conferenza Turing era lettore presso l'Università di Manchester. L'anno prima era stato eletto, dietro proposta di M.H.A. Newman e di Bertrand Russell, *fellow* della Royal Society [ndt].



*Turing*: Sì, qualunque cosa. E le domande non devono essere delle domande vere, non più di quanto lo siano quelle in un'aula di tribunale. Capisci cosa intendo. «Io le contesto che sta solo fingendo di essere un uomo» sarebbe perfettamente appropriata. Inoltre si deve consentire alla macchina di fare appello ad ogni genere di espedienti, in modo che appaia più simile ad un uomo, come aspettare un po' prima di dare la risposta, o fare errori di ortografia, ma essa non può fare sgorbi sulla carta, non più di quanto si possano trasmettere sgorbi via telegrafo. Dovremmo, meglio, supporre che ogni commissione debba giudicare un certo numero di volte, e che talvolta abbia realmente a che fare con un uomo e non con una macchina. Questo impedirà di dire ogni volta «Dev'essere una macchina» senza un'appropriata considerazione.

Bene, questo è il mio *test*. Naturalmente non sto dicendo adesso né che delle macchine potrebbero realmente superare il *test*, né che non potrebbero. La mia proposta è proprio che questa è la questione che andrebbe discussa. Non è la stessa cosa che «Le macchine pensano?», ma sembra vicina quanto basta al nostro scopo attuale, e solleva pressappoco le stesse difficoltà.

*Newman*<sup>4</sup>: Mi piacerebbe esserci al tuo incontro tra un uomo e una macchina e magari cercare per parte mia di mettere insieme alcune delle domande. Ma ci vorrà molto, da adesso, se la macchina deve fronteggiare ogni evenienza, nessuna domanda esclusa?

*Turing*: Oh sì, almeno 100 anni, direi.

*Jefferson*: Newman, come potrebbero figurare in questo *test* le macchine attuali? Che genere di cose possono fare adesso?

*Newman*: Naturalmente il genere in cui sono più forti è la computazione matematica per cui furono progettate, ma potrebbero anche risolvere alcune questioni che non sembrano numeriche, ma possono facilmente diventarlo, come risolvere un problema di scacchi o cercarti un treno in un orario ferroviario.

*Braithwaite*: Davvero possono farlo?

*Newman*: Sì. Entrambi questi compiti si possono eseguire vagliando tutte le possibilità, una dopo l'altra. Tutta l'informazione contenuta in un normale orario ferroviario potrebbe essere inserita come parte di un programma, e la *routine* più semplice possibile sarebbe quella che individua i treni da Londra a Manchester, controllando per ogni treno nell'orario se fa scalo in entrambi i posti, e che stampa quelli che




---

<sup>4</sup> M.H.A. Newman, *fellow* della Royal Society, era professore di matematica all'Università di Manchester e direttore del progetto della macchina omonima [ndt].

lo fanno. Naturalmente questo è un metodo che arranca, e si potrebbe migliorarlo usando una *routine* più complicata, ma se ho capito il *test* di Turing in modo corretto, non vi è permesso andare dietro le quinte e criticare il metodo utilizzato, ma ci si deve attenere solo alle risposte corrette riportate, trovate a una velocità ragionevole.

*Jefferson*: Sì, e tuttavia una persona che deve consultare frequentemente l'orario dei treni può riuscirci, e diventare più bravo, man mano che si familiarizza con esso. Supponiamo che io dia di nuovo alla macchina lo stesso problema. Può imparare a fare meglio senza passare per tutta la tiritera di provare ogni cosa da cima a fondo ogni volta? Mi piacerebbe conoscere la vostra risposta perché è un punto davvero importante. Le macchine possono imparare a migliorare con la pratica?

*Newman*: Sì, possono. Forse il problema degli scacchi ce ne fornisce una illustrazione migliore. Innanzitutto accennerei al fatto che *tutta* l'informazione richiesta per ogni compito – numeri, orari dei treni, posizioni di pezzi, o qualsiasi cosa sia, e anche le istruzioni che dicono ciò che deve essere fatto con loro – tutto questo materiale è memorizzato nello stesso modo. (Nella macchina di Manchester è memorizzato come uno schema [*pattern*] su qualcosa che somiglia ad uno schermo televisivo). Con il procedere del lavoro lo schema cambia. Di solito è la parte del modello che contiene i dati che cambia, mentre le istruzioni rimangono fisse. Ma è davvero semplice cambiare anche le istruzioni di quando in quando. Bene, a questo punto si potrebbe scrivere un programma che determinerebbe la macchina a fare questo: si registra un problema di scacchi a due mosse nella macchina in una codifica appropriata, e ogni volta che si mette in funzione la macchina, viene scelta a caso una mossa del bianco (nella nostra macchina c'è un dispositivo per fare scelte casuali). Si analizzano quindi tutte le conseguenze di questa mossa, e se *non* conduce a scacco matto in due mosse, la macchina stampa, diciamo, «P-Q3, mossa sbagliata» e si ferma. Ma l'analisi mostra che quando si sceglie la mossa giusta la macchina non solo stampa, diciamo, «B-Q5, soluzione», ma cambia l'istruzione che richiede una scelta casuale con una che dice «Prova B-Q5». Ne risulta che ogni volta che si riavvia la macchina, essa stamperà immediatamente la soluzione esatta – e questo senza che qualcuno abbia scritto la *routine* sapendo in anticipo quale fosse.

Si potrebbe scrivere una *routine* di questo tipo senza dubbio già adesso e penso che possiamo chiamarlo a ragione apprendimento.

*Jefferson*: Sì, suppongo che lo sia. Gli esseri umani apprendono ripetendo gli stessi esercizi finché non li hanno perfezionati. Naturalmente fanno più di questo, e nondimeno noi impariamo generalmente a tra-

sferire la conoscenza acquisita rispetto a qualcosa, a un altro insieme di problemi, scorgendo pertinenze e relazioni. Imparare significa ricordare. Per quanto tempo una macchina può memorizzare informazione?

*Newman:* Oh, a lungo almeno tanto quanto la vita di un uomo, se viene rinfrescata di tanto in tanto.

*Jefferson:* Un'altra differenza sarebbe che nel processo di apprendimento c'è un'intromissione molto più frequente da parte di insegnanti, genitori o altri, che guidano le arti del sapere. Voi matematici mettete il programma una volta sola nella macchina e glielo lasciate. Non otterreste affatto alcun progresso con gli esseri umani se faceste così. Di fatto, l'unica volta che intervenite nel periodo di apprendimento è al momento degli esami.

*Turing:* È verissimo che quando si sta istruendo un bambino, i suoi genitori e insegnanti intervengono ripetutamente per impedirgli di fare questo o incoraggiarlo a fare quest'altro. Ma non sarà diverso quando si cercherà di insegnare ad una macchina. Ho fatto degli esperimenti per insegnare ad una macchina a fare alcune semplici operazioni, ed è stata necessaria una dose molto grande di intervento di questo tipo prima che io potessi ottenere qualche risultato. In altre parole la macchina ha appreso così lentamente che c'è stato bisogno di una grande quantità di insegnamento.

*Jefferson:* Ma chi stava imparando, tu o la macchina?

*Turing:* Beh, suppongo che lo facessimo entrambi. Se ci deve essere un successo reale si dovrà scoprire come costruire macchine che imparino più velocemente. Si spera anche che ci sia una specie di effetto valanga. Più cose ha appreso, più facile dovrebbe essere per la macchina impararne altre. Nell'imparare a fare una cosa particolare probabilmente potrà anche imparare ad imparare in modo più efficace. Sono incline a credere che quando le si sia insegnato a fare certe cose si scoprirà che alcune altre cose che si è pianificato di insegnarle stanno accadendo senza che ci sia bisogno di alcun insegnamento specifico. Questo accade indubbiamente con una mente intelligente, e se non accade quando si sta insegnando ad una macchina, c'è qualcosa che manca nella macchina. Che cosa pensi delle possibilità di apprendimento, Braithwaite?

*Braithwaite:* Nessuno ha fatto cenno a quella che mi sembra la grande difficoltà riguardo all'apprendimento, dal momento che abbiamo solo discusso dell'apprendimento della soluzione di un problema particolare. Ma la parte più importante dell'apprendimento umano

è apprendere dall'esperienza. Non imparare da un tipo particolare di esperienza, ma essere in grado di imparare dall'esperienza in generale.

Si può facilmente costruire una macchina con un dispositivo a retroazione in modo che la programmazione della macchina venga controllata dalla relazione della sua produzione con alcune caratteristiche dell'ambiente esterno, cosicché il lavoro della macchina in relazione all'ambiente sia auto-correttivo. Ma questo richiede che vi siano alcuni aspetti particolari dell'ambiente a cui la macchina deve adattarsi. La peculiarità degli uomini e degli animali è che essi hanno la facoltà di adattarsi a quasi tutti gli aspetti. L'aspetto a cui ci si adatta in una circostanza particolare è quello a cui l'uomo si dedica e l'uomo si dedica a ciò che *gli interessa*. I suoi interessi sono determinati, in modo consistente, dai suoi appetiti, desideri, pulsioni, istinti, tutte cose che insieme costituiscono la sua «molla del comportamento». Se vogliamo costruire una macchina che rivolga la sua attenzione agli elementi dell'ambiente in cui si trova, in modo che a volte si adatti all'uno e a volte all'altro, sembrerebbe necessario dotarla di qualcosa che corrisponda a un insieme di appetiti. Se si costruisce la macchina per trattarla come nient'altro che una specie di animale domestico, e la si imbuca con problemi particolari, essa non sarà in grado di imparare nei modi diversificati in cui apprendono gli esseri umani. Questo nasce dalla necessità di adattare il comportamento in modo adeguato all'ambiente se si devono soddisfare gli appetiti umani.

*Jefferson*: Turing, hai parlato con grande sicurezza di ciò che sarai in grado di fare. Ne hai parlato come se fosse facilissimo modificare la costruzione della macchina così che reagisca in modo più simile ad un uomo. Ma mi viene in mente che dai tempi di Descartes e Borelli in poi, molti hanno detto che sarebbe stata solo una faccenda di pochi anni, forse 3 o 4 o forse 50, e sarebbe stato creato artificialmente un facsimile di uomo. Sbaglieremmo di sicuro se dessimo l'impressione che queste cose sarebbero di facile realizzazione.

*Newman*: Sono d'accordo che ci stiamo spingendo molto oltre rispetto alle macchine calcolatrici così come sono al momento. Queste macchine hanno appetiti molto limitati, e non possono arrossire quando sono imbarazzate, ma è già molto difficile, e credo un problema molto interessante, scoprire quanto queste macchine effettivamente esistenti possono avvicinarsi al pensare. Anche se ci limitiamo alla parte razionante del pensiero, è lunga la strada dalla soluzione di problemi di scacchi all'invenzione di nuovi concetti matematici o ad una generalizzazione che unifica idee che erano prima d'uso corrente,

ma che non sono mai state messe insieme come casi di una singola nozione generale.

*Braithwaite:* Per esempio?

*Newman:* I tipi diversi di numero. Ci sono gli interi, 0, 1, -2, e così via; ci sono i numeri reali usati per comparare lunghezze, per esempio la circonferenza di un cerchio e il suo diametro; e i numeri complessi che comprendono  $\sqrt{-1}$ ; e così via. Non è del tutto ovvio che questi siano esempi di una cosa sola, il «numero». I matematici greci usavano parole completamente diverse per gli interi e per i numeri reali, e non avevano nessuna idea unica per comprenderli entrambi. È davvero soltanto di recente che la nozione generale di tipi di numero è stata astratta da questi esempi e definita accuratamente. Per fare questo tipo di generalizzazione bisogna possedere la facoltà di riconoscere somiglianze, di vedere analogie tra cose che in precedenza non sono mai state accostate. Non si tratta soltanto di saggiare le cose in relazione a una proprietà specifica e classificarle di conseguenza. Il concetto stesso deve essere costruito, deve essere creato qualcosa, diciamo l'idea di un campo numerico.

*Turing:* Mi sembra, Newman, che quello che hai detto su «esplorare tutte le possibilità» come metodo si spinga abbastanza avanti, anche quando si vuole che una macchina faccia qualcosa di così avanzato come trovare un nuovo concetto utile. Non vorrei dover definire il significato della parola «concetto», né dare regole per valutarne l'utilità, ma qualsiasi cosa siano hanno forme esterne e visibili, che sono parole e combinazioni di parole. Una macchina potrebbe costituire più o meno a caso tali combinazioni di parole, e quindi assegnare loro dei punteggi caratteristici per i loro diversi meriti.

*Newman:* Non si impiegherebbe un periodo di tempo proibitivamente lungo?

*Turing:* Sarebbe di certo terribilmente lento, ma si potrebbe cominciare con cose facili, come riunire pioggia, grandine, neve e nevischio, sotto la parola «precipitazione». Forse potrebbe fare cose più difficili in seguito se continua a imparare come perfezionare i propri metodi.

*Braithwaite:* Non penso che sia molto difficile vedere le analogie che si possono analizzare formalmente e descrivere esplicitamente. Si tratta allora solo di progettare la macchina in modo che possa riconoscere similitudini tra strutture matematiche. La difficoltà sorge se l'analogia è un'analogia vaga su cui non si può dire molto di più che la si avverte, che c'è un qualche genere di somiglianza tra due situazioni ma non si ha nessuna idea riguardo a ciò per cui i due casi sono simili. Una macchina non può riconoscere somiglianze quando non c'è niente nel

suo programma che dica quali sono le somiglianze che ci si deve aspettare che riconosca.

*Turing:* Credo che si possa fare in modo che una macchina sia in grado di vedere un'analogia, anzi questo è un esempio molto buono di come si potrebbero far fare ad una macchina alcune di quelle cose che si considerano di solito come un monopolio essenzialmente umano.

Supponiamo che qualcuno stia cercando di spiegarmi la doppia negazione, che, per esempio, se qualcosa non è non verde dev'essere verde, e che non riesca a farmelo capire. Potrebbe allora dire «Guarda, è come attraversare la strada. L'attraversi, e poi l'attraversi ancora, e sei tornato al punto di partenza». Questa osservazione potrebbe far scattare la molla. Questa è una delle cose che sarebbe bello provare con le macchine e penso che potrebbe accadere anche con loro. Immagino che nei nostri cervelli l'analogia si formi pressappoco così. Quando due o più insiemi di idee hanno la stessa configurazione di connessioni logiche, il cervello può molto probabilmente economizzare delle parti utilizzandone una sola due volte, per ricordare le connessioni logiche che sono le stesse sia in un caso che nell'altro. Si può supporre che alcune parti del mio cervello siano state impiegate una volta per l'idea della doppia negazione e una volta per il doppio attraversamento della strada. In realtà io dovrei sapere entrambe queste cose ma non riesco a capire dove quel tale vuole andare a parare finché continua a parlare di tutti quei noiosi non e non-non. In qualche modo non tocca la parte giusta del cervello, ma non appena tira fuori l'attraversamento della strada ci arriva: però facendo un percorso differente. Se esiste una spiegazione puramente meccanica del modo in cui questo procedere per analogia avviene nel cervello, dev'essere allora possibile far fare la stessa cosa a un calcolatore digitale.

*Jefferson:* Una spiegazione meccanica in termini di cellule e connessioni nel cervello non c'è di certo.

*Braithwaite:* Ma una macchina potrebbe farlo realmente? Come farebbe?

*Turing:* Ho certamente lasciato molto all'immaginazione. Se avessi dato una spiegazione più particolareggiata avrei potuto fare in modo che ciò che stavo descrivendo solo come realizzabile sembrasse di più sicura realizzazione, ma allora voi esclamereste probabilmente con impazienza: «Beh, sì, vedo che una macchina potrebbe fare tutto questo, ma non lo chiamerei pensare». Non appena si può vedere come causa ed effetto si manifestano nel funzionamento del cervello, lo si considera come se non fosse pensare, ma una sorta di banale lavoro-da-asino privo di inventiva. Da questo punto di vista si potrebbe essere tentati di

definire il pensiero come consistente di «quei processi mentali che noi non capiamo». Se questo è giusto allora costruire una macchina pensante è costruirne una che fa cose interessanti senza che noi in realtà comprendiamo davvero come funziona.

*Jefferson:* Se intendi che noi non conosciamo i circuiti di connessione [wiring] negli uomini, è verissimo.

*Turing:* No, non è affatto quello che intendo. Noi conosciamo l'impianto della nostra macchina, ma già lì succede, entro certi limiti. Talvolta una macchina calcolatrice fa qualcosa di molto strano che non ci siamo aspettati. In linea di principio si potrebbe averlo previsto, ma in pratica è di solito troppo complesso. Ovviamente se si dovesse predire tutto ciò che un calcolatore sta per fare, allora si potrebbe anche farne senza.

*Newman:* È verissimo che le persone rimangono deluse quando scoprono che quello che i grandi calcolatori fanno realmente è solo addizionare e moltiplicare, e usare i risultati per decidere quali addizioni e moltiplicazioni ulteriori fare. «Questo non è pensare», è il commento naturale, ma questa è una petizione di principio. Se andate in una delle chiese antiche a Ravenna vedrete alcune bellissime immagini alle pareti, ma se le guardaste attraverso un binocolo potreste dire: «Che diavole, non sono affatto vere immagini, ma solo un certo numero di piccole pietre colorate con cemento in mezzo». I processi delle macchine sono mosaici di parti standard molto semplici, ma i disegni possono essere di grande complessità, e non è ovvio dove sta il limite ai modelli di pensiero che potrebbero imitare.

*Braithwaite:* Ma quante pietre ci sono nel vostro mosaico? Jefferson c'è una quantità sufficiente di cellule nel cervello perché si comporti come una macchina calcolatrice?

*Jefferson:* Certo, ce ne sono migliaia, tonnellate di migliaia di cellule nel cervello più di quante ce ne siano in una macchina calcolatrice poiché una macchina attuale contiene – quante dite?

*Turing:* Mezzo milione di cifre. Penso che possiamo assumere che sia l'equivalente di mezzo milione di cellule nervose.

*Braithwaite:* Se il cervello lavora come una macchina calcolatrice allora l'attuale macchina calcolatrice non può fare tutte le cose che fa il cervello. Sono d'accordo. Ma se si costruisse una macchina calcolatrice che potesse fare tutte le cose che fa il cervello, non richiederebbe più cifre di quelle che trovano spazio nel cervello?

*Jefferson:* Ecco, non saprei. Supponiamo che sia giusto paragonare le cifre in una macchina con le cellule nervose in un cervello. Ci sono diverse stime, si suppone che lì ci sia qualcosa come dieci/quindicimila

milioni di cellule. Nessuno lo sa con certezza. È un numero colossale. Avreste bisogno di 20.000 o più delle vostre macchine per equiparare le cifre con le cellule nervose. Ma non è, sicuramente, solo una questione di quantità. Ci sarebbe troppa logica nella vostra enorme macchina. Non sarebbe veramente come la produzione di pensiero umano. Per renderla più simile, molte parti della macchina dovrebbero essere progettate in maniera molto diversa per garantire maggiore flessibilità e possibilità di impiego più diversificate. È davvero un grado molto alto.

*Turing:* È proprio la quantità che importa in questo caso. È la quantità di informazione che può essere memorizzata. Se si pensa a qualcosa di molto complicato che si vuole far fare a una di queste macchine, si può scoprire che la macchina particolare che abbiamo non lo farà, ma se una macchina può farlo, allora ciò può essere fatto dal vostro primo calcolatore, semplicemente potenziato nella capacità di memoria.

*Jefferson:* Se davvero ci si vuole avvicinare a qualcosa che possa onestamente chiamarsi «pensiero» non si possono tralasciare gli effetti degli stimoli esterni; l'intervento di ogni genere di fattori estranei, come le preoccupazioni della vita, o del pagare le tasse, o di procurarci il cibo che ci piace. Questi non sono in nessun senso fattori minori, sono molto importanti in realtà, e le preoccupazioni relative possono interferire moltissimo con un buon pensiero, specialmente col pensiero creativo. Vedete, la macchina non dispone di un ambiente mentre una persona è in costante relazione col suo ambiente e l'ambiente, per così dire, preme continuamente i suoi tasti mentre l'uomo preme quelli dell'ambiente. C'è un vasto sfondo di ricordi nel cervello di una persona con cui deve accordarsi ciascuna nuova idea o esperienza. Mi stupirei se foste in grado di dirmi fino a che punto un calcolatore si conforma a questa situazione. La maggior parte della gente è d'accordo che la prima reazione dell'uomo a una nuova idea (come quella che stiamo discutendo oggi) è una reazione di rigetto, spesso un rifiuto immediato e scandalizzato di essa. Non vedo come una macchina potrebbe affermare, per così dire: «Professor Newman o Mr. Turing, non mi piace affatto questo programma che avete inserito adesso, e non voglio averci niente a che fare!».

*Newman:* Una difficoltà nel rispondere è quella a cui Turing ha già accennato. Se qualcuno dice «Potrebbe una macchina fare questo, per esempio potrebbe dire “Non mi piace il programma che avete inserito adesso?”», e si scrive puntualmente un programma per fare proprio quella cosa, è probabile che esso tenda ad avere un'aria artificiale e *ad hoc*, e a somigliare più ad un trucco che ad una risposta seria alla domanda. E come quei passi nella Bibbia, che mi spaventavano da ra-



gazzino, che dicono che fu fatto così e così affinché «fosse realizzata la profezia che dice» così e così. Questo mi è sempre sembrato un modo molto sleale di assicurare che la profezia si avverasse. Se io rispondo alla tua domanda, Jefferson, scrivendo una *routine* che semplicemente fa in modo che la macchina dica proprio le parole «Newman e Turing, non mi piace il vostro programma», certamente tu avrai la sensazione che si tratti in realtà di un trucco da bambini, e che non è la risposta a ciò che davvero vuoi. Ma è ancora difficile «inchiodare» ciò che vuoi tu.

*Jefferson:* Io voglio che la macchina rifiuti il problema perché in qualche modo la offende. Questo mi porta a chiedere quali sono le componenti di idee che rigettiamo perché d'istinto non ci piacciono. Non so perché mi piacciono alcune immagini e certa musica e mi annoio per quelle di genere diverso. Ma non svilupperò questo argomento perché siamo tutti diversi, e le nostre antipatie si basano sulle nostre storie personali e probabilmente anche su piccole differenze di costituzione in tutti noi, credo per eredità. Le vostre macchine non hanno nessun gene, nessun albero genealogico. L'eredità mendeliana non significa nulla per valvole senza fili. Ma non mi interessa guadagnare punti nella discussione. Dovremmo chiarire che neanche Turing pensa che tutto quello che deve fare è mettere una pelle sulla macchina, ed è viva! Noi stiamo affrontando un obiettivo più limitato, chiarire cioè se il tipo di cosa che le macchine fanno può considerarsi pensiero. Ma la vostra macchina non è più certa di ogni essere umano di afferrare subito bene il problema, e infallibilmente?

*Newman:* Oh!

*Turing:* I calcolatori non sono affatto veramente infallibili. Effettuare controlli sulla loro accuratezza è una parte molto importante dell'arte di utilizzarli. Oltre a fare errori, talvolta essi alla fine non hanno fatto proprio il calcolo che ci si sarebbe aspettati, e si può verificare qualcosa che potremmo chiamare un «equivoco».

*Jefferson:* Ad ogni modo, non sono influenzati dalle emozioni. Vi basta soltanto innervosire sufficientemente una persona e lei si confonde, non trova le risposte e può fare una gran brutta figura. È il contenuto altamente emotivo dei processi mentali nell'essere umano a renderlo molto diverso da una macchina. Mi sembra che questo gli derivi dalla grande complessità del suo sistema nervoso con le sue  $10^{10}$  cellule e anche dal suo sistema endocrino che interessa ogni sorta di emozioni e istinti, come quelli che hanno a che fare col sesso. L'uomo è essenzialmente una macchina chimica, è molto esposto a fame e stanchezza e fatica, all'essere «a corto di risorse», come diciamo noi, anche a giudizi innati e a impulsi sessuali. Questa parte chimica è tremendamente im-

portante, soprattutto perché il cervello esercita un controllo a distanza sui più importanti processi chimici che avvengono nei nostri corpi. Le vostre macchine non devono preoccuparsi del fatto di essere stanche o infreddolite o felici o soddisfatte. Non mostrano alcuna delizia per aver fatto qualcosa che non hanno mai fatto prima. No, esse sono «mentalmente» cose semplici. Voglio dire che per quanto sia complicata la loro struttura (e io so che è molto complicata) a paragone con l'uomo è semplicissima ed esegue compiti con un'assenza di pensieri fuorvianti che è davvero *inumana*.

*Braithwaite*: Non sono sicuro di essere d'accordo. Credo che sarà necessario dotare la macchina, di qualcosa che corrisponda ad appetiti o ad altre «molle d'azione», affinché presti attenzione a caratteristiche rilevanti del suo ambiente per essere in grado di imparare dall'esperienza. Molti psicologi hanno sostenuto che le emozioni negli uomini sono sotto-prodotti dei loro appetiti e che assolvono una funzione biologica nel chiamare in gioco livelli più alti di attività mentale quando i livelli più bassi sono incapaci di tener testa ad una situazione esterna. Per esempio, non si può avere paura quando non c'è nessun pericolo, oppure c'è un pericolo che può essere evitato in modo più o meno automatico: la paura è un sintomo che mostra che il pericolo deve essere affrontato dal pensiero cosciente. Forse sarà impossibile costruire una macchina capace di imparare in generale dall'esperienza senza che in essa venga incorporato un apparato emozionale, la funzione del quale sarà di passare a parti differenti della macchina quando l'ambiente esterno differisce troppo da ciò che soddisferebbe gli appetiti delle macchine oltre una certa misura. Non voglio suggerire che sarà necessario che la macchina sia in grado di avere un attacco di nervi. Ma gli scatti di rabbia nell'uomo spesso assolvono una funzione precisa, quella di sfuggire alle responsabilità; e per proteggere una macchina contro un ambiente troppo ostile può essere essenziale permetterle, diciamo così, di mettersi a letto con una nevrosi, o una malattia psicogenica, proprio come, in un modo più semplice, essa è dotata di una valvola che salta se la potenza elettrica che tiene in funzione la macchina minaccia la continuazione della sua esistenza.

*Turing*: Beh, io non immagino di insegnare alla macchina a fare scene isteriche. Penso che sia probabile che alcuni effetti del genere si verifichino come una specie di sotto-prodotto dell'insegnamento vero e proprio e che ci si interesserà più di frenare tali ostentazioni che di incoraggiarle. Tali effetti sarebbero probabilmente nettamente diversi da quelli umani corrispondenti, ma riconoscibili come loro varianti.

Questo vuol dire che se la macchina venisse sottoposta a uno dei

miei *test* dell'imitazione, dovrebbe impegnarsi a fare un bel po' di recite, ma in un confronto meno rigido con le persone la rassomiglianza potrebbe essere molto impressionante.

*Newman:* Mi sembra ancora che troppo del nostro argomento riguardi le caratteristiche ipotetiche future delle macchine. Va benissimo dire che si potrebbe facilmente fare in modo che la macchina faccia questo o quello, ma, per considerare solo un aspetto pratico, che cosa dire del tempo che impiegherebbe per farlo? Si impiegherebbero solo un'ora o due per mettere su una *routine* per fare in modo che la nostra macchina di Manchester analizzi tutte le varianti possibili del gioco degli scacchi, e trovare la mossa migliore in quel modo, *se non* v'importa che essa impieghi migliaia di milioni di anni per scorrere la *routine*. Risolvere un problema sulla macchina non vuol dire trovare un modo per farlo da qui all'eternità, ma in un arco di tempo ragionevole. Questo non è soltanto un dettaglio tecnico di cui ci si occuperà nei perfezionamenti futuri. È molto improbabile che gli ingegneri possano mai darci un fattore di più di mille o duemila volte superiore alle nostre velocità attuali. Assumere che programmi che impiegherebbero migliaia di milioni di anni sulle nostre macchine presenti saranno eseguiti in un attimo sulle macchine del futuro, vuol dire spostarsi nel dominio della fantascienza.

*Turing:* Per me questo fattore tempo è l'unica questione che assorbirà tutte le reali difficoltà tecniche della programmazione. Se non si sapesse già che queste cose possono farle i cervelli in un tempo ragionevole si potrebbe pensare che sia senza speranza cercare di farle con una macchina. Il fatto che un cervello *può* farlo sembra suggerire che le difficoltà non possono essere realmente così intrattabili come sembrano adesso.

*Braithwaite:* Sono d'accordo sul fatto che non dovremmo estendere la nostra discussione alla questione se potrebbero essere costruiti calcolatori che facessero tutto ciò che può fare un uomo. Il punto è, *invero*, se possono fare tutto ciò che si chiama pensare in senso proprio. La comprensione di un'immagine contiene elementi di pensiero, ma contiene anche elementi di emozione; e noi non ci occupiamo del fatto se può essere costruita una macchina che proverà emozioni. Stesso discorso per le questioni morali: ce ne occupiamo soltanto in quanto sono questioni anche intellettuali. Non dobbiamo dare alla macchina un senso del dovere né qualcosa che corrisponda a una volontà: ancora meno dobbiamo darle tentazioni per resistere alle quali dovrebbe allora avere un apparato apposito. Tutto quello che deve fare una macchina per pensare è essere capace di risolvere, o di fare un buon tentativo

di risoluzione, di tutti i problemi intellettuali con cui potrebbe essere messa a confronto dall'ambiente in cui essa stessa si trova. Questo ambiente, naturalmente, deve comprendere Turing che fa domande imbarazzanti così come eventi naturali come essere bagnata dalla pioggia o essere scossa da un terremoto.

*Newman:* Mi pare però che sei tu che dici che una macchina non sarebbe capace di imparare ad adattarsi al suo ambiente se non è stata dotata di un insieme di appetiti e compagnia cantante.

*Braithwaite:* Sì, certo. Ma i problemi sollevati da una macchina con appetiti non sono propriamente ciò di cui ci occupiamo oggi. Può darsi che senza di essi non sarebbe capace di apprendere dall'esperienza; ma noi dobbiamo considerare se comunque potrebbe essere capace di apprendere, poiché sono d'accordo che essere capace di apprendere è una parte essenziale del pensare. Così non dovremmo ritornare a qualcosa centrato sul pensiero? Per esempio, una macchina può formare concetti nuovi?

*Newman:* Si possono in realtà formulare due domande possibili sulle macchine e sul pensiero. La prima è, che cosa chiediamo per convenire che la macchina fa *tutto ciò* che chiamiamo pensare? Questo è in realtà ciò di cui abbiamo parlato per la maggior parte del tempo; ma c'è anche un'altra domanda importante e interessante: dove comincia il territorio dubbio? Qual è la cosa *più vicina* al puro calcolare che le macchine attuali forse non possono fare?

*Braithwaite:* E quale sarebbe la tua risposta?

*Newman:* Penso possa essere risolvere problemi matematici per i quali non si conosce nessun metodo, nel modo in cui lo fanno gli uomini; trovare metodi nuovi. Questa è un'aspirazione molto più modesta che inventare nuovi concetti matematici. Ciò che accade quando cerchi di risolvere nuovi problemi nel modo ordinario è che ci pensi per alcuni secondi, o per alcuni anni, mettendo alla prova tutte le analogie che ti vengono in mente con problemi che siano stati già risolti, e a un tratto hai un'idea. La provi in dettaglio. Se non va bene devi aspettare un'altra idea. Questo è un po' come la *routine* degli scacchi, dove si prova una mossa dopo l'altra, ma con una differenza molto importante, che se sono anche moderatamente un buon matematico le idee che mi vengono in mente non sono proprio idee a caso, ma sono pre-selezionate così che c'è una probabilità apprezzabile che dopo alcuni tentativi una di esse avrà successo. Henry Moore dice degli studi che fa per la sua scultura: «Quando il lavoro è più che un esercizio, si verificano salti inesplicabili. È a questo punto che entra la fantasia [*imagination*]». Se una macchina potesse realmente essere portata a

imitare questo salto improvviso su un'idea, credo che chiunque sarebbe d'accordo che ha cominciato a pensare, anche se non ha appetiti e non si preoccupa dell'imposta sul reddito. E supponendo che ci attennessimo a quanto è noto sulla fisiologia del pensiero umano, Jefferson, che significato e peso avrebbe tale prestazione?

*Jefferson:* Sappiamo moltissimo del prodotto finale, il pensiero stesso. I contenuti non sono in tutte le nostre moderne biblioteche e i nostri musei? La psicologia sperimentale ci ha insegnato molto sul modo in cui usiamo la memoria e l'associazione di idee, come colmiamo le lacune nella conoscenza e come improvvisiamo partendo da pochi dati.

Ma come facciamo esattamente a farlo in termini di azioni di cellule nervose non lo sappiamo. Ignoriamo in modo particolare il punto cruciale che hai appena menzionato, Newman, la fisiologia effettiva del salto su un'idea, dell'ispirazione improvvisa. Pensare è chiaramente un'attività motoria delle cellule del cervello, un'ipotesi questa sostenuta dall'esperienza comune che tante persone pensano meglio con una penna in mano che a voce alta o per immaginazione e riflessione. Ma non si è riusciti finora a produrre idee nella mente di un uomo stimolando il suo cervello esposto a stimoli elettrici. Sarebbe stato davvero eccitante se si fosse potuto fare – se si potessero stimolare pensieri originali intervenendo su certe parti. Non si può fare. Neppure l'elettroencefalogramma ci mostra come si svolge il processo di pensare. Non ti può dire a cosa sta pensando un uomo. Noi possiamo tracciare il corso, diciamo, di una pagina di stampa o di un flusso di parole nel cervello, ma alla fine li perdiamo. Se potessimo seguirli ai loro posti di memorizzazione ancora non potremmo vedere come sono riassemblati più tardi come idee. Voi avete il grande vantaggio di sapere come sia fatta la vostra macchina. Noi sappiamo soltanto che abbiamo nel sistema nervoso umano un affare compatto e a suo modo perfetto per il suo compito. Sappiamo moltissimo della sua struttura microscopica e delle sue connessioni. In effetti, sappiamo tutto eccetto come queste miriadi di cellule ci permettono di pensare. Ma, Newman, prima di dire «non soltanto questa macchina pensa ma qui in questa macchina abbiamo anche una controparte esatta degli impianti e circuiti dei sistemi nervosi umani», vorrei chiedere se siano state costruite, o potrebbero esserlo, macchine che sono, per così dire, anatomicamente differenti, e tuttavia produrre lo stesso lavoro.

*Newman:* Il piano logico di tutte loro è piuttosto simile, ma certo la loro anatomia, e io suppongo tu potresti dire la loro fisiologia, varia molto.

*Jefferson:* Sì, l'avevo immaginato – quindi non possiamo assumere che qualcuna di queste macchine elettroniche sia una replica di parte

del cervello umano, anche se si deve ammettere che il risultato delle sue azioni sia una specie di pensiero. Il valore reale della macchina per voi sta nei suoi risultati finali, la sua prestazione, piuttosto che in quello che il suo progetto ci rivela come un modello dei nostri cervelli e nervi. La sua utilità sta nel fatto che l'elettricità viaggia lungo i fili 2 o 3 milioni di volte più velocemente di quanto gli impulsi nervosi passano lungo i nervi. Potete far sì che faccia cose per completare le quali un uomo avrebbe bisogno di migliaia di vite. Ma l'uomo, questa vecchia e lenta carretta, è il solo che ha idee – o così penso io. Sarebbe buffo un giorno, Turing, sentire una discussione, diciamo sul Quarto Programma, tra due macchine sul perché gli esseri umani pensano che esse pensano!

*(Traduzione di Berenice Chimient)* 



