

## Aspetti filosofico e tecnico della *Turing Machine*

di Davide Dell'Ombra

Niente mi sembra più adatto che cominciare con una profezia: «sono dell'idea che alla fine del nostro secolo, l'uso delle parole e il pensiero delle persone colte si saranno talmente modificati da permettere di parlare di macchine pensanti senza incontrare difficoltà di comunicazione»<sup>1</sup>. Alan Turing ha certamente lasciato dopo di sé la più sviluppata scienza odierna, quella informatica e tecnologicamente legata ai calcolatori, ma senza dubbio ha anche lasciato uno dei quesiti fondamentali, certamente non nuovo ma affrontato in maniera nuova: cos'è l'intelligenza?

L'apporto di Turing all'impresa dell'intelligenza artificiale va considerato come fondamentale. Come Pratt chiaramente illustra, è seguendo le linee di sviluppo della psicologia degli inizi del Novecento, intessuta dello sperimentalismo di Watson che in *Behavior*<sup>2</sup> «raccomandava alla psicologia di restringere la sua attenzione esclusivamente al fenomeno del comportamento»<sup>3</sup> cui fa seguito il comportamentismo di Skinner del 1938, che nel 1950 Turing «propose di interpretare la domanda “Le macchine possono pensare?” senza far riferimento ad alcun loro processo “mentale”»<sup>4</sup>. E' ancora Pratt a chiarire il ruolo svolto da Turing in questo ambito: «Il suo lavoro in logica lo aveva condotto a sviluppare una chiara descrizione di cosa fosse una procedura meccanica, e lo aveva condotto alla formulazione del corollario per il quale tutte le macchine a stati finiti (perché in grado di compiere un repertorio fondamentale di azioni semplici) erano logicamente equivalenti. Data la concezione predominante dell'organismo complesso con una gamma di comportamenti appropriati, fu possibile concepire il cervello come una macchina a stati finiti con funzione di controllo del comportamento. Non parve quindi assurdo credere nella possibilità di costruire un giorno un equivalente meccanico o elettronico»<sup>5</sup>. La macchina potrebbe essere dunque programmata allo scopo di «imitare il funzionamento cerebrale»<sup>6</sup>:

---

<sup>1</sup> Williams, p. 33, riprendendo lo stesso Turing.

<sup>2</sup> Pratt si riferisce a: Watson, *Behavior: An introduction to Comparative Psychology*, New York, Holt, Rinehart, Winston, 1914 (Cfr. Pratt, p. 232).

<sup>3</sup> Cfr. Pratt, p. 222.

<sup>4</sup> Pratt, p. 222.

<sup>5</sup> Pratt, p. 232.

<sup>6</sup> Pratt, p. 235. Pratt aggiunge inoltre riguardo Turing, a p. 302, che «benché il cervello possa apparire estremamente sofisticato, nel momento in cui lo si consideri una macchina a stati finiti, si può ritenere fattibile il progetto di costruirne un equivalente elettronico».

si deve infatti a Turing la concezione di un calcolatore «idealizzato»<sup>7</sup>, la nota *Turing Machine* che basandosi sull'elaborazione di un dato in ingresso considerato il suo stato interno, scandendo ogni elaborazione in stati finiti, sarebbe in grado, se dotata di memoria e tempi di calcolo indefiniti, di computare qualsiasi «funzione calcolabile mediante un procedimento algoritmico», secondo la tesi Church-Turing.

Infatti «è possibile dimostrare che le funzioni calcolabili da una MT [Macchina di Turing] sono equivalenti a quelle comprese in una importante ed estesa classe di funzioni matematiche, dette funzioni ricorsive, le quali, a loro volta, secondo la cosiddetta “tesi di Church” sono equivalenti alle funzioni intuitivamente computabili. Sebbene la tesi di Church sia una congettura non dimostrabile formalmente (dato che cerca di rendere conto dell'idea intuitiva di “calcolabile in modo meccanico”), il credito di cui gode autorizza a concludere che una MT è in grado di calcolare ogni funzione computabile»<sup>8</sup>. Anche Frixione nota come le espressioni «algoritmo» e «funzione computabile in modo algoritmico» in realtà siano «concetti intuitivi, non specificati in modo formale, per cui una dimostrazione rigorosa di equivalenza con il concetto di funzione calcolabile da una MT non è possibile. La Tesi di Church quindi non è un teorema, e neppure una congettura che potrebbe un giorno, in linea di principio, diventarlo. Si tratta di un enunciato che si fonda su un ampio spettro di evidenza matematica di tipo euristico»<sup>9</sup>. Continua Frixione affermando che alcuni studiosi considerano la Tesi di Church «come una sorta di “legge empirica” piuttosto che come un enunciato a carattere logico-formale. Il logico Emil Post, il quale propose un concetto di macchina calcolatrice in parte analogo a quello sviluppato da Turing [...], sottolineava il suo disaccordo con chi tendeva ad identificare la Tesi di Church con un assioma o una mera definizione [giacché andava considerata] una “ipotesi di lavoro”, che, se opportunamente “corroborata”, dovrebbe assumere il ruolo di una “legge naturale”, una “fondamentale scoperta circa le limitazioni del potere matematizzante dell'*Homo sapiens*”»<sup>10</sup>.

Le macchine che riescano a simulare ogni tipo di macchina di Turing sono dette *universali*. Ma non tutto può essere computato, come ricorda Johnson-Laird, «poiché esistono molti problemi che possono essere enunciati ma non hanno alcuna soluzione computabile. E' impossibile ad esempio progettare una macchina universale in grado di determinare se una qualsiasi macchina di Turing selezionata arbitrariamente, sulla base di

---

<sup>7</sup> Interessante notare, come fa *Williams* (p. 29), che Turing in inglese usa il termine «computer» essenzialmente come in italiano diremmo «ragioniere».

<sup>8</sup> *Di Francesco*, p. 140.

<sup>9</sup> *Frixione*, p. 58.

<sup>10</sup> *Frixione*, p. 58.

alcuni dati selezionati arbitrariamente, si fermerà o continuerà a computare per l'eternità. Non c'è alcun test che permetta di decidere in maniera sicura se un problema ha o no una soluzione computabile»<sup>11</sup>. L'interpretazione comportamentista di Pratt, che abbiamo riportato prima, del *gioco dell'imitazione* di Turing, cioè la proposta che le macchine pensino anche solo se riescono a farlo credere a chi facesse loro delle domande e ricevesse risposte per via elettronica, è senza dubbio condivisibile e infatti, come aggiunge Frixione, «ci troviamo di fronte ad una sorta di versione “estremista”, o “radicale”, della Tesi di Church, che, grosso modo, potrebbe essere formulata come segue: *ogni attività linguistico-cognitiva è calcolabile da una macchina di Turing*, (il che non vuol dire, ovviamente, che la nostra mente funziona *come* una macchina di Turing, ma che ogni attività mentale è simulabile da un dispositivo che abbia la stessa potenza computazionale di una macchina di Turing)»<sup>12</sup>. Di Francesco va un po' oltre la questione e afferma che «l'idea di Turing riguarda l'ontologia della mente: se anche l'intelligenza artificiale si rivelasse un programma completamente fallimentare, il modello computazionale non crollerebbe automaticamente con essa»<sup>13</sup>, ovvero il reale interesse di Turing non sta nell'affermare, con atteggiamento comportamentista, che la macchina pensa se imita il comportamento umano bensì che «noi pensiamo perché il pensiero è una forma di computazione»<sup>14</sup>, rivelando la propria attenzione all'uomo più che alla macchina.

«Nell'articolo *Macchine calcolatrici e intelligenza* (1950) Turing propose di sostituire la domanda “Le macchine possono pensare?” con il seguente problema: è possibile progettare una macchina che sia estremamente difficile identificare come tale nel corso di una conversazione “cieca”, in cui cioè l'interlocutore umano non vede il suo partner, e comunica con lui solo attraverso messaggi scritti? [...] L'opinione di Turing era che per la fine del secolo i progressi della tecnologia avrebbero reso naturale parlare di macchine pensanti»<sup>15</sup>.

E' però giunto il momento di affrontare il lato più tecnico, anche se di modello ideale si tratta, della Macchina di Turing, modello che poi verrà utilizzato da von Neumann per realizzare il vero e proprio calcolatore.

---

<sup>11</sup> *Johnson-Laird*, p. 58.

<sup>12</sup> *Frixione*, p. 59. Se non fosse chiaro il riferimento al comportamentismo, Frixione precisa nella stessa pagina che «lo stesso test di Turing è stato a lungo considerato come il criterio più adeguato per stabilire la validità empirica dei modelli computazionali della mente. In realtà è ancora preistoria, ancora pesantemente condizionata da assunzioni di tipo comportamentista».

<sup>13</sup> *Di Francesco*, p. 143.

<sup>14</sup> *Di Francesco*, p. 143.

<sup>15</sup> AA. VV., alla voce «Turing».

In Burattini-Cordeschi si legge: «Gli aspetti fondamentali di questa descrizione informale e intuitivamente visualizzabile di una macchina di Turing possono essere espressi in un linguaggio più astratto e rigoroso. Si consideri la lista di simboli  $s_1, s_2, s_3, \dots; q_1, q_2, q_3, \dots; D, S$ . Un'espressione è una successione finita di simboli presi da questa lista. Un'istruzione è una *quintupla*, cioè un'espressione della forma  $q_i s_j s_k q_l M$  dove  $M$  sta per la lettera  $D$  o per la lettera  $S$ . Intuitivamente, un'istruzione si interpreta in questo modo: se la configurazione interna è  $q_i$  e il simbolo osservato è  $s_j$ , allora la macchina scriverà al suo posto il simbolo  $s_k$  entrando nella configurazione interna  $q_l$  e spostandosi di una casella, a destra o a sinistra, a seconda che  $M$  sia  $D$  oppure  $S$ . Si conviene generalmente che il simbolo  $s_l$  indichi la casella vuota; pertanto rimpiazzare  $s_j$  con  $s_l$  equivale all'azione di cancellare  $s_j$ . *Definizione*: Una macchina di Turing è un insieme finito (ma non vuoto) di quintuple che non contiene nessuna coppia di quintuple che coincidano nei loro primi due simboli»<sup>16</sup>. Accolta la definizione formale della MT, ci interessiamo ora alla descrizione più puntuale delle sue parti, avvalendoci dell'aiuto di Marconi che ne descrive compiutamente la struttura.<sup>17</sup> Una MT è costituita da un nastro di lunghezza idealmente infinita diviso in caselle e di un «cursore» che si sposta di casella in casella con capacità di lettura o scrittura, per ciascuna di esse, di un simbolo per volta. Il cursore può trovarsi in un numero finito di *stati interni*, compreso uno *stato finale*. La macchina esegue istruzioni con la forma di quintuple (insiemi ordinati di cinque elementi) di cui abbiamo già detto sopra. Marconi chiarisce sinteticamente che «in ciascuno stato in cui può trovarsi e per ciascun simbolo che può accaderle di leggere, la macchina 'sa cosa fare', e c'è una sola cosa che può fare»<sup>18</sup>. Naturalmente nessuna istruzione può cominciare con lo «stato finale» di cui sopra.

Utile ora ricorrere alle precisazioni fondamentali che Tabossi fa nel suo testo.<sup>19</sup> La macchina può compiere solo tre operazioni, ovvero:

1. leggere il simbolo della casella in esame;
2. scrivere un altro simbolo al posto di quello;
3. spostare il nastro di una casella a destra o a sinistra.

Per eseguire qualsiasi istruzione nella macchina di Turing, è necessario che tali istruzioni specifichino:

---

<sup>16</sup> Burattini-Cordeschi, p. 20.

<sup>17</sup> Per la descrizione della MT che segue, cfr. Marconi, pp. 38-43.

<sup>18</sup> Marconi, p. 38.

<sup>19</sup> Cfr. Tabossi, p. 42.

- a. lo stato in cui si trova la macchina, esprimibile convenzionalmente con un numero (ad esempio, una certa configurazione d'apertura o chiusura degli interruttori di un circuito elettrico);
- b. il simbolo nella casella in esame (1 o 0);
- c. l'operazione da eseguire (1, 2 o 3 cui sopra);
- d. lo stato in cui si troverà la macchina a operazione avvenuta.

Marconi<sup>20</sup> menziona un aspetto importante della MT e della dicitura di “stato interno” che può indurre in errore concettuale, cioè: la dizione ‘stati interni’ può far pensare che ‘dentro’ la macchina succeda qualcosa, che essa abbia una struttura interna che va al di là del nastro e del cursore con le sue capacità di lettura, scrittura e spostamento, ma non è così dato che gli stati interni servono soltanto a far dipendere l'azione della macchina oltre che dal simbolo che legge anche dal “momento” in cui lo legge ovvero dalla fase della computazione in cui lo legge. Aggiunge che «se non avessimo a disposizione gli stati, alla lettura di uno stesso simbolo corrisponderebbe sempre la stessa azione della macchina; e la macchina non potrebbe terminare nessuna computazione, ma continuerebbe inesorabilmente a scrivere (o eventualmente cancellare) simboli, a seconda del simbolo che legge»<sup>21</sup>.

E' il caso di portare un esempio del tutto indicativo di un ipotetico procedimento di calcolo in una Macchina di Turing. Con l'aiuto di Tabossi<sup>22</sup>, vediamo come sommare «2 + 1» con una MT. Saranno sufficienti otto istruzioni:

- I. la prima specifica
  - a. la configurazione di partenza (*stato 1*),
  - b. il simbolo in esame («1»),
  - c. l'operazione da compiere (sostituire «1» con «0»),
  - d. lo stato risultante (ancora «1»);
- II. nella seconda istruzione, con stato di partenza uguale a «1»,
  - a. il simbolo letto è «0»,
  - b. l'operazione è uno spostamento del nastro di una casella a sinistra;
- III. nella terza (*stato 2*), si sposta il nastro ancora di una casella a sinistra;
- IV. la quarta modifica il simbolo in esame in «1» (*stato 2*);

---

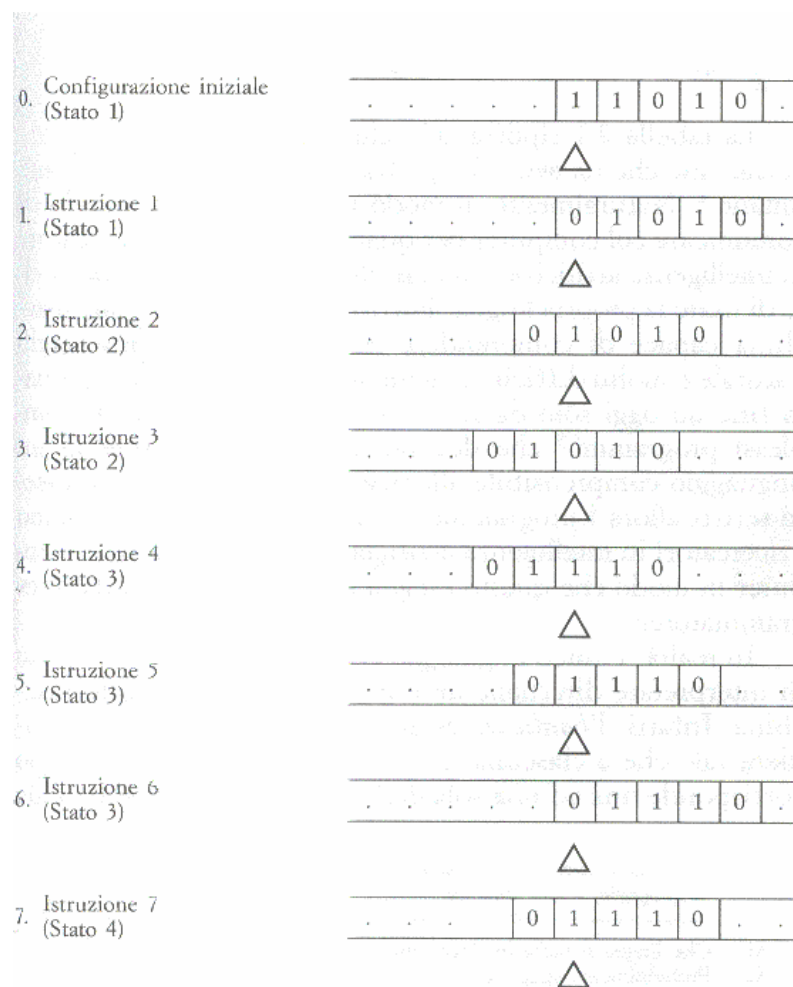
<sup>20</sup> Cfr. Marconi, pp. 40-41.

<sup>21</sup> Marconi, p. 41.

<sup>22</sup> Cfr. Tabossi, pp. 42-43.

- V. la quinta sposta il nastro di una casella a destra (*stato 3*);
- VI. la sesta istruzione invece (ancora *stato 3*)
  - a. sposta nuovamente il nastro a sinistra,
  - b. legge il simbolo «0»,
  - c. passa perciò dallo *stato 3* allo *stato 4*;
- VII. la settima istruzione rappresenta lo stato finale in cui il nastro resta dopo anche la prossima, l'ultima, istruzione;
- VIII. l'ottava e ultima infatti (sempre *stato 4*) sposta di nuovo il cursore a destra di una casella per presentare il risultato della somma computata.

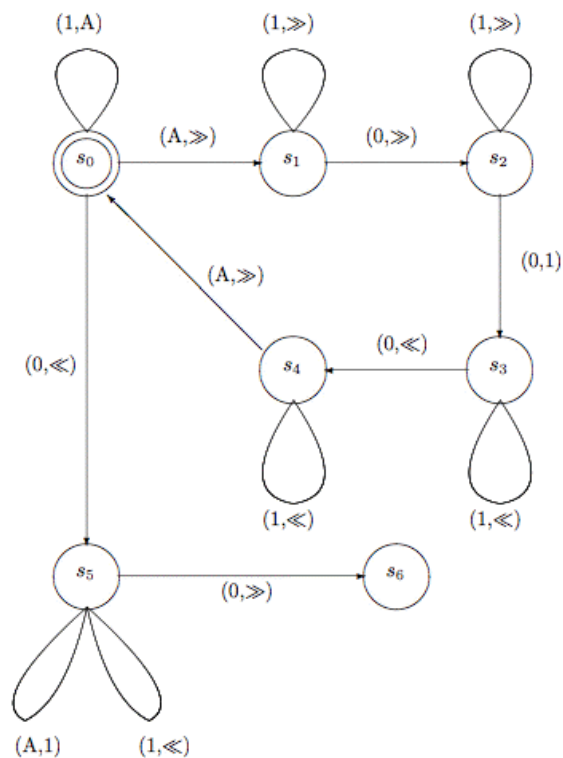
Naturalmente elaborati in binario, i numeri rappresentanti «2» e «1» erano «11» e «1» separati e seguiti da una casella vuota («0»): dunque il risultato è «111» cioè «3» in forma decimale.<sup>23</sup>



**Figura 1.** Esempio di somma nella Turing Machine di «2 + 1».

<sup>23</sup> Vedi **Figura 1**, tratta da *Tabossi*, p. 43.

Ulteriore esempio può essere il seguente: una procedura per la copia di un blocco di molti «1», rappresentata tramite un diagramma di stato.<sup>24</sup> Nella figura gli stati sono rappresentati dai cerchi, mentre doppio cerchio indica lo stato iniziale. Una transizione è rappresentata da una freccia che parte da un cerchio per arrivare ad un altro (anche lo stesso). Le frecce sono etichettate da due simboli: il primo è quello che deve essere letto ed eseguito dalla freccia e il secondo è l'azione da eseguire mentre si compie la transizione. L'azione sarà il simbolo da scrivere oppure potrà essere «or» che indica lo spostamento a destra o sinistra.



**Figura 2.** Diagramma di stato di una procedura di copia di un blocco di molti «1».

Certamente una macchina di Turing coincide con le sue istruzioni.<sup>25</sup> Questo permette che la codifica di tali istruzioni su un alto nastro di un'altra MT rappresenti l'esecuzione del compito di una MT da parte di un'altra MT: le macchine di Turing «universali» simulano così qualsiasi MT ideata. Un computer si può considerare una realizzazione fisica «parziale» di una MT universale, nel senso che il nastro infinito di una MT non può essere realizzabile: la memoria di un computer, per quanto grande e sempre più grande, non è mai infinita. Per il resto, in un computer le singole macchine di Turing – cioè le istruzioni – sono rappresentate dai programmi che il computer può eseguire, mentre

<sup>24</sup> Vedi **Figura 2**, tratta dal sito «<http://plato.stanford.edu/entries/turing-machine/>».

<sup>25</sup> Cfr. *Marconi*, p. 42.

la CPU (l'unità centrale di elaborazione dei dati) e il sistema operativo sono la macchina universale che simula le altre macchine, difatti eseguono i programmi. «Dunque sia sul piano storico che sul piano teorico il concetto di macchina di Turing costituisce la mediazione tra la nozione di procedimento effettivo (algoritmo) e il computer. Concettualmente, è per il fatto di realizzare una macchina di Turing universale che un computer è in grado di eseguire *qualsunque* calcolo»<sup>26</sup>. Infatti aggiunge Tabossi che «in linea di principio tutte queste macchine [s sofisticate di oggi] sono equivalenti tra loro in quanto tutte parimenti imitabili dalla macchina universale ideata da Turing nel lontano 1936, quando il primo calcolatore era ancora di là da venire»<sup>27</sup>.

Per concludere, accenno al probabile scopo ultimo dell'*Imitation game* proposto da Turing, riallacciandomi all'incipit del mio discorso: come ci indica Williams, Turing stesso fa passare il messaggio che «l'intelligenza, come la bellezza, è nell'occhio di chi guarda»<sup>28</sup>.

### Riferimenti bibliografici

**AA. VV.**, *Enciclopedia Garzanti di Filosofia*, Edizione Mondolibri su licenza Garzanti, Milano 1993

**Burattini E., Cordeschi R.** (a cura di), *Intelligenza artificiale. Manuale per le discipline della comunicazione*, Carocci, Roma 2001

**Di Francesco, M.**, *Introduzione alla filosofia della mente*, Carocci, Roma 1996

**Frixione, M.**, *Logica, significato e intelligenza artificiale*, Franco Angeli, Milano 1994

**Johnson-Laird, P. N.**, *La mente e il computer. Introduzione alla scienza cognitiva*, trad. di P. Tabossi, Il Mulino, Bologna 1988

**Marconi, D.**, *Filosofia e scienza cognitiva*, Laterza, Roma-Bari 2001

**Pratt, V.**, *Macchine pensanti. L'evoluzione dell'intelligenza artificiale*, trad. di M. T. Bolla e M. L. Sapino, Il Mulino, Bologna 1990

**Tabossi, P.**, *Intelligenza naturale e intelligenza artificiale*, Il Mulino, Bologna 1988

**Williams, S.**, *Storia dell'intelligenza artificiale. La battaglia per la conquista della scienza del XXI secolo*, trad. di A. Antonini, Garzanti, Milano 2003

---

<sup>26</sup> Marconi, p. 43.

<sup>27</sup> Tabossi, p. 43.

<sup>28</sup> Williams, p. 33.