

P R O F I L I

ALAN MATHISON TURING

di Marcello Frixione e Teresa Numerico

ABSTRACT - L'anno scorso in tutto il mondo è stata celebrata la figura di Alan Turing, il genio britannico che ha cambiato radicalmente la storia della scienza e della tecnologia con i suoi sorprendenti risultati. I suoi lavori furono rivoluzionari nei campi più diversi, dalla matematica alla teoria della computabilità, dall'informatica all'intelligenza meccanica, dalla criptanalisi alla biologia matematica. Tra i suoi risultati più notevoli la soluzione negativa del problema della decisione, l'idea di Macchina Universale, la decodifica del codice Enigma durante la II Guerra Mondiale, il progetto di una delle prime macchine elettroniche a programma memorizzato chiamata ACE, una evidente influenza sul progetto di macchina elettronica di von Neumann, la prima ipotesi di strategie per l'intelligenza meccanica in Gran Bretagna, il test per l'intelligenza delle macchine che porta il suo nome, alcuni importanti risultati in biologia matematica. Il presente profilo cerca, sia pure nei limiti di spazio consentiti, di dare conto dei più rilevanti risultati della sua ricerca tentando anche una ricostruzione culturale del contesto nel quale essi maturarono.

0. PICCOLA INTRODUZIONE BIOGRAFICA

1. IL CONTRIBUTO DI TURING ALLA TEORIA DELLA COMPUTABILITÀ

1.1. La nozione di algoritmo e il problema della decisione

1.2. Macchine di Turing e Tesi di Church

1.2.1. Le macchine di Turing

1.2.2. La Tesi di Church

1.3. Macchine universali e problemi indecidibili

1.3.1. La macchina di Turing universale

1.3.2. Problemi indecidibili: il problema dell'arresto per le macchine di Turing

2. IL CALCOLATORE TRA ANALISI DELLE PRESTAZIONI E COMPORTAMENTI INTELLIGENTI

2.1. Il progetto ACE tra universalità e soluzioni pratiche

2.2. Le macchine non organizzate

2.3. Il gioco dell'imitazione: un test sociale per l'intelligenza

3. UNA SORTA DI CONCLUSIONE

4. PICCOLA GUIDA BIBLIOGRAFICA SU TURING

5. BIBLIOGRAFIA

0. PICCOLA INTRODUZIONE BIOGRAFICA

Il 23 giugno 1912 nasceva a Londra Alan Mathison Turing, uno dei più importanti e controversi scienziati del secolo scorso. Figlio di un burocrate imperiale, trascorse parte della sua infanzia e adolescenza nei collegi britannici. Fin da piccolo mostrava un grande interesse per le attività di ricerca scientifica e in particolare per la soluzione dei problemi matematici.

Nella interessante biografia scritta dalla madre poco dopo la sua morte (Sara Turing 1959/2012) si riportano molti dei giudizi dei suoi maestri, alcuni lusinghieri, mentre altri molto critici. Il personaggio Turing fin da ragazzo si prestava poco ad essere inquadrato negli schemi e a sottoporsi alle ferree regole delle scuole private inglesi. Il preside della Sherborne School scriveva alla madre lettere di questo tenore "E' il tipo di ragazzo che costituirà probabilmente un problema in qualsiasi scuola o comunità, essendo in un certo senso piuttosto antisociale. Ma ritengo che nella nostra comunità abbia buone opportunità di sviluppare le sue speciali capacità e allo stesso tempo imparare un po' l'arte di vivere " (Sara Turing, 1959/2012, p. 28, trad. TN).

Alla Sherborne School Turing trascorse gli anni dal 1926 al 1931 per passare poi al King's College di Cambridge a studiare Matematica, un contesto per lui molto più familiare e indicato. Nel 1935 viene eletto Fellow del King's e segue le lezioni di Max Newman sui fondamenti della matematica. E' nell'ambito di questo corso che lavora al famoso articolo "On computable numbers" (di cui tratteremo nei parr. 1.2-3). L'articolo verrà pubblicato tra il 1936 e il 1937. Turing nel frattempo su consiglio di Newman partì per Princeton dove lavorò alla tesi di dottorato sotto la guida di Alonzo Church.

Nel periodo trascorso a Princeton (1936-1938), Turing incontrò sicuramente John von Neumann e venne da lui tanto apprezzato da ricevere una lettera di raccomandazione per la borsa di studio del suo secondo anno accademico trascorso a Princeton. Pur essendo stato invitato da von Neumann a restare a Princeton come suo assistente, Turing rifiutò di restare oltre il secondo anno negli Stati Uniti e rientrò al King's per un anno accademico, quello del 1938-1939. La permanenza all'università fu di breve durata perché ben presto lo scoppio della Seconda Guerra Mondiale richiese il contributo di tutti gli scienziati in grado di collaborare.

Turing fu chiamato a Bletchley Park, il luogo segretissimo dove si decodificavano le macchine elettromeccaniche usate dai Tedeschi per inviare comunicazioni segrete tra i vari comandi. Turing si dedicò alla decodifica di una delle macchine considerate impossibili da decodificare e per questo negletta dalla maggior parte degli scienziati, l'ENIGMA Navale. Turing progettò una macchina per facilitare la decodifica, la *Bombe*, a partire dai risultati ottenuti precedentemente in Polonia. Entro il 1942 Turing ottenne un metodo che permise la decodifica dell'Enigma Navale. Dal Novembre del 1942 al Marzo 1943 trascorse anche cinque mesi negli Stati Uniti; fu in questo periodo che incontrò senz'altro Claude Shannon ai Bell Labs e discusse con lui della teoria dell'informazione alla quale Shannon stava lavorando in quel periodo.

Alla fine della Guerra, accettò la proposta di lavorare per il National Physical Laboratory (NPL) al progetto di una macchina elettronica a programma memorizzato. Tuttavia il suo progetto era molto più ambizioso rispetto a quello degli ingegneri con i quali lo avrebbe dovuto condividere. Turing, infatti, lavorava fin dal principio al progetto di una macchina capace di comportamenti intelligenti e non era interessato

semplicemente alla costruzione di un piccolo dispositivo per eseguire calcoli. La sua insoddisfazione per il procedere dei lavori si manifestò ben presto nella richiesta di un anno sabbatico da trascorrere al suo amato King's College di Cambridge. Il permesso gli viene accordato e tra settembre 1947 e giugno/luglio 1948 Turing lavorò all'università. Non sarebbe mai più tornato al NPL, sebbene avesse promesso il contrario.

Raggiunse invece il suo vecchio maestro Max Newman che lo invitò a lavorare alla Manchester University, da ottobre del 1948, dove si stava costruendo un'altra macchina elettronica a programma memorizzato, il *Mark I*. Questo progetto non era quello di Turing, ed egli si occupò solo del software della macchina. Durante la permanenza a Cambridge e nei primi anni di Manchester le sue ricerche si concentrarono sulla possibilità di ottenere macchine intelligenti, o almeno capaci di svolgere compiti considerati intelligenti da un pubblico di inesperti. Successivamente a Manchester si dedicò alla nuova area di studi di morfogenesi, attualmente considerati come una parte della genetica matematica e che restarono largamente incompiuti al momento della sua precoce scomparsa. In quel periodo frequentò saltuariamente le riunioni del *Ratio Club*, un gruppo di studiosi britannici, provenienti da diverse discipline riuniti a Londra intorno alla figura di John Bates. Turing presentò due volte le sue ricerche: discusse gli studi sull'intelligenza meccanica e quelli sulla morfogenesi, mostrando un decisivo interesse per l'agenda di ricerca di questo gruppo di studiosi eclettici e impegnati (Per maggiori dettagli vedi Husbands e Holland 2008). Gli anni di Manchester furono anche caratterizzati dal processo per omosessualità che lo vide protagonista nel 1952 dal quale uscì con una condanna tramutata in una cura ormonale, ciò che attualmente sarebbe considerata una vera e propria castrazione chimica. Il trattamento lo aveva spinto anche

a vedere uno psicologo e sicuramente contribuì ad aumentare il livello di instabilità del suo carattere.

Il 7 giugno 1954 morì, molto probabilmente suicida, dopo aver avvelenato una mela con il cianuro, memore della storia di Biancaneve che lo aveva da sempre affascinato.

E' difficile dar conto della capacità creativa e della multiforme attività scientifica di Alan Turing, così come è impossibile sottovalutarne il contributo alla scienza del secolo scorso. Le celebrazioni che sono avvenute in tutto il mondo nel 2012 per il centenario della sua nascita hanno confermato la ricchezza della sua eredità e l'ampiezza della sua influenza in tanti campi della scienza. Resta però evidente che il contributo più notevole, insieme più duraturo e più sorprendente che ha davvero reso la sua memoria incancellabile è l'invenzione delle Macchine di Turing e i risultati ottenuti nell'ambito della teoria della computabilità. A questo tema sarà dedicata la prossima sezione.

1. IL CONTRIBUTO DI TURING ALLA TEORIA DELLA COMPUTABILITÀ

1.1. La nozione di algoritmo e il problema della decisione

Al fine di esporre il contributo di Turing a quella che sarebbe poi diventata la teoria della computabilità è necessario introdurre alcune nozioni preliminari. In questo paragrafo presenteremo la nozione intuitiva di algoritmo, e descriveremo il contesto storico-culturale in cui si svilupperà il lavoro di Turing.

In generale, un *algoritmo* è un metodo per la risoluzione di un problema. Un algoritmo consiste di un insieme finito di istruzioni tali da essere applicabili in modo deterministico; inoltre, la soluzione, se esiste, deve poter essere ottenuta in un numero finito di passi. Quindi, gli algoritmi (detti anche *metodi effettivi*) sono procedimenti che

consentono di risolvere un problema in modo completamente meccanico, senza far ricorso ad alcuna forma di creatività o di intuizione.

Possiamo trovare molti esempi di algoritmi nelle matematiche elementari. Sono algoritmi i procedimenti mediante i quali eseguiamo le quattro operazioni aritmetiche, come pure è un algoritmo il procedimento euclideo delle divisioni successive per la ricerca del massimo comun divisore di due numeri diversi da zero. In logica, il metodo delle tavole di verità è un algoritmo che permette di stabilire se una formula del linguaggio proposizionale è o meno una tautologia.

Collegata alla nozione di algoritmo è la nozione di funzione effettivamente calcolabile. Una funzione (nel senso matematico del termine) si dice *calcolabile* (o *computabile*) *in modo algoritmico*, o anche *calcolabile in modo effettivo*, o *effettivamente calcolabile* (*computabile*), se e solo se esiste un algoritmo che, per ciascun argomento, consente di calcolarne il valore.

Nel corso dell'intera storia delle matematiche sono stati sviluppati algoritmi per risolvere problemi della più diversa natura. Tuttavia, soltanto in anni relativamente recenti il concetto stesso di algoritmo è stato fatto oggetto diretto di studio matematico. Ciò è avvenuto negli anni trenta del secolo scorso, nell'ambito delle ricerche sui fondamenti della matematica. Prima di allora ci si era basati su una nozione intuitiva di algoritmo, che coincide sostanzialmente con quella enunciata all'inizio del paragrafo. Ciò fu del tutto sufficiente fin tanto che lo scopo era individuare algoritmi che risolvessero problemi (o classi di problemi) specifici. Ma con le ricerche sui fondamenti della matematica avvenne un radicale cambiamento di prospettiva. Furono poste domande di tipo nuovo, che concernevano la classe dei procedimenti algoritmici nella

sua globalità. Soprattutto nel contesto della scuola formalista (il progetto fondazionalista proposto da David Hilbert nei primi decenni del secolo scorso) diventava cruciale stabilire se esistessero o meno problemi matematici che neppure in linea di principio possono essere risolti con un algoritmo. Ciò si ricollega direttamente al cosiddetto *problema della decisione* (*Entscheidungsproblem*), che fu enunciato da Hilbert alla fine degli anni venti (Hilbert e Ackermann 1928), e che possiamo formulare come segue: dato un sistema logico-formale L , è sempre possibile disporre di un algoritmo che consenta di stabilire se una certa formula f è o meno un teorema di L ? Nel momento in cui la ricerca si indirizzò allo studio della classe di *tutti* i procedimenti algoritmici, tale classe dovette essere caratterizzata in maniera rigorosa, e non fu più sufficiente la definizione intuitiva tradizionale. Nacque così quel settore di ricerca detto *teoria della computabilità* (o della *calcolabilità*) *effettiva* (oppure anche *teoria della ricorsività*), nel quale vengono indagati concetti come quello di algoritmo e di funzione computabile in modo algoritmico.

Turing, allora giovanissimo, diede alcuni contributi fondativi alla nascente teoria della computabilità. Nel 1935 (allora era ventitreenne) venne a conoscenza dell'*Entscheidungsproblem* e decise di affrontarlo, giungendo a ottenere una risposta negativa. Nello stesso periodo, in maniera del tutto indipendente, lo stesso risultato venne ottenuto anche dal logico americano Alonzo Church (Church 1936 e 1936a). Tuttavia, rispetto all'impostazione di Church, Turing seguì una strada diversa e originalissima: per analizzare il concetto di processo algoritmico, egli definì una classe di macchine astratte (quelle che in seguito diverranno note appunto come *macchine di Turing*) e ne studiò le proprietà. Rispetto agli studi coevi sulla computabilità,

L'impostazione di Turing presenta dunque il vantaggio di affrontare il problema in maniera estremamente diretta, senza presupporre nozioni o strumenti matematici preesistenti. Inoltre, la macchine di Turing avranno in seguito un ruolo fondamentale nello studio del calcolo automatico, in quanto possono essere interpretate come modelli idealizzati degli attuali calcolatori digitali. Ogni algoritmo si presta, almeno in linea di principio, ad essere automatizzato, cioè ad essere eseguito da una macchina opportunamente progettata. Pertanto, con lo sviluppo dell'informatica, la teoria della computabilità ha assunto il ruolo di "teoria dei fondamenti" per questa disciplina. Con un lieve anacronismo, potremmo dunque affermare che con le macchine di Turing nasce quella che poi sarà l'informatica teorica. Infine, parte dell'interesse per il lavoro di Turing risiede nelle sue implicazioni per altri ambiti di problemi, in parte da lui stesso affrontati in anni successivi, come lo studio filosofico della mente, la scienza cognitiva, l'intelligenza artificiale e lo sviluppo di modelli meccanici della cognizione.

1.2. Macchine di Turing e Tesi di Church

In questo paragrafo e nel prossimo esamineremo il contributo specifico di Alan Turing alla nascita della teoria della computabilità; in particolare, in questo paragrafo introdurremo quelle che oggi sono note come *macchine di Turing* e presenteremo la cosiddetta *Tesi di Church*, alla cui plausibilità il lavoro di Turing ha dato un apporto fondamentale. Si tenga presente però che quella che segue non è una ricostruzione filologica: ai fini di una maggiore chiarezza, ci siamo discostati in molti punti dalla formulazione originaria di Turing. Per chi fosse interessato a un'analisi diretta e dettagliata dell'articolo di Turing, rimandiamo a (Petzold 2008).

1.2.1. Le macchine di Turing

Turing affrontò il problema di fornire un equivalente rigoroso del concetto intuitivo di algoritmo analizzando l'attività di un essere umano che sta eseguendo un calcolo algoritmico. Egli elaborò il suo modello di calcolo algoritmico definendo una classe di macchine calcolatrici astratte, dette in seguito *macchine di Turing* (d'ora in avanti MT). Seguiamo l'analisi del processo di calcolo com'è condotta in (Turing 1936-37). Un calcolo, osserva Turing, consiste nell'operare su un certo insieme di simboli scritti su di un supporto fisico, ad esempio un foglio di carta. Il fatto che abitualmente venga usato un supporto bidimensionale, osserva Turing, è inessenziale; possiamo quindi assumere, senza nulla perdere in generalità, che la nostra macchina calcolatrice utilizzi per la scrittura un *nastro* monodimensionale di lunghezza potenzialmente illimitata in entrambe le direzioni (tuttavia, come vedremo, in ogni fase del calcolo la macchina potrà disporre soltanto di una porzione finita di esso). Assumiamo inoltre che tale nastro sia suddiviso in celle, in "quadretti", "come un quaderno di aritmetica per bambini" dice Turing. Ciascuna cella potrà ospitare un solo simbolo alla volta (fig. 1).

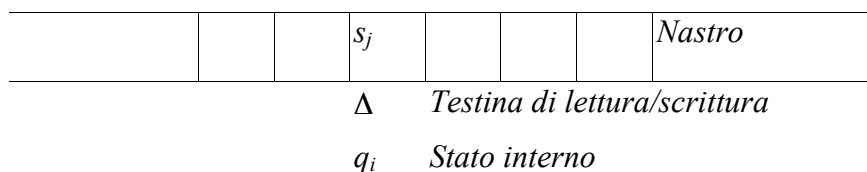


Figura 1

Quanto ai simboli da utilizzare per il calcolo, ogni macchina potrà disporre soltanto di un insieme finito di essi, che costituisce l'*alfabeto* di quella macchina (e che

chiameremo Σ). Il fatto che l'alfabeto di cui si può disporre sia finito non costituisce una grave limitazione. È infatti sempre possibile rappresentare un nuovo simbolo mediante una sequenza finita di simboli dell'alfabeto. Sia dunque $\Sigma = \{s_1, s_2, \dots, s_n\}$ l'alfabeto di una generica MT. Ogni cella del nastro potrà contenere uno di tali simboli, oppure, in alternativa, restare vuota (indicheremo con s_0 la cella vuota).

Vi è senza dubbio un limite al numero di simboli scritti che un calcolatore umano può osservare senza spostare lo sguardo. Secondo Turing si può assumere senza perdita di generalità che la macchina possa esaminare soltanto una cella alla volta, e “osservare” dunque ad ogni passo al più un singolo simbolo. A tal fine la macchina sarà dotata di una *testina di lettura*, che, in ogni fase del calcolo, sarà collocata su di una singola cella (in fig. 1 la testina è rappresentata da Δ). Per poter accedere alle altre celle del nastro, la testina dovrà spostarsi verso destra o verso sinistra. Chi sta eseguendo un calcolo ha poi la possibilità di scrivere nuovi simboli, di cancellare quelli già scritti o di sostituirli con altri. La testina eseguirà anche tali compiti di cancellazione e di scrittura. Anche in questo caso però essa potrà agire soltanto sulla cella “osservata”, e, per accedere ad altre celle, dovrà prima spostarsi lungo il nastro. Poiché ogni cella può contenere un solo simbolo, quando in una cella si scrive un nuovo simbolo, il simbolo eventualmente già presente verrà cancellato.

Nell'eseguire un calcolo, un essere umano tiene conto delle operazioni che ha già eseguito e dei simboli che ha osservato in precedenza sfruttando la propria memoria (cambiando cioè il proprio “stato mentale”). Per simulare ciò, supporremo che una macchina possa assumere, nel corso del calcolo, un certo numero di *stati interni* (uno e non più di uno alla volta), che corrispondano agli “stati mentali” del calcolatore umano.

Tali stati saranno sempre in numero finito, poiché, usando le parole dello stesso Turing, “se ammettessimo un’infinità di stati mentali, alcuni di essi sarebbero ‘arbitrariamente prossimi’, e sarebbero quindi confusi”. Siano allora q_1, \dots, q_m gli stati che una generica MT può assumere. Nella rappresentazione grafica indicheremo sotto la testina di lettura/scrittura Δ lo stato della macchina nella fase di calcolo rappresentata (come in fig. 1).

Definiamo *configurazione* di una MT in una data fase di calcolo la coppia ordinata costituita dallo stato in cui la macchina si trova e dal simbolo osservato dalla testina. Ad esempio, la configurazione della macchina rappresentata nella fig. 1 è (q_j, s_j) .

Una MT può dunque eseguire operazioni che consistono nello spostare la testina lungo il nastro, nella scrittura e nella cancellazione di simboli, e nel cambiare lo stato interno. Scomponiamo tali operazioni in *operazioni atomiche*, tali cioè da non poter essere ulteriormente scomposte in operazioni più semplici. Ogni operazione di una MT può essere scomposta in un numero finito delle operazioni seguenti: (1) sostituzione del simbolo osservato con un altro simbolo (eventualmente con S_0 ; in tal caso si ha la cancellazione del simbolo osservato), e/o (2) spostamento della testina su di una delle celle immediatamente attigue del nastro. Ognuno di tali atti può inoltre comportare (3) un cambiamento dello stato interno della macchina. Dunque, nella sua forma più generale, ogni operazione atomica dovrà consistere di un’operazione di scrittura e/o di uno spostamento della testina sotto una delle celle attigue, ed eventualmente di un mutamento di stato. Indicheremo d’ora in avanti rispettivamente con le lettere S , D e C il fatto che la testina debba eseguire uno spostamento di una cella verso sinistra, verso destra, oppure non debba eseguire alcuno spostamento (C sta per “centro”). Grazie a ciò

potremo rappresentare ogni operazione atomica mediante una terna, il primo elemento della quale è il simbolo che deve essere scritto sulla cella osservata, il secondo è lo spostamento della testina che deve essere eseguito (*S*, *D* o *C*), il terzo è lo stato che la macchina deve assumere alla fine dell'operazione.

Ogni singola MT è “programmata” per eseguire un tipo di calcolo specifico, dispone cioè di una serie di istruzioni che le permettono di eseguire il compito per il quale è stata progettata. Abbiamo insistito sul fatto che un calcolo algoritmico deve essere deterministico: in ogni fase del calcolo deve essere possibile effettuare al più un solo passo, ossia applicare una sola istruzione. Nel caso di un calcolatore umano, ogni mossa deve dipendere esclusivamente dal ricordo delle operazioni già eseguite e dai simboli che egli può osservare (e non, ad esempio, dall'inventività, dalla fantasia o dal caso). In una MT, in ogni fase del calcolo, la macchina “sa” soltanto in quale stato si trova e qual è il simbolo sulla cella osservata (cioè, “sa” qual è la sua configurazione). Inoltre, ogni operazione può essere scomposta in operazioni atomiche. Di conseguenza, ogni istruzione di una MT avrà la forma seguente:

configurazione → operazione atomica

In altri termini, ogni istruzione deve specificare quale operazione atomica deve essere eseguita in corrispondenza di una determinata configurazione. Ad esempio, la seguente è un'istruzione di una MT:

$$q_j, s_j \rightarrow s_{j'} D q_{i'}$$

Essa deve essere interpretata come segue: qualora la macchina si trovi nello stato q_i ed il simbolo osservato sia s_j , allora si deve scrivere sul nastro il simbolo $s_{j'}$, al posto di s_j , la testina dovrà spostarsi di una cella verso destra e la macchina dovrà assumere lo stato $q_{i'}$. In generale, poiché ogni configurazione è rappresentabile mediante una coppia, e ogni operazione atomica mediante una terna, un'istruzione avrà la forma di una *quintupla* (di solito il simbolo “ \rightarrow ” viene omissso e considerato sottinteso). Le istruzioni di cui dispone ogni singola MT per eseguire il calcolo per il quale è stata progettata avranno quindi la forma di un opportuno insieme finito di quintuple, che viene detto la *tavola* della MT. Una volta fissato l'alfabeto, ciò che caratterizza ogni singola MT rispetto a tutte le altre è appunto la tavola delle sue quintuple. Affinché un insieme di quintuple costituisca la tavola di una MT è indispensabile che venga rispettata la seguente condizione: poiché il calcolo deve essere deterministico, a partire da una singola configurazione non devono essere applicabili istruzioni diverse. Ciò corrisponde alla condizione che, nella tavola di una macchina, non possano comparire più quintuple con i primi due elementi uguali.

Affinché il calcolo possa terminare, è necessario che ad alcune delle configurazioni possibili non corrisponda alcuna quintupla, altrimenti, qualunque fosse il risultato di una mossa, esisterebbe sempre un'altra istruzione che si può applicare, e il calcolo non giungerebbe mai a termine. Chiameremo *configurazioni finali* tali configurazioni.

I dati vengono forniti a una MT sotto forma di una sequenza finita di simboli dell'alfabeto scritti sul nastro prima dell'inizio del calcolo. Chiameremo *input* tale sequenza di simboli. Il risultato è costituito da ciò che è scritto sul nastro al momento

della fermata, e ciò costituisce l'*output* del calcolo. Stabiliamo convenzionalmente che all'inizio del calcolo la testina debba essere collocata in corrispondenza del primo simbolo a sinistra dell'input, e che lo stato interno della macchina debba essere q_1 . Quando la testina si trova in corrispondenza del primo simbolo a sinistra dell'input diremo che è in *posizione standard*.

Vediamo un esempio semplicissimo di MT. Si consideri l'alfabeto $\Sigma = \{\}$, composto come unico simbolo da una barra verticale. Definiamo una macchina che, presa come input una successione di barre consecutive, restituisca come output tale successione aumentata di un elemento. A tal fine è sufficiente disporre di due stati interni, q_1 e q_2 .

La tavola della macchina è costituita dalle due istruzioni seguenti:

1. q_1 | | D q_1
2. q_1 s_0 | C q_2

Secondo le convenzioni stabilite, alla partenza la testina deve essere collocata sul primo simbolo a sinistra dell'input, e lo stato di partenza deve essere q_1 (fig. 2).

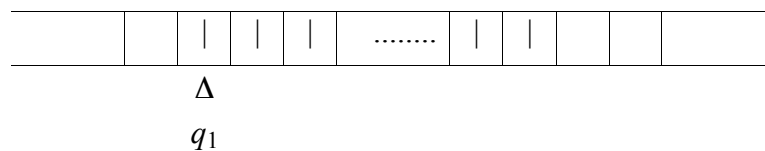


Figura 2

Fintanto che la testina trova celle segnate con |, allora, in virtù della prima quintupla, viene riscritto | sulla cella osservata (cioè, vengono lasciate le cose come stanno), e la

testina si sposta a destra di una cella mantenendo lo stato q_1 . Quando la testina incontra una cella vuota viene applicata la seconda quintupla: la macchina segna con una barra la cella osservata, non esegue alcuno spostamento, ed assume lo stato q_2 . Siccome nella tavola non esiste nessuna quintupla che inizia con q_2 e $|$ - ossia, in questa macchina $(q_2, |)$ è una configurazione finale - la macchina si ferma.

È importante notare che l'esistenza di almeno una configurazione finale è una condizione necessaria, ma non sufficiente, perché il calcolo di una MT termini. È facile infatti costruire MT le quali, sebbene nella loro tavola figurino una o più configurazioni finali, danno origine a un calcolo che procede all'infinito per alcuni possibili input (o, al limite, per tutti).

Fin qui abbiamo considerato MT che eseguono calcoli manipolando simboli generici. Si possono tuttavia codificare i numeri naturali con l'alfabeto di una MT, in modo da definire macchine che calcolino funzioni aritmetiche.

Utilizziamo ancora l'alfabeto $\Sigma \equiv \{\}$. La maniera più semplice per codificare un numero naturale n consiste nel rappresentarlo per mezzo di una sequenza di $n + 1$ barre scritte in celle adiacenti: una barra costituirà la codifica di 0, due barre costituiranno la codifica di 1, e così via. A questo punto, possiamo codificare sul nastro una n -pla di numeri naturali (k_1, \dots, k_n) scrivendo la sequenza di barre corrispondente ad ogni k_i e lasciando una cella vuota come separatore tra le sequenze. Una MT che calcola una funzione aritmetica sarà una MT che prende in input codifiche numeri naturali (o di n -uple di numeri naturali) e a sua volta produce in output codifiche di numeri naturali.

Diremo allora che una macchina M_φ *calcola* (*computa*) una funzione aritmetica φ ad n argomenti *se solo se*, per ogni n -pla (x_1, \dots, x_n) di numeri naturali, se M_φ prende in input

la codifica di (x_1, \dots, x_n) , allora essa produce in output la codifica del numero naturale y tale che $\varphi(x_1, \dots, x_n) = y$.

Diremo inoltre che una funzione φ è *T-computabile* se e soltanto se esiste una MT M_φ che la computa.

Si noti che la MT dell'esempio precedente può essere vista come una macchina che computa una funzione aritmetica. In questo caso la funzione calcolata è la funzione *successore*, ossia la funzione s tale che $s(x) = x + 1$. Di conseguenza, si può concludere che la funzione *successore* è T-computabile. Risultano essere T-computabili tutte le usuali funzioni aritmetiche, come ad esempio addizione, sottrazione, moltiplicazione, elevamento a potenza, e così via.

1.2.2. La Tesi di Church

L'analisi della nozione di calcolo algoritmico sviluppata da Turing ha avuto un ruolo centrale nel sostenere la plausibilità della cosiddetta Tesi di Church. Come abbiamo anticipato nel paragrafo precedente, la teoria della computabilità è nata con l'intento di fornire un equivalente rigoroso della nozione intuitiva di procedimento algoritmico. Ebbene, usualmente si assume che tale caratterizzazione possa essere ottenuta identificando la classe delle funzioni calcolabili con un algoritmo con la classe delle funzioni calcolabili per mezzo di una MT. Ossia, si assume che valga il seguente enunciato:

una funzione è effettivamente calcolabile se e solo se è T-computabile

Si tratta appunto della *Tesi di Church* (d'ora in avanti TdC). Un modo più informale ma sostanzialmente equivalente per enunciare la TdC è il seguente:

*qualunque compito che può essere svolto da un algoritmo
può essere svolto da una MT e viceversa*

Quando nel 1936 Alonzo Church enunciò per la prima volta la TdC (Church 1936) non fece riferimento alle MT, ma utilizzò un sistema equivalente, il cosiddetto λ -calcolo. Tuttavia, come vedremo tra poco, l'analisi di Turing ha svolto un ruolo fondamentale nell'avvalorare la TdC; proprio per riconoscere questo fatto essa talvolta viene indicata anche come *Tesi di Church-Turing*.

Che ogni funzione T-computabile sia effettivamente computabile è facile da constatare, e segue banalmente dalla definizione di MT. Ciò che invece è interessante nella TdC è l'implicazione inversa, secondo la quale ogni procedimento algoritmico è riconducibile alla T-computabilità. Come abbiamo già detto, “algoritmo” e il correlato “funzione calcolabile in modo effettivo” sono concetti intuitivi, non specificati in modo inequivocabile, per cui non è possibile una dimostrazione rigorosa di equivalenza del secondo con il concetto di funzione T-computabile, che invece è un concetto definito in modo rigoroso. La TdC non è dunque una congettura che, in linea di principio, potrebbe un giorno diventare un teorema. Tuttavia, la nozione intuitiva di algoritmo è contraddistinta da un insieme di caratteristiche (quali determinismo, finitezza delle istruzioni, e così via) che possiamo considerare in larga misura “oggettive”. Questo fa sì che sia praticamente sempre possibile una valutazione concorde nel decidere se un dato procedimento di calcolo debba essere considerato algoritmico o meno. Quindi, almeno in linea di principio, è concepibile che venga “scoperto” un controesempio alla TdC,

che si individui cioè una funzione effettivamente calcolabile secondo questi parametri informali, ma che non sia allo stesso tempo T-computabile. Esporremo qui le ragioni per cui si ritiene molto improbabile che un evento del genere si possa verificare.

È opportuna una precisazione. Le funzioni T-computabili sono esclusivamente funzioni aritmetiche, cioè funzioni che hanno come argomenti e valori numeri naturali. Ciò potrebbe sembrare troppo restrittivo, in quanto esistono algoritmi definiti su oggetti diversi dai numeri naturali (ad esempio algoritmi che manipolano le espressioni simboliche di un sistema formale). Tuttavia sono state sviluppate tecniche mediante le quali algoritmi di tipo diverso, come quelli sopra citati, possono essere ricondotti ad algoritmi che calcolano funzioni aritmetiche. Infatti, mediante opportune codifiche, i numeri naturali possono essere utilizzati per rappresentare dati o informazioni di varia natura, purché di tipo discreto.

Una prima fonte di evidenza a favore della TdC è quella che possiamo chiamare *evidenza euristica*. In sintesi, consiste nel fatto che è stato studiato un grande numero di metodi di tipo algoritmico, e tutti sono risultati riconducibili alla T-computabilità. In altri termini, la TdC ha resistito (e tuttora resiste) a tutti i tentativi di falsificazione. Un secondo gruppo di argomenti pone l'accento sull'*equivalenza delle diverse caratterizzazioni rigorose di algoritmo* che sono state proposte. Tutti i tentativi che sono stati elaborati per caratterizzare in modo rigoroso la classe di tutte le funzioni effettivamente calcolabili si sono rivelati tra loro equivalenti (ed equivalenti alle MT).

Un ruolo privilegiato nella corroborazione della TdC è giocato però dall'analisi del concetto di calcolo algoritmico compiuta da Turing. L'impostazione di Turing si distingue in quanto le MT non sono un concetto matematico elaborato per ragioni

autonome e impiegato in un secondo tempo come caratterizzazione rigorosa del concetto di algoritmo: esse costituiscono un tentativo diretto di sviluppare un modello dell'attività di un essere umano che esegue un calcolo deterministico. Storicamente, fu proprio l'analisi compiuta da Turing ad aumentare notevolmente la plausibilità della TdC (tant'è vero che, come abbiamo già ricordato, essa viene detta talvolta *Tesi di Church-Turing*).

Prima di concludere questo paragrafo sulla TdC è importante una precisazione: la TdC non afferma che le MT possono eseguire *qualsiasi algoritmo*. Ciò sarebbe banalmente falso. Ad esempio, le MT sono dispositivi di calcolo sequenziali, che cioè eseguono le operazioni sequenzialmente, una alla volta. Esistono invece modelli di calcolo ed algoritmi di tipo *parallelo*, in cui unità di calcolo diverse operano contemporaneamente sui dati, comunicando tra loro e scambiandosi i risultati. E' chiaro che una MT non potrà mai eseguire un algoritmo parallelo. Quello che la TdC invece afferma è che, per qualsiasi algoritmo (fosse anche, ad esempio, un algoritmo parallelo), esiste una MT che *fa la stessa cosa*, ossia che calcola la stessa funzione.

1.3. Macchine universali e problemi indecidibili

In questo paragrafo esamineremo brevemente le più notevoli conseguenze teoriche della proposta di Turing, ossia la possibilità di definire macchina di Turing universale, e la dimostrazione dell'esistenza di problemi indecidibili (vale a dire problemi che non possono essere risolti per mezzo di alcun algoritmo).

1.3.1 La macchina di Turing universale

Le MT come sono state caratterizzate nel paragrafo precedente sono in grado di effettuare un solo tipo di calcolo; ogni macchina è dotata cioè di un insieme di quintuple che le consentono di svolgere un singolo compito, ossia di calcolare una singola funzione (come ad esempio il successore, l'addizione, la sottrazione, la moltiplicazione, eccetera). Esiste tuttavia la possibilità di definire una MT, detta *Macchina di Turing Universale* (d'ora in poi MTU), che è in grado di simulare il comportamento di ogni altra MT. Ciò è reso possibile dal fatto che le quintuple di ogni MT possono essere codificate in modo da poter essere scritte a loro volta sul nastro di una MT. Infatti è possibile codificare mediante numeri naturali la tavola di una qualsiasi MT. Ciò consente di scrivere il codice di una MT sul nastro di un'altra MT, in modo da fornirglielo in input. Inoltre, tale codifica si può definire in maniera tale che, dato un codice, si possa ottenere la tavola corrispondente mediante un procedimento algoritmico, e, viceversa, data la tavola si possa ottenere alitmicamente il suo codice. Una codifica che goda di questa proprietà è detta una *codifica effettiva*.

A questo punto, si può dimostrare che esiste una MT, ossia la macchina di Turing universale MTU, che, preso in input il codice effettivo di una qualsiasi altra MT, ne simula il comportamento. In altre parole, la MTU è una macchina il cui input è composto da due parti (si veda la parte superiore di fig. 3):

1. la codifica della tavola di una MT che chiamiamo M
2. un input per M che chiamiamo n

Per ogni M e per ogni n , la MTU “decodifica” le quintuple di M , e le applica ad n , ottenendo lo stesso output che avrebbe ottenuto M a partire da n (come schematizzato nella parte inferiore di fig. 3).

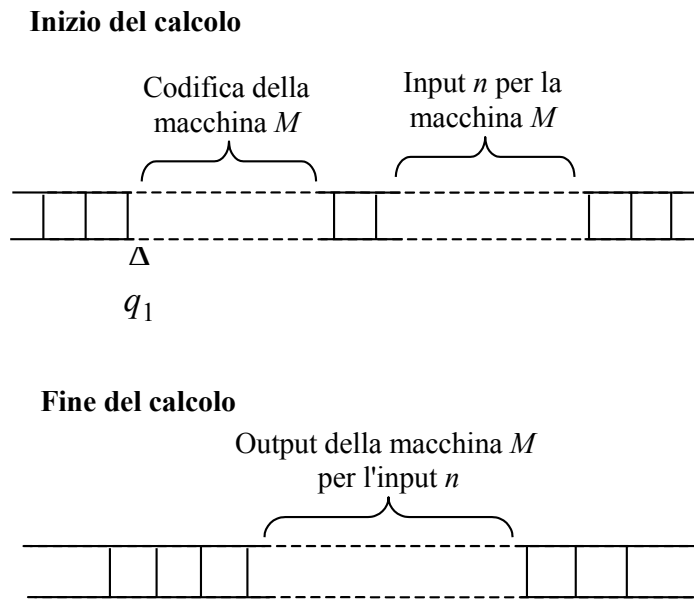


Figura 3

La dimostrazione che esiste la MTU è lunga e complessa (la tavola della MTU contiene un numero molto elevato di istruzioni). Tuttavia qui possiamo accontentarci di giustificare questo risultato usando la TdC come “scorciatoia”. Consideriamo il seguente procedimento:

- 1) Prendi in input il codice di una macchina M e un input n per M
- 2) Decodifica M , ossia, a partire dal codice, ottieni le quintuple della macchina M
- 3) Esegui il calcolo della macchina M con input n , e ottieni l’output corrispondente O
- 4) Produci in output O

L’operazione che corrisponde a 2) è un’operazione algoritmica in quanto per codificare le MT abbiamo utilizzato una codifica effettiva. L’operazione che corrisponde a 3) è anch’essa algoritmica, perché le MT sono definite in modo da eseguire solo calcoli algoritmici. Di conseguenza, l’intero procedimento risulta dalla composizione di procedimenti algoritmici, ed è quindi a sua volta un procedimento algoritmico. Per la

TdC, se un procedimento è algoritmico, allora esiste una MT che “fa la stessa cosa”, ossia che, a partire dagli stessi dati, produce il medesimo risultato. Quindi, esiste una MT che esegue il calcolo di questo procedimento: si tratta appunto della MTU.

Ribadiamo ancora una volta che qui l’appello alla TdC è solo una questione di comodità, e non è essenziale alla dimostrazione: l’esistenza della MTU si può dimostrare in maniera diretta esibendone la tavola (come ha fatto Turing nel suo articolo del 1936), senza fare alcun riferimento alla TdC. Quindi, il fatto che esista la MTU non dipende dal fatto di accettare la TdC. Dalla TdC segue invece che, poiché la MTU è in grado di simulare il comportamento di qualsiasi MT, allora essa è in grado di calcolare qualsiasi funzione che sia calcolabile mediante un algoritmo.

Ciò che caratterizza la MTU rispetto alle MT usuali è costituito dal fatto di essere una macchina *programmabile*. Mentre infatti le normali macchine di Turing eseguono un solo programma, che è “incorporato” nella tavola delle loro quintuple, la MTU assume in input il programma che deve eseguire (cioè, la codifica delle quintuple della MT che deve simulare). Le quintuple che compongono la sua tavola hanno esclusivamente la funzione di consentirle di interpretare e di eseguire il programma ricevuto in input.

Una caratteristica importante della MTU consiste anche nel tipo di trattamento riservato ai programmi. La MTU tratta i programmi (cioè la codifica delle quintuple della MT da simulare) e i dati (l’input della MT da simulare) in maniera sostanzialmente analoga: essi vengono memorizzati sullo stesso supporto (il nastro), rappresentati utilizzando lo stesso alfabeto di simboli ed elaborati in modo simile. Come vedremo nel paragrafo 2.1, questo aspetto si è rivelato in seguito di importanza cruciale per la tecnologia dei calcolatori digitali.

1.3.2. Problemi indecidibili: il problema dell'arresto per le macchine di Turing

L'analisi del concetto di algoritmo proposto da Turing ha importanti conseguenze dal punto di vista teorico. In particolare, da essa segue l'esistenza di *problemi indecidibili*, ossia problemi che non sono risolubili mediante alcun algoritmo. In particolare, Turing ha dimostrato che non è effettivamente decidibile il *problema dell'arresto* (*halting problem*) per le MT, cioè il problema di stabilire se, data una MT M e dato un input n , M con input n dà origine a un calcolo che termina o meno. Da ciò segue una risposta negativa all'*Entscheidungsproblem* di Hilbert: il problema dell'arresto può essere formulato come un problema aritmetico; sia allora L un sistema logico che formalizza l'aritmetica (ad esempio un sistema formale basato sugli assiomi di Peano formalizzati nella logica dei predicati del primo ordine); a questo punto L non può essere decidibile, perché se lo fosse dovrebbe essere decidibile anche il problema dell'arresto.

L'indecidibilità del problema dell'arresto per le MT comporta che non esista alcun algoritmo che, data una generica MT (o il suo codice secondo un'opportuna codifica effettiva) e dato un generico input per essa, consenta sempre di stabilire se il calcolo di quella macchina con quell'input termina o meno.

La dimostrazione dell'indecidibilità del problema dell'arresto per le MT ha molte conseguenze sul piano teorico. Da essa segue l'indecidibilità di altri problemi, ed è strettamente collegata a risultati limitativi della logica matematica, tra i quali i teoremi di incompletezza di Gödel. In informatica, il risultato comporta che esistono problemi i quali non possono, neppure in linea di principio, essere risolti da un calcolatore (ad esempio, non può esistere un programma che sia sempre in grado di stabilire se un

programma scritto in un certo linguaggio con un certo input terminerà o no il suo calcolo).

2. IL CALCOLATORE TRA ANALISI DELLE PRESTAZIONI E COMPORTAMENTI INTELLIGENTI

Il lavoro teorico di Turing ebbe velocemente la possibilità di essere usato in contesti pratici, sia perché fin dagli anni trascorsi a Princeton Turing cominciò a intravedere le possibilità applicative del suo concetto astratto di macchina, sia perché l'arrivo della Guerra impose anche agli scienziati di convogliare tutte le ricerche ai fini della preparazione della sconfitta del nemico. Fu in questo contesto di mobilitazione che Turing venne coinvolto dal suo maestro di Cambridge Max Newman nel progetto di decodifica dell'ENIGMA Navale, una complessa macchina elettromeccanica usata dai tedeschi per mandare messaggi cifrati. Il lavoro della decodifica a Bletchley Park, il quartier generale dell'intelligence inglese durante la Guerra, fu molto importante per l'evoluzione delle sue posizioni intellettuali a proposito dell'idea stessa di macchina e in particolare di una macchina intelligente.

2.1. Il progetto ACE tra universalità e soluzioni pratiche

Dopo l'esperienza di Bletchley Park, Turing decise di non tornare a Cambridge. Come abbiamo già accennato, accettò la proposta di John Womersley di dirigere la sezione del *National Physical Laboratory* (NPL) che gli offrì la possibilità di lavorare al progetto di costruzione di un calcolatore elettronico a programma memorizzato, chiamato ACE (Automatic Computing Engine). Questa decisione si spiega riconoscendo che l'esperienza della guerra aveva cambiato profondamente le sue priorità, i suoi interessi e i suoi progetti. Tornare ai corsi sui fondamenti della matematica sarebbe stato del tutto

anacronistico per lui. Era entrato a Bletchley Park come il “tipo accademico”, ma ne usciva come progettista di dispositivi elettronici con il pallino di realizzare una macchina intelligente.

Turing lavorò formalmente alla proposta della macchina ACE dall’ottobre al dicembre del 1945, ma forse cominciò a dedicarsi al progetto già dall’estate del 1945, subito dopo che gli venne offerto un lavoro al *National Physical Laboratory*. Scrisse il documento (Turing 1945) relativo alla costruzione del suo calcolatore completamente da solo, facendo riferimento unicamente al *First Draft*, il documento che von Neumann aveva preparato, frutto degli incontri del gruppo di lavoro della Moore School, per descrivere il modello di un calcolatore a programma memorizzato, EDVAC (Electronic Discrete Variable Automatic Computer) (von Neumann 1945). Anche in questo caso, come avvenne per von Neumann, il progetto non venne mai realizzato così come era stato concepito, ma il modello di calcolatore descritto nella proposta e raffinato nei due anni trascorsi da Turing all’NPL (ottobre 1945 – settembre 1947) rappresenta un archetipo di macchina diverso, anche se non alternativo a quello di von Neumann. Non sempre l’originalità del modello dell’ACE è stata riconosciuta in tutta la sua portata, anzi spesso il progetto è stato completamente ignorato, privilegiando l’analisi delle macchine realizzate negli Stati Uniti meglio organizzate, più finanziate e, soprattutto, più pubblicizzate!

Alcuni studiosi, tra i quali ad esempio Hodges (1983: 318), sostengono che la proposta per la costruzione dell’ACE fosse completamente indipendente da ogni altro testo sull’argomento, ma fu Turing stesso a dichiarare espressamente che essa andava letta insieme a quella di von Neumann (1945). Nonostante il legame esplicito tra le due

macchine, il progetto dell'ACE era a suo modo originale rispetto a quello del *First Draft*. Vediamo le differenze nei dettagli:

1) Lo scopo attribuito alla macchina: von Neumann la immaginava come uno strumento per eseguire calcoli complessi, Turing vedeva nel dispositivo uno strumento di manipolazione di simboli, che poteva ugualmente essere impiegato ad emulare una calcolatrice o una procedura per prendere decisioni nel gioco degli scacchi. Ognuna di queste funzioni così difformi comportava evidentemente l'elaborazione di metodi diversi.

2) L'accesso alla memoria: per Turing era necessario permettere l'accesso casuale ad ogni cella della memoria, per migliorarne l'efficienza; von Neumann non affermava nulla di esplicito al riguardo nel *First Draft*.

3) Il programma memorizzato: questo concetto per Turing comprendeva anche la possibilità di elaborare i programmi attraverso altri programmi, sfruttando il fatto che istruzioni e dati erano inseriti negli stessi registri di memoria e venivano rappresentati nello stesso linguaggio. Von Neumann fu all'inizio più cauto su questo punto: nel suo progetto le istruzioni si potevano modificare solo nella parte in cui era contenuto l'indirizzo.

4) Il modello di programmazione: nel progetto dell'EDVAC era possibile eseguire operazioni complesse direttamente al livello di unità di controllo. In quello dell'ACE invece le procedure erano descritte come una successione di istruzioni elementari e i programmi erano per Turing strutture modulabili – proprio come le tavole di istruzioni della sua macchina astratta – le quali si ripetevano ricorsivamente, tanto che per tenere

sotto controllo la chiamata dal programma centrale alle subroutine propone l'uso di un automa a pila.

2.2. Le macchine non organizzate

Uno dei temi centrali dell'intelligenza meccanica secondo Turing è rappresentato dalla discussione sulle macchine non organizzate. Questo argomento costituisce il centro della relazione del 1948, la sola nella quale Turing espliciti con convinzione la sua posizione a riguardo e esprima una vera e propria ipotesi su cosa costituisca l'intelligenza e come fare per riprodurla nei dispositivi meccanici. La capacità di auto-organizzarsi era considerata da Turing come una delle caratteristiche principali del pensiero. Tali macchine venivano introdotte senza approfondire il significato della mancanza di organizzazione, ma facendo esplicitamente riferimento alla casualità nella connessione delle componenti:

Potremmo invece considerare cosa succede quando montiamo una macchina in modo relativamente non sistematico, a partire da certi componenti standard. Potremmo considerare qualche macchina particolare di questa natura e indagare quali cose sarebbe in grado di fare. Macchine costruite in modo molto casuale, nel senso ora detto, saranno chiamate 'macchine non organizzate. Il termine non vuole essere accurato: è possibile che la stessa macchina sia considerata da un osservatore come organizzata e da un altro come non organizzata (Turing 1948/1994: 98 trad. parzialmente modificata).

Si trattava perciò di macchine composte di elementi standard collegati tra loro in modo casuale. Il termine "non organizzato" deriva probabilmente dalla tradizione della Cibernetica inglese, i cui studi anticiparono la nascita della disciplina datata ufficialmente con la pubblicazione del libro di Wiener nel 1948. In quegli anni, Turing intratteneva una corrispondenza con Ross Ashby, uno dei principali ideatori di dispositivi in grado di auto-organizzarsi ai fini del mantenimento dell'equilibrio (cfr. Ashby 1948, 1952-1960).

Già McCulloch e Pitts (1943) avevano introdotto il tema della costruzione, sia pure teorica, di una rete neurale, ma Turing non citò questo articolo, a nostro parere non tanto per ignoranza, ma piuttosto per scelta.¹ Egli, infatti, pensava a dispositivi pratici e non più solo astratti, tanto da aver compiuto degli esperimenti concreti su di essi. Il rafforzamento o l'indebolimento dei legami tra i singoli nodi interconnessi avrebbe permesso l'elaborazione di un percorso "di apprendimento", cioè una procedura di progressiva auto-organizzazione della topologia delle relazioni tra componenti. In termini moderni tali dispositivi potrebbero essere considerati come un sottoinsieme delle reti booleane casuali studiate successivamente da Stuart Kauffman (1993, 1995, 2000) (cfr. Teuscher 2002, Teuscher 2004a: 506).

Al di là delle caratteristiche tecniche di queste macchine² e degli errori che probabilmente Turing commise nel fornire la loro descrizione, assumendo che esse fossero universali (cfr. Teuscher 2004a: 516-517), il punto centrale, oseremmo dire filosofico, della questione risiede, a nostro avviso, nella nozione di interferenza.

I tipi di macchine che abbiamo considerato finora sono, in genere, macchine a cui è permesso di continuare la loro strada per periodi indefiniti senza interferenza dall'esterno. Le macchine universali rappresentano un'eccezione in quanto di tanto in tanto è possibile che si cambi la descrizione della macchina da simulare. Ora prenderemo in considerazione macchine in cui tale interferenza è la norma piuttosto che l'eccezione. (Turing 1948/1994: 101)

Turing sosteneva per la prima volta esplicitamente una posizione che pochi storici del pensiero scientifico sono stati disposti a riconoscergli: da questo momento in poi si interessò solo a macchine diverse da quelle tradizionali, che erano prive di contatti con

¹ Turing non poteva ignorare l'articolo se non altro per averne letto l'indicazione bibliografica nel *First Draft* di von Neumann e per averne utilizzato, sia pure modificandola, la notazione dei circuiti. Inoltre McCulloch e Pitts dichiararono di essersi direttamente ispirati a Turing 1937 nel loro lavoro (cfr. Copeland e Proudfoot 1996: 371), ed è probabile, quindi, che avessero cercato di contattarlo per fargli conoscere la loro conclusione.

² Per la descrizione tecnica delle macchine non organizzate di tipo A e B, cfr. Teuscher 2002, 2004a, Copeland e Proudfoot 1996.

l'esterno. I dispositivi ai quali voleva dedicare la sua attenzione erano quelli per cui l'interferenza non fosse l'eccezione, come nel caso della macchina universale, ma la regola. A questo fine distinse due tipi di interferenza: quella "del cacciavite" che consisteva nella sostituzione di alcune parti della macchina, e quella "di carta" che si limitava a passare alla macchina informazioni che ne modificavano il comportamento. Ma dichiarò che si sarebbe occupato solo dell'interferenza cartacea, dal momento che essa era in grado comunque di emulare l'altra.

Ma perché l'interferenza era così importante nella costruzione delle macchine non organizzate? Non dobbiamo dimenticare che questa relazione nasce con il chiaro intento di simulare attraverso la macchina l'intelligenza umana e Turing è convinto che:

nella misura in cui una persona è una macchina, lo è solo come macchina soggetta a moltissime interferenze (in effetti l'interferenza sarà la norma piuttosto che l'eccezione). Un uomo è in costante comunicazione con altri uomini e riceve continuamente stimoli visivi e di altro genere che di per sé costituiscono una forma di interferenza [...].

[...] è importante ricordare che, benché una persona quando si concentra può comportarsi come una macchina non soggetta a interferenza, il suo comportamento durante la concentrazione è determinato in larga misura dai modi in cui è stato condizionato dall'interferenza precedente. (Turing 1948/1994: 105)

In questo passo ci pare di ravvisare un'ipotesi sull'intelligenza umana. Essa è una capacità definita socialmente e frutto dei condizionamenti ambientali. Turing ammetteva la possibilità che una persona fosse paragonabile a una macchina, ma doveva trattarsi di un dispositivo molto particolare, in quanto il suo addestramento passava attraverso innumerevoli strati di interferenza frutto della continua interazione con gli altri individui, di stimoli di ogni genere. E anche nel momento della concentrazione, durante il quale egli si comportava come una macchina non soggetta ad interferenza, le sue reazioni e le sue scelte dipendevano in larga misura dai condizionamenti ricevuti. Già abbiamo visto l'importanza degli aspetti sociali per

mantenere una sorta di *fairness* nel paragonare i risultati ottenuti dagli uomini e dalle macchine invocata nella lezione del 1947 (Turing 1947). Dobbiamo notare purtroppo che nel più famoso articolo pubblicato su *Mind* (Turing 1950) egli non fu così esplicito a proposito della descrizione dell'intelligenza umana e del metodo da seguire per ottenerla nelle macchine. Le affermazioni al riguardo contenute nell'articolo del 1948 sono state sostanzialmente ignorate o male interpretate nella ricostruzione della posizione di Turing. D'altra parte nell'articolo vengono esplicitamente dichiarate alcune posizioni molto rilevanti: il paragone tra la macchina non organizzata, la corteccia cerebrale dei bambini e la macchina universale.

Noi crediamo che ci siano ampie zone del cervello, soprattutto nella corteccia, la cui funzione è in larga misura indeterminata. Nel neonato queste parti non svolgono un ruolo decisivo [...]. Nell'adulto hanno un effetto importante e finalizzato: il tipo di questo effetto dipende dall'addestramento dell'individuo nel corso della fanciullezza. Un notevole residuo del comportamento casuale dell'infanzia rimane nell'adulto. Tutto questo suggerisce che la corteccia del neonato sia una macchina non organizzata, che può essere organizzata con un opportuno addestramento tramite interferenza (Turing 1948/1994: 108).

Al momento della nascita, la corteccia cerebrale del neonato sarebbe simile a una macchina non organizzata destinata ad evolvere nel tempo attraverso le interferenze. Nell'età adulta resterebbe un residuo del comportamento casuale delle origini. Sebbene Turing proponga un modello troppo semplificato e inadeguato rispetto alla complessità della corteccia reale,³ in quanto dispositivo sincronizzato mentre la corteccia non lo è (Teuscher 2004a), vale la pena tener presente il notevole cambio di prospettiva sia rispetto alle macchine che all'intelligenza umana. Eppure non ci sembra neanche adeguato leggere questo testo come una diretta anticipazione del connessionismo (come fanno Copeland e Proudfoot 1996, Teuscher 2002, 2004a): esso è di meno e insieme di

³ Cfr. Zigmond *et al.* 1999 per una descrizione dello stato dell'arte della conoscenza nelle neuroscienze.

più. Turing, infatti, non riuscì a realizzare dei veri e propri esperimenti sulle macchine non organizzate per mancanza di mezzi adatti alla simulazione (ricordiamo che l'unica macchina in funzione all'epoca era la Baby Machine di Manchester che certo non permetteva prestazioni tali da testare una macchina non organizzata). Solo nell'anno della sua morte sarà realizzata al MIT (Farley e Clark 1954) la prima vera simulazione di una piccola rete neurale (cfr. Copeland e Proudfoot 1996: 368). D'altra parte non possiamo limitare la "filosofia" esposta da Turing sull'intelligenza meccanica ad una semplice anticipazione del connessionismo, perché sarebbe riduttivo. Ciò non toglie che in conseguenza delle convincenti argomentazioni di Copeland (Copeland, Proudfoot 1996) e Teuscher (2002, 2004a), risulta impossibile ascrivere *de jure* Turing alla lista degli studiosi che hanno sostenuto la posizione che l'intelligenza potesse essere prodotta attraverso una semplice messa a punto della struttura computazionale di manipolazione di simboli secondo regole precise e immutabili. Possiamo quindi sintetizzare la nostra posizione dicendo che Turing non possa essere considerato un sostenitore dell'Intelligenza artificiale forte, perché condivideva alcune delle idee ascrivibili all'approccio cibernetico, sebbene non avesse potuto sperimentare materialmente con le reti neurali, a causa delle limitazioni della macchina. D'altra parte egli non è nemmeno un 'anticipatore del connessionismo', sebbene condivida con questo movimento, tanto posteriore a lui, la tesi che la manipolazione simbolica secondo regole di per sé non sia sufficiente a produrre l'intelligenza meccanica. Era necessario un residuo. Le interpretazioni in senso connessionista dell'articolo del 1948, sebbene di per sé non pienamente condivisibili, hanno il merito di sottolineare e mettere in luce alcune delle meno note (eppure cruciali) direzioni di ricerca di Turing in materia di intelligenza.

Torniamo per un momento all'interferenza e al suo ruolo nel processo di addestramento della macchina. La sua importanza risiedeva nella capacità di modificare il comportamento della macchina, guidandone autonomamente l'apprendimento. Il processo di modifica prodotto dall'interferenza poteva essere provocato dalla macchina relativamente alla modifica delle proprie istruzioni, e in questo caso si attivava la capacità di automodificarsi. Sebbene tale termine potesse sembrare privo di senso, Turing invitava a utilizzarlo in quanto 'conveniente' rispetto all'obiettivo dell'apprendimento (cfr. Turing 1948/1994: 101). Nell'articolo del 1948 questa capacità veniva però spinta alle estreme conseguenze. Se una macchina era predisposta per interferire con la propria tavola di istruzioni, modificando il proprio comportamento in modo da reagire diversamente a uno stesso stimolo nel corso del tempo, si doveva ammettere che si era trasformata in un'altra macchina (Turing 1948/1994: 102). La tavola di istruzioni era infatti l'elemento che definiva l'identità stessa del dispositivo, che si trasformava nel tempo, pur essendo tale processo interamente sotto il controllo del programma. La macchina che consentiva tale processo implicava così la propria molteplicità, dal momento che dava vita a diverse trasformazioni che corrispondevano a diverse fasi dell'addestramento, impossibili da ottenere se non in presenza dell'interferenza. Il dispositivo intelligente dunque comprendeva una relazione stretta tra macchina (compresa della sua tavola di istruzioni) e fonti dell'interferenza (ambientali, magistrali, casuali ecc.).

2.3. Il gioco dell'imitazione: un test sociale per l'intelligenza

L'articolo del 1948 si concludeva con la presentazione di un piccolo esperimento relativo al gioco degli scacchi per dimostrare la precarietà di ogni giudizio che

attribuisca intelligenza a certi comportamenti. Secondo Turing, infatti, le azioni erano considerate intelligenti sulla base dello “stato mentale”, dell’addestramento o dell’esperienza che il valutatore aveva maturato rispetto al comportamento da giudicare. Poteva perciò capitare che l’ignoranza della regola che governava le azioni traesse in inganno l’esaminatore. Tale posizione lo spingeva a proporre un esperimento per controllare la difficoltà di esprimere un giudizio corretto sull’intelligenza meccanica. Il test si può considerare una versione ridotta, sebbene concettualmente identica, del “gioco dell’imitazione” (Turing 1950) di cui parleremo tra breve. Il gioco consisteva in una serie di partite a scacchi, ma presupponeva che i giocatori non fossero particolarmente esperti. Dei tre giocatori (A, B, C) B avrebbe manovrato una cosiddetta ‘macchina di carta’, ossia una procedura che stabiliva per ogni situazione sulla scacchiera una mossa da compiere. C doveva giocare due partite una con A e una con B, comunicando con l’avversario sempre attraverso una telescrivente. Il gioco avrebbe dovuto mostrare che C non sarebbe riuscito facilmente a capire se il suo avversario era un avversario umano o meccanico. Turing sostenne addirittura di aver già effettuato un simile esperimento, ottenendo risultati che confermavano la sua ipotesi (Turing 1948/1994: 120). Questa prima versione del gioco dell’imitazione ci offre una chiave di lettura per interpretare il significato del metodo più generale proposto nel più famoso articolo del 1950 pubblicato su *Mind* sulla quale ci soffermeremo dopo averlo brevemente introdotto. È impossibile ignorare i dettagli del gioco per chiunque si sia avvicinato anche solo brevemente ai temi dell’intelligenza artificiale, ma vale la pena

sintetizzarne le caratteristiche⁴ perché la sua popolarità ha comportato una serie di malintesi sul suo significato.⁵

In un dibattito radiofonico tenuto a gennaio del 1952 con Geoffrey Jefferson, professore di neurochirurgia all'università di Manchester e Newman, coordinato dall'epistemologo Richard Braithwaite, Turing condensava brevemente, ma efficacemente, il senso del suo test:

Vorrei suggerire un tipo particolare di *test* che si potrebbe applicare a 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 di Grado A. 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 finzione (*pretence*) è ragionevolmente convincente. Una parte considerevole della giuria, che non dovrà essere composta di esperti di macchine, dovrà essere ingannata dalla finzione (*pretence*) (Turing 1952a/1998: 28)

Questa descrizione chiarisce alcune caratteristiche del gioco che può essere considerato un esame che la macchina supera se riesce a raggirare una giuria di non esperti che pone domande generiche per un breve lasso di tempo. Inoltre si fa esplicitamente riferimento all'inganno che la macchina deve mettere in atto. Ciò impedisce di interpretare il test come una definizione operativa di intelligenza, come invece sostengono, tra gli altri, Hodges (1983: 415) e French (2000). L'intento di Turing era quello di proporre un metodo per riconoscere socialmente dei comportamenti intelligenti paragonabili a quelli umani e prodotti da dispositivi meccanici. Egli riteneva lecita anche la domanda «Non

⁴ Per ulteriori approfondimenti sull'argomento rimandiamo a un recente articolo che analizza il test e la sua ricezione negli ultimi cinquanta anni (Saygin *et al.* 2000). Il tema continua ad essere oggetto di interesse lo dimostra la mole di studi pubblicati sull'argomento. Ne citiamo tre che fanno il punto sulla produzione di questi anni: Shieber 2004, Moor 2003, Epstein, Roberts e Beber 2009.

⁵ I principali articoli contrari all'ipotesi che il test possa servire come strumento di valutazione dell'intelligenza delle macchine sono: Searle 1980, 1990 e recentemente 2009, Block 1981, 1990. Il presente paragrafo non si prefigge di confutare queste obiezioni che per lo più sono il frutto di una interpretazione del test che non è quella proposta da Turing, ma di descrivere correttamente il significato del test secondo Turing.

possono forse le macchine comportarsi in qualche maniera che dovrebbe essere descritta come pensiero ma che è molto differente da quanto fa un uomo?» (Turing 1950/1994: 123), ma il test aveva proprio lo scopo di evitare di essere costretto a risolvere problemi troppo complessi, con l'obiettivo di ottenere risposte comunque condivisibili. Il gioco garantiva una valutazione obiettiva dei comportamenti della macchina indipendentemente da come venivano prodotti (cfr. Copeland 2000: 522-524). Come nota Gualtiero Piccinini «Turing definiva l'intelligenza come un "concetto emotivo" intendendo che non esisteva un metodo obiettivo per applicarla» (2000: 579). Il gioco insomma evidenziava la tendenza dell'esaminatore umano a definire l'intelligenza attraverso un processo di riconoscimento sociale, lasciandosi, per così dire, "ingannare" da un certo tipo di prestazioni. Non si tratterebbe quindi di un metodo per stabilire le effettive capacità della macchina, ma solo per valutare le sue prestazioni in relazione a certi compiti considerati intelligenti. L'uso del test in sostituzione della domanda: "possono le macchine pensare?" serviva per eliminare l'intricatissimo nodo gordiano dell'intelligenza meccanica che Turing tentava di tagliare mentre rispondeva alle molte obiezioni filosofiche e del senso comune. L'articolo, infatti, proseguiva proprio con la discussione di alcune critiche che potrebbero efficacemente riassumere gran parte del dibattito sull'intelligenza artificiale che venne sollevato negli anni successivi e che è tuttora in corso.

Il lettore si sarà già accorto che non ho alcuna argomentazione molto convincente di carattere positivo per sostenere il mio punto di vista. Se l'avessi avuta non mi sarei certo dedicato con tanta cura a indicare gli errori dei punti di vista opposti al mio (Turing 1950/1994: 148-149).

Turing ammetteva quindi di non avere delle confutazioni conclusive da opporre ai detrattori del suo progetto, ma riteneva anche che la scienza non fosse solo una successione di certezze prive di congetture e solidamente posate sui fatti:

L'opinione popolare che gli scienziati procedano inesorabilmente da un fatto ben stabilito a un altro fatto ben stabilito, senza che intervenga mai l'influenza di una congettura non ancora provata è del tutto errata. Purché venga chiaramente messo in evidenza quali siano i fatti provati e quali siano le congetture, non può risultrarne alcun danno. Le congetture sono di importanza fondamentale, dato che suggeriscono utili linee di ricerca (Turing 1950/1994: 133).

Al momento egli aveva solo ipotesi e congetture da opporre alla lunga lista di obiezioni già espresse o potenziali sull'intelligenza meccanica e il test era la sua scommessa per ottenere consenso non appena i prototipi dei programmi intelligenti sarebbero entrati in funzione. Non si trattava di un metodo generale come quello per definire la calcolabilità effettiva che aveva inventato negli anni '30, perché non garantiva un paragonabile livello di universalità, affidabilità e credibilità, ma era pur sempre una procedura relativamente facile da realizzare che permetteva un controllo potenzialmente condivisibile sulla valutazione di alcune prestazioni meccaniche intelligenti.

Vale la pena notare che il test si proponeva di emulare l'abilità linguistica, cioè di valutare la capacità della macchina di sostenere una conversazione con uno sconosciuto non esperto di macchine per cinque minuti ingannandolo almeno nel trenta per cento dei casi (Turing 1950/1994: *ibidem*). L'attività della conversazione era per Turing l'applicazione di un sistema di regole socialmente condiviso. Come nella macchina di carta capace di giocare a scacchi, l'interazione linguistica con uno sconosciuto mediata attraverso una telescrivente non era altro che l'applicazione di regole socialmente riconoscibili per eseguire “mosse corrette”, anche se non sempre “vincenti”.

3. UNA SORTA DI CONCLUSIONE

Temo di essere molto lontano dallo stadio nel quale mi sento pronto a cominciare a pormi qualsiasi domanda di tipo anatomico [sul cervello]. Secondo le mie nozioni su come affrontare il problema, quel momento non arriverà che in uno stadio piuttosto avanzato, quando avrò una teoria adeguatamente definita su come stanno le cose.

Al momento non sto per niente lavorando al problema, ma alla mia teoria matematica dell'embriologia [...]

Lo sto facendo perché essa si presta più facilmente ad essere affrontata. Non penso tuttavia che sia del tutto sconnesso con l'altro problema. La struttura del cervello deve essersi realizzata attraverso un meccanismo di embriologia genetica, e io spero che questa teoria alla quale sto lavorando possa fare maggiore chiarezza su quali restrizioni ciò realmente implichi. Quello che mi dici a proposito della crescita dei neuroni sotto stimolazione è molto interessante alla luce di tale connessione. Suggerisce metodi attraverso i quali i neuroni possono esser fatti crescere per formare un particolare circuito, piuttosto che per raggiungere un particolare posto. (Lettera di Turing a J. Z. Young, datata: 8/2/1951).⁶

Dalla fine degli anni '40 e fino alla sua morte, avvenuta per suicidio nel 1954, Turing si interessò di problemi di morfogenesi ed embriologia usando il calcolatore di Manchester per risolvere problemi relativi al trattamento matematico dello sviluppo della vita.⁷ Pur non potendo approfondire il tema dei suoi ultimi studi, come meriterebbero a causa di limiti di spazio non possiamo ignorare l'importanza della lettera scritta a Young che ci offre una chiave di lettura che consente di tenere insieme i diversi interessi che hanno caratterizzato negli anni il lavoro di Turing. In tutto il suo lavoro riteniamo che Turing abbia cercato di realizzare un unico obiettivo che consisteva nella comprensione del funzionamento del cervello e dei meccanismi dell'intelligenza, ai fini di una loro simulazione o riproduzione attraverso adeguati dispositivi meccanici. Il suo atteggiamento nei confronti degli esperimenti di simulazione dell'intelligenza e di quelli per la comprensione del funzionamento dell'evoluzione della vita dal punto di vista embriologico sembrano essere esattamente

⁶ Una copia della lettera si trova nei Modern Archives del King's College di Cambridge (collocazione K. 78). La lettera è pubblicata in Copeland 2004, p. 517.

⁷ Per approfondimenti su questo tema cfr. Swinton 2003e 2004. Da notare che gran parte dei risultati ottenuti da Turing in questo campo furono pubblicati solo postumi, cfr. Turing 1952*b*, Turing 1992*a*.

sullo stesso piano e soprattutto sono guidati da un unico principio. La strategia di ricerca di Turing potrebbe essere perfettamente esemplificata dalla descrizione del metodo della modellizzazione:

La morale è che non c'è nessun valido criterio di decisione a priori per stabilire quali vincoli debbano essere presi in considerazione. Piuttosto, essi sono suggeriti dallo stato della conoscenza sperimentale, da ipotesi opposte psicologiche o neurologiche, e dalle prestazioni dei modelli reali (Cordeschi 1998/2002: 255).

Turing, cioè, non aveva pregiudizi a proposito dei diversi modelli di intelligenza da riprodurre meccanicamente, e cercava di esplorare tutte le possibilità che a suo avviso avrebbero avuto risultati interessanti per il suo obiettivo finale: realizzare macchine intelligenti.

Pur non potendo soffermarci adeguatamente sui suoi ultimi studi vogliamo brevemente ricapitolare i suoi risultati nell'ambito della biologia matematica. Nell'articolo del 1952 *The chemical basis of Morphogenesis* Turing fornì un'ipotesi su come si riproducevano le forme in una grande varietà di casi biologici come nel caso dei petali dei fiori, delle macchie sulla pelle, della formazione delle foglie sui giovani ramoscelli ecc. L'idea era che esistessero degli agenti chimici detti morfogeni che se presenti in una certa densità creavano le forme attraverso meccanismi di reazione e diffusione (Swinton 2004, p. 481). Per spiegare questo meccanismo Turing adotta un principio noto come *Turing instability* che dovrebbe naturalmente generare le macchine sulla pelle degli animali. Questo modello di spiegazione nasceva dall'interazione tra la dinamica reazione-diffusione e la geometria dello spazio al quale si applicava. Tale principio offrì un paradigma condiviso per la spiegazione del fenomeno della morfogenesi dal punto di vista matematico (Swinton 2004: 484). Tuttavia Turing proseguì i suoi studi anche dopo la pubblicazione dell'articolo del 1952 alla ricerca di una spiegazione casuale del perché

tali meccanismi funzionavano sulla base di principi di natura matematica come l'applicazione della serie di fibonacci alla descrizione di alcune forme in natura come per esempio quella della margherita. La sua ipotesi era che vi fosse un fenomeno di Fillotassi Geometrica che avrebbe potuto spiegare certi comportamenti nella costituzione delle forme naturali, ma non c'era nessuna prova che le cose stessero così. Swinton, uno dei più acuti interpreti di questa ultima parte delle ricerche di Turing, suggerisce che nella parte finale della sua vita egli fosse arrivato a spiegare le regole per l'apparizione del modello fibonacci di organizzazione delle forme, sebbene non fosse arrivato a comprendere le ragioni del funzionamento delle regole. Per ottenere questa spiegazione ci vollero altri cinquanta anni di studi, spesso all'oscuro del lavoro di Turing sull'argomento (Swinton 2004: 496).

In questo profilo abbiamo cercato di dare conto dei diversi contributi forniti da Turing a posizioni anche contrapposte in questo campo. Inoltre abbiamo illustrato la sua visione senza arretrare di fronte alle contraddizioni alle quali in alcuni casi egli è andato incontro. Abbiamo mostrato la progressiva presa di distanza dalla concezione della logica che prevedeva che essa fosse l'unico strumento legittimo per la soluzione di problemi. Nell'ultimo articolo da lui pubblicato nell'anno della sua morte su alcuni problemi logici insolubili egli concludeva con un'affermazione che possiamo considerare il suo testamento spirituale:

I risultati che sono stati descritti in questo articolo sono per la maggior parte di carattere negativo, stabilendo certi vincoli a ciò che possiamo sperare di ottenere esclusivamente attraverso il ragionamento. Questi e altri risultati di logica matematica possono essere considerati come una strada verso una dimostrazione, entro la matematica stessa, dell'inadeguatezza della 'ragione' senza il supporto del senso comune (Turing 1954/1992: 203).

Turing vedeva quindi perfettamente i limiti di un approccio puramente logico o algoritmico alla matematica e, più in generale, alla soluzione di problemi. Era infatti consapevole che la sua macchina astratta implicava, nella sua versione deterministica, quelle caratteristiche della logica che non le permettevano di andare oltre confini prestabiliti. Secondo Copeland (2004: 579-580) l'argomento qui sollevato può essere paragonato a quello in favore dell'intuizione sostenuto da lui già nel 1938, ma ci pare che qui egli vada anche oltre, e proponga l'uso del senso comune come strumento di soluzione di quelle questioni nelle quali la ragione rimane paralizzata a causa della propria inadeguatezza. La parabola intellettuale di Turing può essere in fondo sintetizzata dal passaggio dagli studi di teoria della computabilità alle ricerche sulla simulazione di meccanismi adattivi e a retroazione propri della cibernetica. L'obiettivo finale del suo lavoro consisteva sostanzialmente nella realizzazione di macchine capaci di apprendimento, evoluzione, crescita, auto-organizzazione e interazione con l'ambiente. Tale nuova prospettiva si spiega con un interesse sempre più spiccato per la simulazione meccanica della capacità creativa e innovativa umana. Dal primo impegno nell'ambito della teoria della computabilità agli ultimi lavori sulla crescita delle forme vitali, c'è una linea che, passando per varie fasi, potrebbe essere unificata dal proposito di indagare le possibilità, i metodi e i limiti della conoscenza umana per tentare di riprodurli meccanicamente. Se affrontiamo la sua parabola intellettuale da questa prospettiva, c'è un unico filo che lega il problema della decisione, in quanto contributo alla discussione sulla legittimità dell'uso di certi metodi di dimostrazione matematica, alla simulazione di comportamenti intelligenti ottenuti con la possibilità di commettere errori da parte del dispositivo, fino all'organizzazione delle forme della vita. Sono tutti

passi verso la scoperta dei meccanismi che condizionano e regolano la conoscenza intesa come capacità di adattamento e di apprendimento rispetto all'ambiente esterno e agli altri agenti. Il percorso di ricerca di Turing posto in questa luce si può descrivere attraverso il continuo, sebbene non univoco, processo di spostamento dalla meccanizzazione del calcolo, alla macchina intelligente, fino alla matematizzazione dello sviluppo della vita, al centro degli interessi di genetica matematica ai quali si era dedicato negli ultimi anni.

4. PICCOLA GUIDA BIBLIOGRAFICA SU TURING

Orientarsi nella bibliografia su Turing può essere un'impresa complicata. Essendo un autore tanto importante e tanto poliedrico ci sono molte diverse prospettive dalle quali guardare la sua opera. Per orientarsi in questa difficile mole di lavori proponiamo alcuni testi dai quali è più utile partire per addentrarsi nella sua opera.

Innanzitutto le opere complete: ci sono quattro volumi di opere complete (Turing 1992, 1992a, 1992b, 2001) nei quali si può trovare il grosso dei lavori pubblicati e inediti. I quattro volumi coprono le quattro aree principali delle sue ricerche: intelligenza meccanica, matematica, morfogenesi, logica matematica. Prima di questi volumi sulla sua opera era disponibile un'interessante antologia dei suoi scritti curata da Carpenter e Doran (1986), che includeva anche il progetto per la realizzazione della macchina ACE. Più recentemente è stato pubblicato un'antologia di sue opere curata da Jack Copeland (2004) che fa una selezione abbastanza interessante dei suoi lavori e include anche alcune lettere e testi che non sono raccolti nelle opere complete. In Italiano la disponibilità dei testi di Turing è piuttosto scarsa. I testi sull'intelligenza meccanica sono disponibili in Turing 1994, una raccolta di scritti a cura di Gabriele Lolli. Inoltre alcuni testi inediti di Turing su questo tema, soprattutto interviste e lezioni radiofoniche sono stati pubblicati nel vol. 10 della rivista *Sistemi Intelligenti* (1998).

Per quanto riguarda la biografia intellettuale, il lavoro principale è quello di Andrew Hodges 1983 (ripubblicato per l'anniversario). Hodges ha inoltre scritto un altro testo breve di introduzione al suo pensiero (1997). Un altro testo biografico importante è quello della madre Sara Turing (1959) anch'esso ripubblicato per l'anniversario. Subito dopo la sua morte nel 1955 Newman, il suo maestro, ha scritto un articolo coccodrillo

nel quale si proponeva una biografia intellettuale del suo amato allievo e collaboratore. Nell'ambito delle celebrazioni del centenario di Turing avvenute nel 2012 molti sono stati i testi usciti per ricordare l'attività e il contributo di Turing, citiamo tra tutti il testo a lui dedicato da Copeland (2012a). Molti sono anche i testi di fiction ispirati a Turing, ma esiste anche una biografia scritta da uno scrittore di professione che risulta molto piacevole sebbene un po' romanzata (Leavitt 2006).

Per quanto riguarda la teoria della computabilità, i contributi di Turing costituiscono parte integrante della disciplina, e come tali vengono trattati in numerosi manuali. Tra essi ricordiamo Boolos *et al.* 2007 (per chi avesse interessi rivolti soprattutto alla logica), Davis *et al.* 1994 e Sipser 2012 (più orientati all'informatica). Petzold 2008 è un volume che offre una spiegazione e un commento capovero per capovero del classico articolo in cui per la prima volta vengono descritte le macchine di Turing e dimostrate sia l'esistenza della macchina universale, sia l'indecidibilità dell'*Entscheidungsproblem* (Turing 1936-37). In italiano ricordiamo Frixione e Palladino 2011 (più sintetico e introduttivo) e Frixione e Palladino 2004 (più approfondito).

In Copeland 2004 è possibile trovare anche un'ampia rassegna di testi e discussioni relative al lavoro di Turing durante la seconda guerra mondiale, a cui sono dedicati i capp. 4-8. Sull'attività di decodifica dell'Enigma, anche se non specificamente su Turing si possono leggere vari testi di cui segnaliamo Smith e Erskine (2001) e Sebag-Montefiore (2000).

Per quanto riguarda il lavoro sul calcolatore molti sono i testi che ne discutono. Possiamo citare Davis 2000, Numerico 2005, Lavington 2012. Un altro testo importante

sull'argomento è un articolo che mette a confronto il progetto di Turing e quello di von Neumann (Carpenter e Doran 1977). Sui rapporti tra Turing e von Neumann si può citare anche il recente (Dyson 2012), testo ben documentato da solide ricerche d'archivio sulle origini del calcolatore e della digitalizzazione. Inoltre la raccolta di testi a cura di Copeland (2012) offre un'ampia panoramica di questioni intorno alla costruzione del calcolatore da parte di Turing.

Sull'intelligenza meccanica e il test di Turing la bibliografia è molto vasta e complessa, il punto di partenza resta senza dubbio Copeland 2004 che si occupa di questo argomento nei capp. 9-14 e 16-17. Un'antologia di autori che si sono occupati di intelligenza artificiale tra i quali anche Turing è (Boden 1990). Per quanto riguarda il test di Turing sarebbe impossibile dare conto della letteratura che se ne occupa citiamo tre antologie che possono facilitare il compito di introdurre alle questioni principali in discussione Shieber (2004), Moor (2003), Epstein *et al.* (2009).

Non sono molti i lavori che affrontano il tema della morfogenesi, l'ultimo interesse di Turing. Ne citiamo due che possono costituire un valido punto di partenza sull'argomento: Swinton 2003, 2004.

Molte risorse su Turing sono reperibili in rete. Citiamo quelle secondo noi più significative e importanti.

La pagina mantenuta da Andrew Hodges <http://www.turing.org.uk/turing/> .

L'archivio dei documenti di Turing <http://www.turingarchive.org/> mantenuto dal King's college.

La pagina a cura di Jack Copeland dove è possibile consultare la maggior parte del materiale reperibile su Turing <http://www.alanturing.net/> .

Inoltre Jack Copeland e Diane Proudfoot hanno pubblicato nel *Rutherford Journal* del 2012 un'ampia biografia intellettuale disponibile online <http://rutherfordjournal.org/article040101.html>.

Una pagina molto interessante è quella che si occupa dei festeggiamenti dell'anno del centenario 2012, dove è possibile avere una ricognizione di tutte le istituzioni che hanno voluto rendere omaggio al grande scienziato. La pagina è disponibile alla URL: <http://www.mathcomp.leeds.ac.uk/turing2012/>

Per concludere la rassegna di testi online su Turing citiamo la voce a lui dedicata della *Stanford Encyclopedia of Philosophy*, scritta da Andrew Hodges, disponibile alla Url: <http://plato.stanford.edu/entries/turing/>

5. BIBLIOGRAFIA

Ashby W. R. (1948), "Design for a brain", *Electronic Engineering*, 20:379-383.

Ashby W. R. (1952- 1960), *Design for a brain*, Chapman and Hall, London, II ed. Wiley New York 1960. Trad. it. della II ed. *Progetto per un cervello*, Bompiani, Milano, 1970].

Block N. (1981), "Psychologism and behaviorism", *Philosophical Review*, 90: 5-43.

Block N. (1990), "The computer model of the mind", in Osherson, D. *et al.* (a cura di), *Thinking. An Invitation to cognitive science*, vol. 3, MIT Press, Cambridge, Mass: 247-289.

Boden M. A. (1990), *The philosophy of artificial intelligence*, Oxford University Press, Oxford.

Boolos G.S., Burgess J.P. e Jeffrey R.C. (2007), *Computability and Logic* (5a edizione), Cambridge University Press, Cambridge.

- Carpenter B. E. e Doran R. W. (1977), “The other Turing machine”, *Computer Journal*, 20:269-279.
- Carpenter B. E. e Doran R. W. (1986) (a cura di), *A.M. Turing's Ace report of 1946 and other papers*, MIT Press, Cambridge Mass.
- Church A. (1936), “An unsolvable problem of elementary number theory”, *American Journal of Mathematics* 58, pp. 345-363.
- Church A. (1936a), “A note on the Entscheidungsproblem”, *The Journal of Symbolic Logic* 1, pp. 40-41 e “A correction”, *ibid.*, pp. 101-102.
- Copeland, J. B. e Proudfoot, D. (1996), “On Alan Turing’s anticipation of connectionism”, *Synthese*, 108: 361-377.
- Copeland, J. B. (2000), “The Turing Test”, *Minds and Machines*, 10: 519-539.
- Copeland, J. B. (2004) (a cura di), *The essential Turing*, Clarendon Press, Oxford.
- Copeland, J.B. (2012) (Ed.), *Alan Turing's Electronic Brain: The Struggle to Build the ACE, the World's Fastest Computer*, Oxford University Press, Oxford.
- Copeland J.B. (2012a), *Turing. Pioneer of the information age*, Oxford Univ. Press, Oxford.
- Cordeschi, R. (1998/2002), *La scoperta dell’artificiale*, Dunod, Salerno; [trad. ingl. *The discovery of the artificial*, Kluwer Academic Press, Dordrecht, 2002].
- Dyson, G. (2012), *Turing's cathedral. The origins of the digital universe*, Allen Lane, Penguin Books, London.
- Davis M. (2000), *The Universal Computer: The Road from Leibniz to Turing*, W.W. Norton & Company, New York [trad. It. *Il calcolatore universale: Da Leibniz a Turing*, Adelphi, Milano, 2012]
- Davis M., Sigal R. e Weyuker E. (1994), *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science* (2a edizione), Academic Press,

New York.

Epstein, R., Roberts, G., Beber, G. (2009) (Eds.), *Parsing the Turing Test*, Springer Science.

Farley, B. G. e Clark, W. A. (1954), "Simulation of self-organizing systems by digital computer", *Institute of Radio Engineers transactions on Information Theory*, 4: 76-84.

French, R. M. (2000), "The Turing test: The first 50 years." *Trends in Cognitive Sciences*, 4(3), 115-122.

Frixione M., Palladino D. (2004), *Funzioni, macchine, algoritmi. Introduzione alla teoria della computabilità*, Carocci, Roma.

Frixione M., Palladino D. (2011), *La computabilità: algoritmi, logica, calcolatori*, Carocci, Roma.

Hilbert D. e Ackermann W. (1928), *Grundzüge der Theoretischen Logik*, Springer, Berlin.

Hodges A. (1997), *Turing*, Phoenix Paperback, London.

Hodges, A. (1983), *Alan Turing The enigma of intelligence*, II Ed. Vintage, London, 1992. [trad. it. *Storia di un enigma*, Bollati Boringhieri, Torino, 2003].

Husbands, P. and Holland O. (2008), "The Ratio Club. A hub of British Cybernetics", in O. Husbands, P., Holland, O., Wheeler, M., (eds.) *The Mechanical Mind in History*, MIT Press, Princeton: 91-148.

Kauffman, S. (1993), *The origins of order: Self-organization and selection in evolution*, Oxford University Press, New York, Oxford.

Kauffman, S. (1995), *At home in the universe*, Oxford University Press, New York, Oxford.

Kauffman, S. (2000), *Investigations*, Oxford University Press, New York, Oxford.

- Lavington, S. (2012) (Ed.), *Alan Turing and his contemporaries*, British Informatics society, Swindon.
- Leavitt, D. (2006), *The man who knew too much*, W.W. Norton & Company, New York, trad. it. *L'uomo che sapeva troppo*, Codice, Milano, 2012.
- McCulloch, W. S. e Pitts, W. (1943), "A logical calculus of the ideas immanent in nervous activity", *Bulletin of the Mathematical biophysics*, 5: 115-133. Ristampato in Boden 1990: 22-39.
- Moor, J.H., (2003) (Ed.), *The Turing test. The elusive standard of artificial intelligence*, Kluwer Academic Publishers, Dordrecht.
- Neumann (von) J. (1945), *First draft of a report on the Edvac*, (30 giugno), Contract, n. W-670-ORD-4926 Moore School of Electrical Engineering, University of Pennsylvania; parzialmente stampato in Randell B. (1982) (a cura di), *The origins of digital computers*, Springer-Verlag, Berlin, 1982: 383-392, ristampa con correzioni in *Annals of the History of Computing*, 15: 25-75, (1993).
- Newman M. H. A. (1955), "A.M. Turing", *Biographical Memoirs of fellows of the Royal Society*, 1: 252-263.
- Numerico T. (2005), *Alan Turing e l'intelligenza meccanica*, Franco Angeli, Milano.
- Petzold C. (2008), *The Annotated Turing. A Guided Tour through Alan Turing's Historic Paper on Computability and the Turing Machine*, Wiley, New York.
- Piccinini, G. (2000), "Turing's Rules for the Imitation game", *Minds and Machines*, 10: 573-582.
- Saygin, A. P., Cicekli, I. e Akman, V. (2000), "Turing Test: 50 years later", *Minds and Machines*, 10: 463-518.
- Searle, J. (1980), "Minds, brains and programs" *Behavioural and brain sciences*, 3: 417-424, Trad. It *Menti, macchine e programmi*, in Hofstadter D.R. e Dennett D.C., *L'io della mente*, Adelphi, Milano, 1985, pp.341-360.

- Searle, J. (1990), “Is the Brain's Mind a Computer Program?”, *Scientific American* 262 (1): 26–31.
- Searle, J. (2009), “The Turing test: 55 years later”, in Epstein *et al.*: 139-150.
- Sebag-Montefiore H. (2000), *Enigma: the battle for the code*, Weidenfeld & Nicolson, London.
- Shieber, S. (2004) (Ed.), *The Turing test*, A Bradford Book, The MIT Press, Cambridge, (Mass.).
- Sipser M. (2012), *Introduction to the Theory of Computation*, 3a edizione, Cengage Learning, Andover, UK.
- Smith M. e Erskine R. (2001) (a cura di), *Action this day*, Bantam Press, London.
- Swinton, J., (2003) Website: Turing and morphogenesis
<http://www.swintons.net/jonathan/turing.htm>.
- Swinton J. (2004), *Turing and Fibonacci Phyllotaxis*, in Teuscher (2004), pp. 477-498.
- Teuscher, C. (2002), *Turing's connectionism: an investigation of neural network architectures*, Springer-Verlag, London.
- Teuscher, C. (2004) (a cura di), *Alan Turing: life and legacy of a great thinker*, Springer, Berlin.
- Teuscher, C. (2004a), “Turing’s Connectionism”, in Teuscher 2004: 499-529.
- Turing A. (1936-37), “On computable numbers, with an application to the Entscheidungsproblem”, *Proceedings of the London Mathematical Society* 42, pp. 230-365; “A correction”, *ibid.*, 43 (1937), pp. 544-546.
- Turing, A. M., (1939), “Systems of logic based on ordinals”, *Proc. London Mathematical Society*, (2), 45: 161-228; ristampato in Copeland 2004: 146-204.
- Turing, A. M. (1945), *Proposal for the development in the Mathematical Division of an Automatic computing engine (Ace)*, rapporto all’Executive Committee del

- National Physical Laboratory del 1945. Pubblicato in Carpenter e Doran 1986: 20-105; e in Turing 1992: 1-86; [parziale trad. it. in Turing 1994: 29-62].
- Turing, A. M. (1947), “Lecture to the London Mathematical Society on 20 February 1947”, Turing 1992: 87-105, ristampato in Copeland 2004: 378-394; [trad. it. in Turing 1994: 63-87].
- Turing, A. M. (1948), “Intelligent Machinery” Report, National Physics Laboratory, in B. Meltzer D. Michie (a cura di), *Machine intelligence*, 5 (Edinburgh Univ. Press, 1969: 3-23; ristampato in Turing 1994: 107-127 e in Copeland 2004: 410-432; [trad. it. in Turing 1994: 88-120].
- Turing, A. M. (1950), “Computing Machinery and Intelligence”, *MIND*, 59: 433-460. ristampato in Turing 1992: 133-160 e in Copeland 2004: 441-464, [trad. it. in Turing 1994: 121-157].
- Turing, A. M. (1952a), *Can automatic calculating machines be said to think?* Conferenza alla BBC, Turing’s paper, Modern Archives King’s College Cambridge; pubblicato in Copeland 2004: 494-506 [trad. it. in *Sistemi intelligenti*, 10: 27-40 (1998)].
- Turing, A. M. (1952b), “The chemical basis of morphogenesis”, *Philosophical Transactions of the Royal Society of London*, B 237: 37-72, ristampato in Copeland 2004: 519-561.
- Turing, A. M. (1954), “Solvable and unsolvable problems”, *Science news*, 31: 7-23; ristampato in Turing 1992: 187-216 e in Copeland 2004: 582-595.
- Turing, A. M. (1992), *Collected Works of A. M. Turing, Vol.1 : Mechanical intelligence*, Ince D. C. (a cura di), North-Holland, Amsterdam.
- Turing, A. M. (1992a), *Collected Works of A.M. Turing, Vol.3: Morphogenesis*, Saunders P.T. (a cura di), North-Holland, Amsterdam.
- Turing A. M. (1992b), *Collected Works of A.M. Turing, Vol.2: Pure Mathematics*, Britton, J.L. (a cura di), North-Holland, Amsterdam.

Turing, A. M. (1994), *Intelligenza Meccanica*, Lolli, G. (a cura di), Bollati Boringhieri, Torino.

Turing, A.M. (2001) *Collected Works of A.M. Turing, Vol. IV: Mathematical Logic*, Gandy, R. e Yates, C.E.M., (Eds.), North-Holland, Amsterdam.

Turing, S. (1959-2012) *Alan M. Turing*, Cambridge University Press, Cambridge (UK), Centenary Edition.

Wiener, N. (1948), *Cybernetics: or control and communication in the animal and the machine*, Technology Press, Cambridge Mass [trad. ital. *La Cibernetica: controllo e comunicazione nell'animale e nella macchina*, Il Saggiatore, Milano, 1968].

Zigmond, M.J., Bloom, F.E., Landis, S.C., Roberts, J. L. e Squire, L.R. (1999), (a cura di), *Fundamental Neuroscience*, Academic Press, San Diego.

APhEx.it è un periodico elettronico, registrazione n° ISSN 2036-9972. Il copyright degli articoli è libero. Chiunque può riprodurli. Unica condizione: mettere in evidenza che il testo riprodotto è tratto da www.aphex.it

Condizioni per riprodurre i materiali --> Tutti i materiali, i dati e le informazioni pubblicati all'interno di questo sito web sono "no copyright", nel senso che possono essere riprodotti, modificati, distribuiti, trasmessi, ripubblicati o in altro modo utilizzati, in tutto o in parte, senza il preventivo consenso di APhEx.it, a condizione che tali utilizzazioni avvengano per finalità di uso personale, studio, ricerca o comunque non commerciali e che sia citata la fonte attraverso la seguente dicitura, impressa in caratteri ben visibili: "www.aphex.it". Ove i materiali, dati o informazioni siano utilizzati in forma digitale, la citazione della fonte dovrà essere effettuata in modo da consentire un collegamento ipertestuale (link) alla home page www.aphex.it o alla pagina dalla quale i materiali, dati o informazioni sono tratti. In ogni caso, dell'avvenuta riproduzione, in forma analogica o digitale, dei materiali tratti da www.aphex.it dovrà essere data tempestiva comunicazione al seguente indirizzo (redazione@aphex.it), allegando, laddove possibile, copia elettronica dell'articolo in cui i materiali sono stati riprodotti.

In caso di citazione su materiale cartaceo è possibile citare il materiale pubblicato su APhEx.it come una rivista cartacea, indicando il numero in cui è stato pubblicato l'articolo e l'anno di pubblicazione riportato anche nell'intestazione del pdf. Esempio: Autore, *Titolo*, <<www.aphex.it>>, 1 (2010).
