

Nicola Turrini

Abitare l'incomputabile. Alcune considerazioni sul programma di ricerca di Alan Turing

A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine.

A.M. Turing

1. Numeri (non)computabili

“I believe that in about fifty years’ time it will be possible to programme computers, with a storage capacity of about 109, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent. chance of making the right identification after five minutes of questioning”.¹ Così si esprimeva Alan Turing verso la fine degli anni Quaranta, periodo in cui l’idea che i computer digitali avrebbero potuto riprodurre il pensiero umano era chiaramente giunta a maturazione. Com’è noto, nel celebre articolo pubblicato su *Mind* nel 1950, *Computing Machinery and Intelligence*, egli propose il “gioco delle imitazioni” – diventato poi il “Test di Turing” – per misurare l’intelligenza di simili macchine. Sebbene il test sia stato talvolta considerato come una “operational definition of ‘thinking’ or ‘intelligence’ or ‘consciousness’ by means of a sexual guessing game”,² agli occhi del matematico britan-

¹ A.M. Turing, *Computing Machinery and Intelligence*, in “*Mind*”, v. LIX, n. 236, 1950, p. 442.

² Cfr. A. Hodges, *Alan Turing: The Enigma*, Princeton University Press, Princeton-Oxford 2014 p. 523 (edizione ampliata rispetto a quella originale del 1983). Sul test di Turing e sul Pensiero di Turing in generale cfr.: J. Floyd, A. Bokulich (eds.), *Philosophical Explorations of the Legacy of Alan Turing*, Springer, Cham 2017; R.I. Soare, *Turing Computability. Theory and Applications*, Springer-Verlag Berlin Heidelberg 2016; G. Sommaruga, T. Strahm (eds.), *Turing’s Revolution. The Impact of His Ideas about Computability*, Birkhäuser, Basel 2010; R. Epstein, G. Roberts, G. Beber (eds.), *Parsing the Turing Test. Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Springer, Cham 2009; C. Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*, Springer, Berlin-Heidelberg 2004.

nico la sua finalità era molto più modesta: esso doveva fornire solamente una condizione sufficiente affinché il “comportamento [behaviour]” di una macchina potesse essere considerato intelligente o pensante.

Turing giunse a queste posizioni a partire dal suo lavoro nel campo della logica matematica iniziato negli anni Trenta; un lavoro che lo aveva portato alla formulazione della Tesi di Church-Turing e del concetto di Macchina di Turing Universale – idee che ebbero successivamente un’influenza fondamentale sul suo pionieristico programma di ricerca sull’intelligenza artificiale. Più in particolare, le idee di Turing sull’intelligenza meccanica affondano le loro radici nel suo interesse per i metodi matematici di dimostrazione da una parte, e per l’analisi delle capacità, dell’ampiezza e dei limiti del ragionamento matematico umano dall’altra. In quegli anni, Kurt Gödel, Alonzo Church e lo stesso Turing avevano infatti dimostrato che nessun insieme finito di regole poteva essere utilizzato per generare tutte le verità matematiche.

Oggi il concetto di computabilità – o ricorsività³ – insieme a una varietà di formulazioni della Tesi di Church-Turing si ritrovano in molti contesti, ben oltre il campo ristretto dell’informatica e delle scienze della computazione; si tratta tuttavia di declinazioni che non esistevano negli anni Trenta, e la cui proiezione sulle idee originali di Turing può portare a fraintendimenti, talvolta significativi. Per seguire lo sviluppo del suo programma di ricerca sull’intelligenza meccanica è essenziale comprendere il lavoro logico che svolse sul concetto di computabilità e di incomputabilità – concetti che, nel pensiero di Turing, rimarranno sempre intimamente connessi – nel suo contesto epistemologico e storico.

La prima formulazione della Tesi di Church-Turing (CTT) da parte di Turing⁴ – nel suo celebre articolo del 1936, *On computable numbers*,

³ Cfr. P. Odifreddi, *Classical Recursion Theory*, North-Holland, Amsterdam 1989.

⁴ La Tesi di Church-Turing prende ovviamente il suo nome dai matematici che la introdussero, per la prima volta, tra gli anni Trenta e gli anni Quaranta: Alonzo Church e Alan Turing. Il lavoro di entrambi su questo tema prese avvio dal tentativo di risolvere l’*Entscheidungsproblem* di David Hilbert e, per affrontare il problema, Church e Turing si preoccuparono di definire rigorosamente il concetto di algoritmo, arrivando indipendentemente agli stessi risultati ma percorrendo strade piuttosto diverse. Nel 1936, Church pubblicò un articolo nel quale definiva il λ -calcolo, e, qualche mese dopo, Turing pubblicò un saggio nel quale introduceva il concetto di Macchina di Turing, del quale poi si servì per dimostrare la non solvibilità dell’*Entscheidungsproblem*. I due matematici giunsero alla conclusione che i sistemi di calcolo automatico che avevano proposto – insieme ad altri che continuavano a venire alla luce, come quello proposto da Emil Post – erano equivalenti sotto il punto di vista della potenza computazionale e quindi che entrambi incarnavano il concetto stesso di algoritmo. Si noti, tuttavia, che sebbene la tesi di Church e la tesi di Turing siano equivalenti in questo senso, hanno comunque significati distinti e sono quindi due tesi diverse. L’approccio generale di Church e di Turing verso l’emergente scienza del calcolo automatico era infatti piuttosto differente: la tesi di Turing riguarda le “computing machines” – la “Turing Machi-

with an application to the Entscheidungsproblem – affermava che i numeri computabili da una Macchina di Turing (TM) “include all numbers which could naturally be regarded as computable”,⁵ o, in altre parole, che qualsiasi calcolo [*computation*] poteva essere effettuato da una TM. I suoi primi lavori sulla (non)computabilità, tuttavia, non implicavano la meccanizzabilità della mente; Turing giunse al problema di come meccanizzare la mente umana, e di come il pensiero potesse essere riprodotto da un computer, solo successivamente.

Le origini dell'articolo del 1936 risalgono probabilmente all'anno precedente, quando Turing frequentò le lezioni sui fondamenti della matematica tenute da Max Newman al King's College di Cambridge. Nel suo *memoir* di Turing per la Royal Society, Newman lega direttamente l'articolo del 1936 al tentativo di dimostrare rigorosamente che il problema della decisione per la logica del primo ordine è irrisolvibile in senso assoluto: “the application he had principally in mind, namely, the breaking down of the Hilbert programme”.⁶ Si tratta, ovviamente, dell'*Entscheidungsproblem* così come era stato formulato alcuni anni prima da David Hilbert nel suo programma di formalizzazione del ragionamento matematico in risposta alla *Grundlagenkrise der Mathematik* di inizio Novecento.⁷ La parte “decisionale” del programma di Hilbert aveva come obiettivo la definizione rigorosa di una procedura generale per decidere la verità o la falsità di un qualsiasi teorema matematico espresso in forma completamente simbolica.

Per dimostrare che non esiste una “procedura effettiva [*effective procedure*]” (un “processo decisionale [*decision process*]”) capace di risolvere l'*Entscheidungsproblem* – e decidere quindi la dimostrabilità logica del primo ordine – Turing aveva bisogno di dare una definizione precisa e adeguata dei concetti intuitivi di “procedura effettiva” e di “calcolabilità effettiva [*effectively calculability*]”, nozioni che i matematici avevano

ne”, come Church stesso la soprannominò nella sua recensione dell'articolo del 1936, mentre quella di Church non menzionava in alcun modo le macchine di calcolo. Cfr. A. Church, *An Unsolvability Problem of Elementary Number Theory*, in “American Journal of Mathematics”, v. 58, n. 2, 1936, pp. 345-363; Id., *A Note on the Entscheidungsproblem*, in “Journal of Symbolic Logic”, v. 1, n. 1, pp. 40-41; Id., *Review: A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem*, in “Journal of Symbolic Logic”, v. 2, n.1, 1937, pp. 42-43.

⁵ A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, in “Proceedings of the London Mathematical Society”, v. 42, n. 1, 1937, p. 230.

⁶ M.H.A. Newman, *Alan Mathison Turing 1912-1954*, in *Biographical Memoirs of Fellows of the Royal Society*, Vol. 1, Nov 1955, The Royal Society Publishing, London, p. 258.

⁷ Cfr. D. Hilbert, W. Ackermann, *Grundzüge der theoretischen Logik*, Springer, Berlin-Heidelberg 1928. Questo volume conteneva la prima esposizione in assoluto della logica del primo ordine e poneva il problema della sua completezza e decidibilità (*Entscheidungsproblem*).

spesso utilizzato in modo informale per descrivere funzioni calcolabili con metodi, per così dire, cartacei. Come significativamente ricorda Newman, la dimostrazione che Turing aveva in mente era una dimostrazione di impossibilità:

Many were convinced that no such process was possible, but Turing set out to demonstrate the impossibility rigorously. The first step was evidently to give a definition of ‘decision process’ sufficiently exact to form the basis of a mathematical proof of impossibility. To the question ‘What is a ‘mechanical’ process?’ Turing returned the characteristic answer ‘Something that can be done by a machine’, and he embarked on the highly congenial task of analyzing the general notion of a computing machine. It is difficult today to realize how bold an innovation it was to introduce talk about paper tapes and patterns punched in them, into discussions of the foundations of mathematics.⁸

Dimostrare la non solvibilità dell’*Entscheidungsproblem* – un risultato matematico essenzialmente negativo o limitativo – richiedeva di analizzare accuratamente la nozione generale di “computing machine”, nozione che Turing ha dovuto costruire praticamente da zero. Ed è precisamente in questo processo di definizione che le nozioni di computabilità e incomputabilità si saldano insieme: Turing raggiunge la prima solo dopo aver trovato la seconda.

2. Computabilità e macchine

Quando discuteva della nozione di “calcolo [*computation*]”, Turing intendeva sequenze di operazioni su simboli logici o matematici eseguite da esseri umani (o da dispositivi meccanici) secondo un numero finito di regole, senza l’intervento di intuizioni, creatività o ipotesi e la cui esecuzione generava sempre la soluzione corretta attraverso un numero finito di *step*.⁹ Il termine “calcolo” non si riferiva quindi a tutto ciò che

⁸ M.H.A. Newman, *Alan Mathison Turing 1912-1954*, cit., p. 256.

⁹ La terminologia utilizzata da Turing per spiegare il concetto intuitivo di “procedura effettiva” è piuttosto variegata ma sempre consistente: “computable” come “calculable by finite means” (A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, cit., p. 231), “effectively calculable” (Ivi, p. 263), “The class of problems capable of solution by the machine [...] are those problems which can be solved by human clerical labour, working to fixed rules, and without understanding” (A.M. Turing [1945], *Proposal for Development in the Mathematical Division of an Automatic Computing Engine (ACE)*, in B.E. Carpenter; R.W. Doran (eds.), *A.M. Turing’s ACE Report of 1946 and Other Papers*, MIT Press, Cambridge MA, 1986, pp. 38-39), “‘rule of thumb’ or ‘purely mechanical’” (A.M. Turing, *Intelligent Machinery*,

un matematico o una mente umana (o una macchina) potevano fare ma delimitava un'area specifica di operazioni mentali (o meccaniche). Il suo lavoro logico sulla (non)computabilità deve essere compreso nel contesto storico degli anni Trenta, quando non esistevano né computer digitali funzionanti né le scienze cognitive: a quel tempo, un "computer" era semplicemente una persona che eseguiva dei calcoli con l'ausilio di carta, matita e gomma.¹⁰

Turing utilizzava il termine "meccanico" come sinonimo di "effettivamente calcolabile": "A function is said to be 'effectively calculable' if its values can be found by some purely mechanical process".¹¹ Questa osservazione – se ci atteniamo alle spiegazioni di Turing – è legata al punto essenziale per cui il significato dei simboli non doveva influenzare i calcoli né avere alcun ruolo nell'esecuzione di una procedura efficace.

La spiegazione che Turing dà della nozione di calcolabilità effettiva in termini puramente meccanici non era inoltre intesa a tracciare una distinzione rispetto a procedure effettive ma non meccaniche, come fece Gödel alcuni anni più tardi; come ha sottolineato Gualtiero Piccinini, Turing non aveva infatti alcuna nozione di procedura efficace non meccanica, e affermare che la definizione fosse limitata alle procedure meccaniche – in contrapposizione a quelle non meccaniche – lo coinvolge in un dibattito iniziato dopo la sua epoca.¹²

Nella tesi di dottorato del 1939, Turing precisò il termine "effettivamente calcolabile" in questo modo:

testo pubblicato postumo in C.R. Evans, A.D.J. Robertson (eds.), *Cybernetics: Key Papers*, Butterworths, London 1968, p. 7), "definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner" (A.M. Turing, *Programmers' Handbook for the Manchester Electronic Computer Mark II*, with errata sheets 28 Mar. and 9 Jul. 1951, and Programme sheets, 1951, Contemporary Scientific Archives Centre, King's College Library, Cambridge, AMT-B-32: <https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-32>).

¹⁰ Esistevano alcune macchine di calcolo piuttosto sofisticate come i computer analogici. Turing era di certo a conoscenza dell'analizzatore differenziale di Manchester e, come ricorda Hodges, aveva in mente di utilizzarlo per trovare i valori della funzione zeta di Riemann (cfr. Cfr. A. Hodges, *Alan Turing: The Enigma*, cit., p. 179).

¹¹ A.M. Turing, *Systems of Logic Based on Ordinals*, in "Proceedings of the London Mathematical Society", Series 2, v. 45, 1939, p. 166.

¹² Cfr. G. Piccinini, *Alan Turing and the Mathematical Objection*, in "Minds and Machines" v. 13, 2003, p. 27; si veda anche G. Piccinini, *Computationalism, The Church-Turing Thesis, and The Church-Turing Fallacy*, in "Synthese", v. 154, 2007, pp. 97-120. Negli anni Sessanta Gödel ipotizzò la possibilità di definire procedure efficaci ma non meccaniche da distinguere dalle procedure meccaniche efficaci analizzate da Turing; una procedura efficace non meccanica avrebbe a suo avviso consentito al significato dei simboli di determinarne il risultato (cfr. K. Gödel, *Postscriptum to 'On Undecidable Propositions of Formal Mathematical Systems'*, in M. Davis (ed.), *The Undecidable*, Raven Press, New York 1965, pp. 72-73).

We shall use the expression ‘computable function’ to mean a function calculable by a machine, and we let ‘effectively calculable’ refer to the intuitive idea without particular identification with any one of these definitions. We do not restrict the values taken by a computable function to be natural numbers; we may for instance have computable propositional functions.¹³

In questo passo Turing sta cercando di spiegare il significato intuitivo di “effettivamente calcolabile” in termini di processi meccanici. Il termine “mechanical process”, utilizzato prima in modo intuitivo, viene poi identificato con la computabilità delle macchine di Turing:

We may take this statement literally, understanding by a purely mechanical process one which could be carried out by a machine. It is possible to give a mathematical description, in a certain normal form, of the structures of these machines. The development of these ideas leads to the author’s definition of a computable function, and an identification of computability with effective calculability.¹⁴

Si tratta, in effetti, di un passo in avanti rispetto all’articolo del 1936, dove Turing aveva introdotto la nozione intuitiva di numeri computabili non come quei numeri che erano computabili mediante un processo meccanico, ma come quei numeri “may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means”.¹⁵

3. Le Macchine di Turing

La nozione formale di Macchina di Turing entra in gioco proprio per definire rigorosamente il concetto di calcolabilità effettiva: una procedura è effettiva solo se può essere eseguita da una macchina di Turing. In questo caso, anche il termine “macchina” richiede di essere precisato. A seguito dell’articolo del 1936, i matematici più vicini a Turing interpretarono il suo articolo come una teoria generale della computabilità, che stabiliva cosa potesse e cosa non potesse essere calcolato, non solo dagli esseri umani ma

¹³ A.M. Turing, *Systems of Logic Based on Ordinals*, cit., p. 166 (cfr. anche Id., *Computability and [un]definability*, in “Journal of Symbolic Logic”, v. 2, n. 4, 1937, p. 153).

¹⁴ *Ibidem*. Cfr. anche A.M. Turing, *Lecture to the London Mathematical Society on 20 February 1947*, in B.E. Carpenter; R.W. Doran (eds.), *A.M. Turing’s ACE Report of 1946 and Other Papers*, cit., p. 107: “One of my conclusions was that the idea of a ‘rule of thumb’ process and a ‘machine process’ were synonymous. The expression ‘machine process’ of course means one which could be carried out by the type of machine I was considering”.

¹⁵ A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, cit., p. 230.

anche dai dispositivi meccanici;¹⁶ solo successivamente, diversi autori adottarono una posizione più restrittiva. Dato che l'obiettivo dell'articolo del 1936 era stabilire un limite a ciò che poteva essere ottenuto in matematica con metodi di prova effettivi, è chiaro che le macchine di Turing rappresentavano almeno le capacità computazionali degli esseri umani: le operazioni eseguite da queste macchine erano infatti determinate da un elenco di istruzioni che dovevano essere comprensibili in modo non ambiguo dagli esseri umani – che, dopotutto, avevano il compito di programmarle.

Leggendo l'articolo del 1936 ci si imbatte di frequente in descrizioni antropomorfe delle macchine, che Turing introduceva solitamente tra virgolette, molto probabilmente per indicarne l'uso metaforico: macchine che “scansionano [*scan*]” il nastro, “vedono [*see*]” simboli e sono dotate di “memoria [*memory*]” e “stati mentali [*states of mind*]”.¹⁷ L'utilizzo di simili figure del linguaggio negli scritti di Turing non è tuttavia una mera cifra stilistica: se è certamente vero che il suo stile argomentativo era piuttosto eccentrico rispetto ai matematici del suo tempo (e del nostro a dire il vero), l'uso di metafore non assume mai una funzione puramente evocativa ma svolge il ruolo di presentare al lettore precise questioni teoriche. Lo stesso Turing sembra per altro essere ben consapevole che la stessa macchina di Turing, al di là del suo statuto di modello formale rigoroso, funzioni anche come un paradigma metaforico.

Anche in questo caso, il punto è l'essenziale riferimento all'esecuzione, da parte di un computer umano, di una procedura effettiva in cui il significato non doveva avere alcun ruolo. Le macchine di Turing non erano descritte come capaci di comprendere le istruzioni, né tantomeno come dotate di intelligenza: una TM è insensibile al significato o, se si vuole, semanticamente vuota. Se l'esecuzione di un calcolo “significativo” – piuttosto che di un processo fisico “privo di significato” – presupponesse la comprensione

¹⁶ Cfr. ad esempio A. Church, *Review: A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem*, cit., e Id., *Introduction to Mathematical Logic*. Vol. 1, Princeton University Press, Princeton 1956, n. 119, p. 52: “The notion of effectiveness may also be described by saying that an effective method of computation, or algorithm, is one for which it would be possible to build a computing machine”. Si veda anche A.G.D. Watson, *Mathematics and Its Foundations*, in “Mind”, v. 47, n. 188, 1938, pp. 448 ss., M.H.A. Newman, *Alan Mathison Turing*, cit., p. 258). Secondo Kleene, “Turing's formulation comprises the functions computable by machines” (S.C. Kleene, *On Notation for Ordinal Numbers*, in “Journal of Symbolic Logic” v. 3, n. 4, 1938, pp. 150). Lo stesso John von Neumann pose, definendo la sua teoria degli automi finiti, identificò il problema affrontato da Turing come quello di dare “a general definition of what is meant by a computing automaton” (J. von Neumann, *The General and Logical Theory of Automata*, in Id. *Collected Works*. Vol. 5: *Design of Computers, Theory of Automata and Numerical Analysis*, Pergamon Press, London 1963, p. 313).

¹⁷ Cfr. A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, cit., pp. 117-118.

delle istruzioni, si dovrebbe concludere che solo gli esseri umani eseguono calcoli autentici. Una TM – in quanto “computing machine” – è piuttosto una rappresentazione astratta e idealizzata di un essere umano nell’atto del calcolo. È probabilmente il fraintendimento di questo punto che, tra gli altri, ha indotto alcuni autori a fornire un’interpretazione restrittiva della proposta di Turing, secondo cui la sua teoria riguarderebbe esclusivamente la (non)computabilità da parte degli esseri umani e non delle macchine. In questa prospettiva le macchine di Turing sarebbero allora – riprendendo un’espressione di Ludwig Wittgenstein – “humans who calculate”.¹⁸

Queste interpretazioni sono in aperto contrasto con l’uso esplicito che Turing fa dei termini “calcolo” e “macchina”. Nonostante egli abbia insistito sul fatto che le sue macchine potessero imitare qualsiasi routine umana, non affermò mai che tali macchine dovessero essere considerate come esseri umani astratti. Turing ha sempre considerato il calcolo come una manipolazione fisica dei simboli e la TM fu introdotta per definire matematicamente tale processo di manipolazione nel modo più rigoroso ed esatto possibile. Secondo l’uso che ne faceva Turing erano piuttosto le TM a diventare esse stesse dei dispositivi meccanici idealizzati, modelli formali di un processo di calcolo eminentemente fisico e pratico. È proprio in quanto modelli puramente formali che tali macchine – e le loro limitazioni – potevano essere studiate dal punto di vista matematico: il loro comportamento laplaciano era infatti definito con precisione in termini di operazioni discrete ed effettive.

4. L’implementazione fisica della macchina di Turing universale

Turing aveva probabilmente intravisto la possibilità di implementare fisicamente la sua macchina universale già nel 1935.¹⁹ Nel 1947, durante

¹⁸ Cfr. L. Wittgenstein, *Remarks on the Philosophy of Psychology. Bemerkungen über die Philosophie der Psychologie*, Vol. 1, G.E.M. Anscombe, G.H. von Wright (eds.), Blackwell, Oxford 1980, § 1096. Wittgenstein conosceva Turing, che frequentò il suo corso sui fondamenti della matematica nel 1939. Sui loro diversi punti di vista si vedano S.G. Shanker, *Wittgenstein versus Turing on the Nature of Church’s Thesis*, in “Notre Dame Journal of Formal Logic”, v. 28, n. 4, 1987, pp. 615-649 e D. Proudfoot, B.J. Copeland, *Turing, Wittgenstein and the Science of the Mind*, in “Australasian Journal of Philosophy”, v. 72, n. 4, 1994, pp. 497-519. Su questo punto, Robin Gandy è stato molto più radicale di Wittgenstein: “*Turing’s analysis makes no reference whatsoever to calculating machines. Turing machines appear as a result, as a codification, of his analysis of calculations by humans*” (R. Gandy, *The Confluence of Ideas in 1936*, in R. Henken (ed.), *The Universal Turing Machine: A Half-Century Survey*, Oxford University Press, Oxford 1988, pp. 83-84). La lettura di Gandy è di certo pertinente ma non sembra cogliere il ruolo essenziale che tale codifica assumeva nell’argomentazione di Turing.

¹⁹ Cfr. M.H.A. Newman, *Obituary Notice for Alan Turing*, in “The Times”, 16 June 1954. Questo aspetto è particolarmente evidente nel *resumé* che Turing preparò per la rivista

un intervento presso la London Mathematical Society, egli riprese il lavoro del 1936 come contenente il progetto di un computer digitale universale e come teoria che stabiliva l'efficacia e i limiti delle nuove macchine di calcolo:

Some years ago I was researching on what might now be described as an investigation of the theoretical possibilities and limitations of digital computing machines. I considered a type of machine which had a central mechanism, and an infinite memory which was contained on an infinite tape. This type of machine appeared to be sufficiently general. One of my conclusions was that the idea of a 'rule of thumb' process and a 'machine process' were synonymous [...]. Machines such as the ACE may be regarded as practical versions of this same type of machine.²⁰

Per Turing, una TM non era quindi una mera idealizzazione matematica di alcune capacità cognitive della mente umana: essa doveva essere propriamente intesa come un dispositivo meccanico ipotetico, dotato di un funzionamento specifico. Inoltre, egli riteneva che le sue macchine

francese *Comptes Rendues* e che conteneva una descrizione sintetica dei risultati del suo articolo del 1936. In questa sede la definizione di "computable" e il risultato principale sono puntualmente espressi in termini di macchine e gli esseri umani non vengono menzionati in alcun modo: "On peut appeler 'computable' les nombres dont les décimales se laissent écrire par une machine. [...] On peut démontrer qu'il n'y a aucun procédé général pour décider si une machine *m* n'écrit jamais le symbole 0" (A.M. Turing, Typescript copy made by Mrs Turing of first rough draft of précis of 'Computable Numbers', made for *Comptes Rendues*, 1936, Contemporary Scientific Archives Centre, King's College Library, Cambridge, AMT-K-4: <https://turingarchive.kings.cam.ac.uk/material-given-kings-college-cambridge-1960-amtk/amt-k-4>).

²⁰ A.M. Turing, *Lecture to the London Mathematical Society on 20 February 1947*, cit., pp. 106-107. Inoltre, nel testo del 1948, *Intelligent Machinery*, le macchine di Turing sono considerate "chiefly of interest when we wish to consider what a machine could in principle be designed to do" (A.M. Turing, *Intelligent Machinery*, cit., p. 6). In questo articolo, è piuttosto evidente che Turing, più che descrivere le sue macchine logiche come esseri umani che calcolano, pensi invece gli esseri umani come computer digitali universali: "It is possible to produce the effect of a computing machine by writing down a set of rules of procedure and asking a man to carry them out. Such a combination of a man with written instructions will be called a 'Paper Machine.' A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine (ivi, p. 9). Cfr. anche A.M. Turing, *Can digital computers think?*, Typescript with AMS annotations of a talk broadcast on BBC Third Programme, 15 May 1951, Contemporary Scientific Archives Centre, King's College Library, Cambridge, AMT-B-5: <https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-5>: "A digital computer is a universal machine in the sense that it can be made to replace [...] any rival design of calculating machine, that is to say any machine into which one can feed data and which will later print out results". Ciò suggerisce per altro che, per Turing, qualsiasi dispositivo matematicamente definito come in grado di fornire i valori di una funzione non calcolabile, cioè una funzione che nessuna TM potrebbe calcolare – come le "oracle machines" descritte nella tesi del 1939 – non potrebbe essere implementato fisicamente.

logiche definissero – e, di conseguenza, delimitassero – la potenza di calcolo di una qualsiasi macchina, e quindi ciò che essa poteva o non poteva fare. Questa posizione, tuttavia, non implicava che ogni sistema fisico fosse una macchina di Turing o potesse essere imitato da una di esse; ad esempio, il risultato di un ipotetico processo casuale non poteva essere replicato da nessuna TM, ma solo da una macchina contenente un “elemento casuale”.²¹

La Tesi di Church-Turing ha indotto Turing a passare dall’idealizzazione di un essere umano che calcola all’idea di un dispositivo meccanico immaginario ma formalmente possibile. Come ha mostrato Rodney Brooks, dopo il 1936 Turing modificò gradualmente la natura dell’astrazione su cui stava lavorando: dalle macchine universali del suo articolo del 1936 – ancora oggi il modello essenziale per tutti i moderni computer digitali – a semplici meccanismi computazionali adattabili che potevano apprendere autonomamente, e che erano concepiti come elementi computazionali attivi. In quest’ultimo caso, il modello utilizzato si ispirava evidentemente al cervello.²²

In questa direzione dovrebbero essere stabilite le implicazioni e la portata della tesi di Church-Turing, secondo cui i numeri calcolabili da una macchina di Turing includono tutti i numeri “which could na-

²¹ A.M. Turing, *Intelligent Machinery*, cit., p. 9; cfr. Anche A.M. Turing, *Computing Machinery and Intelligence*, cit., p. 438.

²² Cfr. R. Brooks, *The Case for Embodied Intelligence*, in S.B. Cooper, J. van Leeuwen, (eds.), *Alan Turing. His Work and Impact*, Elsevier, Amsterdam 2012, p. 499; riportiamo integralmente l’analisi particolarmente lucida di Brooks: “For me Alan Turing’s 1948 paper *Intelligent Machinery* was more important than his 1950 paper *Computing Machinery and Intelligence*. [...] The bulk of the paper gives examples of how simple computational mechanisms could be adaptable, could be taught, and could learn for themselves. The examples and mechanisms Turing used in this exposition where networks of active computational elements. Although he connected them back to the universal machines of his 1937 paper, it is remarkable in hindsight, how different this abstraction was than the one that he had previously introduced, of the central processing element with a tape memory – still the essential model for all modern digital computers. Here, instead, he used a model inspired by brains. One can only wonder how different our technological world might be if Turing had lived to fully develop this set of ideas himself. Others carried on this second tradition, but one must think that perhaps Turing’s intellectual influence might have been stronger as he would have been arguing against the approach that was adopted from his earlier work. For me, the critical, and new, insights in *Intelligent Machinery* were two fold. First, Turing made the distinction between embodied and disembodied intelligence. While arguing that building an embodied intelligence would be a ‘sure’ route to produce a thinking machine he rejected it in favor of disembodied intelligence on the grounds of technical practicalities of the era. Second, he introduced the notion of ‘cultural search’, that people’s learning largely comes from the culture of other people in which they are immersed. Modern researchers are now seriously investigating the embodied approach to intelligence and have rediscovered the importance of interaction with people as the basis for intelligence”.

turally be regarded as computable”.²³ Per stabilire la Tesi di Church-Turing, Turing aveva paragonato “a man in the process of computing a real number to a machine”,²⁴ costruendo così la sua argomentazione sui limiti cognitivi che condizionano un essere umano che esegue operazioni di calcolo. All’inizio di *On Computable Numbers* si legge che “the justification [of CTT] lies in the fact that the human memory is necessarily limited”.²⁵ Nell’argomentazione Turing utilizza – non a caso – i limiti sensoriali per giustificare la sua restrizione a un numero finito di simboli primitivi, e i limiti della memoria per giustificare la sua restrizione a un numero finito di “stati mentali”.²⁶

Questo tipo di argomentazioni a favore della Tesi di Church-Turing rispondono – come spesso accade nello stile argomentativo del matematico britannico – a considerazioni limitative. Detto altrimenti, la CTT non implica che tutte le operazioni di una mente umana siano computabili da una TM; Turing non ha mai affermato qualcosa di simile. La sua idea era, in fondo, molto più modesta: le operazioni di una TM “include all those which are used in the computation of a number” da un essere umano.²⁷ Poiché il concetto di processo umano di calcolo è, come quello di calcolabilità effettiva, essenzialmente intuitivo, Turing poteva affermare che “all arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically”.²⁸ Un altro modo per dire che la tesi di Church-Turing non era un teorema matematico ma piuttosto “something between a theorem and a definition”.²⁹

²³ A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, cit., p. 116. La generalità della terminologia di Turing, che si chiede “What are the possible processes which can be carried out in computing a number?” (ivi, p. 135), e la conclusione a cui voleva giungere, cioè l’assoluta non solvibilità dell’*Entscheidungsproblem*, sono un’ulteriore prova del fatto che né la nozione di Gödel di “procedura efficace ma non meccanica” né l’ipotesi di una macchina di calcolo più potente delle TM avevano qui alcun senso. Se esistessero procedure efficaci non meccaniche, cioè macchine che calcolano funzioni non calcolabili, forse una di esse potrebbe risolvere l’*Entscheidungsproblem*; ma secondo Turing il problema della decisione non può avere alcuna soluzione (ivi, p. 117). Su questo si veda G. Piccinini, *Alan Turing and the Mathematical Objection*, cit., pp. 26-27.

²⁴ Ivi, p. 117

²⁵ *Ibidem*.

²⁶ Cfr. ivi, pp. 135-136.

²⁷ Ivi, p. 118.

²⁸ Ivi, p. 135.

²⁹ A.M. Turing 1954 *Solvable and Unsolvable Problems*, in “Science News”, n. 31, 1954, pp. 7-23. La tesi di Church-Turing è generalmente considerata una tesi non dimostrabile per la quale esistono prove convincenti. Per un’attenta analisi dei “disaccordi” sulla dimostrabilità della tesi si veda J. Folina, *Church’s Thesis: Prelude to a Proof*, in “Philosophia Mathematica”, v. 6, n. 3, 1998, pp. 302-323.

5. Thinking is (not) an effective procedure

Dall'articolo del 1936 Turing trasse l'idea che le procedure effettive potevano essere eseguite dalle sue macchine; questo valeva non solo per procedure che operavano su simboli matematici, ma per qualsiasi procedura simbolica che fosse effettiva. Un aspetto particolarmente rilevante dell'universalità della TM era che, in linea di principio, potevano essere implementati anche programmi difettosi – che portavano a calcoli che non si arrestavano mai – e in generale programmi di cui non era possibile decidere a priori quali input portassero a un calcolo corretto; purché, ovviamente, tali procedure fossero definite in modo esaustivo da un insieme finito di istruzioni. Si tratta del ben noto Problema della fermata [*Halting Problem*] che, come Turing stesso ha dimostrato, risulta essere anch'esso irrisolvibile. L'insieme degli input che portano a un calcolo che termina è un insieme numerabile in modo ricorsivo, cioè è possibile implementare progressivamente la serie di tutti i calcoli possibili e osservare quali producono un risultato; questo permette di enumerare gli input per cui la macchina si ferma, ma questo insieme non è a sua volta un insieme computabile, poiché non si può mai essere certi che un calcolo in corso, un giorno, non avrà successo.

Se dotata delle tavole di istruzioni appropriate, una macchina universale poteva eseguire tutti questi processi. Si trattava di una tesi di grande forza speculativa, ma per nulla equivalente alla tesi – sostenuta ad esempio da Stuart C. Shanker – secondo cui “thinking is an effective procedure”.³⁰ In *On Computable Numbers* Turing non sosteneva che il pensiero umano potesse essere totalmente meccanizzato: al contrario, il

³⁰ Secondo Shanker, era questa l'idea di Turing (cfr. S.G. Shanker, *Turing and the Origins of AI*, in “*Philosophia Mathematica*”, v. 3, n. 1, 1995, p. 55). Turing, tuttavia, non ha mai fatto una simile affermazione. Cfr. W. Sieg, J. Byrnes, *K-graph Machines: Generalizing Turing's Machines and Arguments*, in P. Hájek (ed.), *Gödel '96: Logical Foundations of Mathematics, Computer Science and Physics – Kurt Gödel's Legacy*, Springer, Berlin 1996, p. 117; Sieg e Byrnes difendono giustamente l'idea che Turing non abbia mai sostenuto che i processi mentali non possano andare oltre quelli meccanici: “First, Turing analyzed *mechanical* processes of a human computer. The reduction of string machines or of K-graph machines to letter machines over a two-element alphabet does not show that *mental* processes cannot go beyond mechanical ones; it only shows that Turing machines can serve as a “normal form” for machines, because of the simplicity of their description”. Nella nota 13 illustrano lucidamente il motivo per cui le macchine di Turing sono particolarmente utili per l'analisi metamatematica e le esigenze della *proof theory*: “For this reason Turing machines are most suited for theoretical investigations. This state of affairs is analogous to that involving logical calculi: natural deduction calculi reflect quite directly the structure of ordinary arguments, but have a somewhat involved metamathematical description; in contrast, axiomatic logical systems are not suited as frameworks for direct formalizations, but – due to their simple description – are most suitable for metamathematical investigations”.

pensiero in quanto tale non può essere integralmente ridotto a procedura efficace.

In fondo, è la consapevolezza degli effetti limitativi legati alla non computabilità ad essere al centro della dimostrazione che nessuna macchina di Turing – e quindi nessun metodo uniforme ed effettivo –³¹ era in grado di risolvere il problema decisionale della logica del primo ordine. Turing aggiunse anche che un processo non uniforme poteva generare una sequenza non computabile, ovvero una sequenza che nessuna macchina di Turing sarebbe stata in grado di generare. In un passaggio poco commentato di *On Computable Numbers* Turing definisce una certa sequenza binaria infinita δ , che dimostra essere non calcolabile, e afferma: “It is (so far as we know at present) possible that any assigned number of figures of δ can be calculated, but not by a uniform process. When sufficiently many figures of δ have been calculated, an essentially new method is necessary in order to obtain more figures”.³²

I processi non uniformi, che generano oggetti matematici non computabili, compaiono sotto diverse forme nei lavori successivi di Turing sui fondamenti della matematica, sull'intelligenza artificiale e sulla morfogenesi, e hanno segnato in modo significativo il suo programma di ricerca sull'intelligenza artificiale. Se alcuni hanno messo in discussione la definizione di “teoria della computabilità” per la disciplina che Turing fondò negli anni Trenta insieme a Gödel, Church, Post e Kleene, è forse perché in essa il ruolo svolto dalla non computabilità e dalla non uniformità è essenziale.³³ Turing è stato un matematico che si è confrontato con i problemi del suo tempo e che ha cercato di dare sostanza matematica ai processi fisici e mentali, fornendoci il modello essenziale di ciò che oggi intendiamo per computazione. Osservò la computabilità come un'unità organica con l'incomputabilità, scoprendo quest'ultima come un'estensione della teoria della definibilità del teorema di incompletezza di Gödel; e, come ha giustamente affermato Barry Cooper “He encouraged us to see the universe as something that does compute and to engage with its features,

³¹ In questo contesto, un metodo di calcolo uniforme si riferisce al fatto che la macchina utilizza un insieme identico di regole o istruzioni indipendentemente dalle caratteristiche dell'*input* (come ad esempio le dimensioni), un concetto che Turing utilizzò per formalizzare il calcolo e successivamente esplorare la teoria della computazione. In un metodo di calcolo non uniforme, la macchina potrebbe ricevere nuove informazioni aggiuntive esterne al proprio insieme di istruzioni. In questa direzione si muove la tesi del 1939 sulle logiche ordinali. Sulla rilevanza dei “processi non uniformi” cfr. G. Piccinini, *Alan Turing and the Mathematical Objection*, cit.

³² A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, cit., p. 253.

³³ Sulla rilevanza della nozione di “incomputabile” cfr. S. Barry Cooper, M.I. Soskova (eds.), *The Incomputable Journeys Beyond the Turing Barrier*, Springer, Cham 2017.

tracing embodied analogues of his halting problem within biology and neuroscience”.³⁴

Nonostante l'apparente disincarnazione del lavoro del 1936 – che aveva motivazioni tecniche, matematiche e concettuali – successivamente, sin dai suoi anni a Bletchley Park, Turing si occupò in molti modi di quella che potremmo definire “embodied and disembodied intelligence”.³⁵ Implicito, in questo senso dell'incarnazione, è il riconoscimento di una differenza nel modo in cui gli esseri umani – e le macchine intelligenti in generale – interagiscono con le informazioni, ovvero vi sono immersi: “What we want is a machine that can learn from experience. The possibility of letting the machine alter its own instructions provides the mechanism for this”³⁶ ricordava Turing nel 1947. Questa interazione è considerata come un attributo necessario dell'intelligenza, ma solo se ricordiamo che quest'ultima non è solo “disciplina” ma anche “iniziativa”:³⁷ “if a machine is expected to be infallible, it cannot also be intelligent”.³⁸

Bibliografia

- Barry Cooper, S., Soskova M.I. (eds.), *The Incomputable Journeys Beyond the Turing Barrier*, Cham, Springer, 2017.
- Barry Cooper, S., van Leeuwen, J. (eds.), *Alan Turing. His Work and Impact*, Amsterdam, Elsevier, 2012.
- Barry Cooper, S., *Incomputability after Alan Turing*, in “Notices of the AMS”, v. 59, n. 6, 2012, pp. 776-784.
- Brooks, R., *The Case for Embodied Intelligence*, in S.B. Cooper, J. van Leeuwen, (eds.), *Alan Turing. His Work and Impact*, Amsterdam, Elsevier, 2012.
- Church, A., *An Unsolvable Problem of Elementary Number Theory*, in “American Journal of Mathematics”, v. 58, n. 2, 1936, pp. 345-363.
- Church, A., *A Note on the Entscheidungsproblem*, in “Journal of Symbolic Logic”, v. 1, n. 1, pp. 40-41.

³⁴ S. Barry Cooper, *Incomputability after Alan Turing*, in “Notices of the AMS”, v. 59, n. 6, 2012, p. 783. L'articolo del 1952 dedicato alla morfogenesi è, in questo senso, esemplificativo; cfr. A.M. Turing, Alan, *The Chemical Basis of Morphogenesis*, in “Philosophical Transactions of the Royal Society of London B”, v. 237, n. 641, 1952, pp. 37-72.

³⁵ Cfr. R. Brooks, *The Case for Embodied Intelligence*, cit.

³⁶ A.M. Turing, *Lecture to the London Mathematical Society on 20 February 1947*, cit., p. 123.

³⁷ Cfr. A.M. Turing, *Intelligent Machinery*, cit., p. 9: “If the untrained infant's mind is to become an intelligent one, it must acquire both discipline and initiative. So far we have been considering only discipline. To convert a brain or machine into a universal machine is the extremest form of discipline. [...] But discipline is certainly not enough in itself to produce intelligence. That which is required in addition we call initiative. This statement will have to serve as a definition. Our task is to discover the nature of this residue as it occurs in man, and try to copy it in machines”.

³⁸ A.M. Turing, *Lecture to the London Mathematical Society on 20 February 1947*, cit, p. 124.

- Church, A., *Introduction to Mathematical Logic*. Vol. 1, Princeton, Princeton University Press, 1956.
- Church, A., Review: A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, in "Journal of Symbolic Logic", v. 2, n.1, 1937, pp. 42-43.
- Epstein, R., Roberts, G., Beber G., (eds.), *Parsing the Turing Test. Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Cham, Springer, 2009.
- Floyd, J., Bokulich A., (eds.), *Philosophical Explorations of the Legacy of Alan Turing*, Cham, Springer, 2017.
- Folina, J., *Church's Thesis: Prelude to a Proof*, in "Philosophia Mathematica", v. 6, n. 3, 1998, pp. 302-323.
- Gandy, R., *The Confluence of Ideas in 1936*, in R. Henken (ed.), *The Universal Turing Machine: A Half-Century Survey*, Oxford, Oxford University Press, 1988.
- Gödel, K., *Postscriptum to 'On Undecidable Propositions of Formal Mathematical Systems'*, in M. Davis (ed.), *The Undecidable*, New York, Raven Press, 1965.
- Hilbert, D., Ackermann, W., *Grundzüge der theoretischen Logik*, Berlin-Heidelberg, Springer, 1928.
- Hodges, A., *Alan Turing: The Enigma*, Princeton-Oxford, Princeton University Press, 2014.
- Kleene, S.C., *On Notation for Ordinal Numbers*, in "Journal of Symbolic Logic" v. 3, n. 4, 1938, pp. 150-155.
- Newman, M.H.A., *Alan Mathison Turing 1912-1954*, in *Biographical Memoirs of Fellows of the Royal Society*, Vol. 1, Nov 1955, London, The Royal Society Publishing.
- Newman, M.H.A., *Obituary Notice for Alan Turing*, in "The Times", 16 June 1954.
- Odifreddi, P., *Classical Recursion Theory*, Amsterdam, North-Holland, 1989.
- Piccinini, G., *Computationalism, The Church-Turing Thesis, and The Church-Turing Fallacy*, in "Synthese", v. 154, 2007, pp. 97-120.
- Piccinini, G., *Alan Turing and the Mathematical Objection*, in "Minds and Machines" v. 13, 2003, pp. 23-48.
- Proudfoot, D., Copeland, B.J., *Turing, Wittgenstein and the Science of the Mind*, in "Australasian Journal of Philosophy", v. 72, n. 4, 1994, pp. 497-519.
- Shanker, S.G., *Turing and the Origins of AI*, in "Philosophia Mathematica", v. 3, n. 1, 1995, pp. 52-85.
- Shanker, S.G., *Wittgenstein versus Turing on the Nature of Church's Thesis*, in "Notre Dame Journal of Formal Logic", v. 28, n. 4, 1987, pp. 615-649.
- Sieg, W., Byrnes, J., *K-graph Machines: Generalizing Turing's Machines and Arguments*, in P. Hájek (ed.), *Gödel '96: Logical Foundations of Mathematics, Computer Science and Physics – Kurt Gödel's Legacy*, Berlin, Springer, 1996.
- Soare, R.I., *Turing Computability. Theory and Applications*, Springer-Verlag Berlin Heidelberg 2016.
- Sommaruga, G., Strahm T., (eds.), *Turing's Revolution. The Impact of His Ideas about Computability*, Basel, Birkhäuser, 2010.
- Teuscher (ed.), C., *Alan Turing: Life and Legacy of a Great Thinker*, Berlin-Heidelberg, Springer, 2004.

- Turing, A.M., *Solvable and Unsolvable Problems*, in "Science News", n. 31, 1954, pp. 7-23.
- Turing, A.M., *The Chemical Basis of Morphogenesis*, in "Philosophical Transactions of the Royal Society of London B", v. 237, n. 641, 1952, pp. 37-72.
- Turing, A.M., *Programmers' Handbook for the Manchester Electronic Computer Mark II, with errata sheets 28 Mar. and 9 Jul. 1951, and Programme sheets, 1951*, Contemporary Scientific Archives Centre, King's College Library, Cambridge, AMT-B-32: <https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-32>).
- Turing, A.M., *Can digital computers think?*, Typescript with AMS annotations of a talk broadcast on BBC Third Programme, 15 May 1951, Contemporary Scientific Archives Centre, King's College Library, Cambridge, AMT-B-5: <https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-5>.
- Turing, A.M., *Computing Machinery and Intelligence*, in "Mind", v. LIX, n. 236, 1950, pp. 433-460.
- Turing, A.M., *Intelligent Machinery*, in C.R. Evans, A.D.J. Robertson (eds.), *Cybernetics: Key Papers*, London, Butterworths, 1968.
- Turing, A.M., *Lecture to the London Mathematical Society on 20 February 1947*, in B.E. Carpenter; R.W. Doran (eds.), *A.M. Turing's ACE Report of 1946 and Other Papers*, Cambridge MA, MIT Press, 1986.
- Turing, A.M., *Proposal for Development in the Mathematical Division of an Automatic Computing Engine (ACE)*, in B.E. Carpenter; R.W. Doran (eds.), *A.M. Turing's ACE Report of 1946 and Other Papers*, Cambridge MA, MIT Press, 1986.
- Turing, A.M., *Systems of Logic Based on Ordinals*, in "Proceedings of the London Mathematical Society", Series 2, v. 45, 1939, pp. 161-228.
- Turing, A.M., *Computability and λ -definability*, in "Journal of Symbolic Logic", v. 2, n. 4, 1937, pp. 153-163.
- Turing, A.M., *Typescript copy made by Mrs Turing of first rough draft of précis of 'Computable Numbers', made for Comptes Rendues, 1936*, Contemporary Scientific Archives Centre, Cambridge, King's College Library, AMT-K-4: <https://turingarchive.kings.cam.ac.uk/material-given-kings-college-cambridge-1960-amtk/amt-k-4>).
- Turing, A.M., *On computable numbers, with an application to the Entscheidungsproblem*, in "Proceedings of the London Mathematical Society", v. 42, n. 1, 1937, pp. 230-265.
- Von Neumann, J., *The General and Logical Theory of Automata*, in Id. *Collected Works. Vol. 5: Design of Computers, Theory of Automata and Numerical Analysis*, London, Pergamon Press, 1963.
- Watson, A.G.D., *Mathematics and Its Foundations*, in "Mind", v. 47, n. 188, 1938, pp. 440-451.
- Wittgenstein, L., *Remarks on the Philosophy of Psychology. Bemerkungen über die Philosophie der Psychologie*, Vol. 1, G.E.M. Anscombe, G.H. von Wright (eds.), Oxford, Blackwell, 1980.

Abitare l'incomputabile. Alcune considerazioni sul programma di ricerca di Alan Turing

Verso la fine degli anni Quaranta, Alan Turing aveva maturato l'idea che i computer digitali avrebbero potuto riprodurre il pensiero umano e, per misurarne l'intelligenza, propose il celebre "gioco delle imitazioni", diventato poi il "Test di Turing". Sebbene sia stato talvolta considerato una definizione operativa di pensiero o intelligenza, nella prospettiva di Turing il test forniva solamente una condizione sufficiente affinché il comportamento di una macchina potesse essere considerato intelligente. Turing giunse a queste considerazioni a partire dal suo lavoro – iniziato negli anni Trenta – nel campo della logica matematica per mostrare la non solvibilità dell'*Entscheidungsproblem* proposto da David Hilbert. Questo articolo ambisce a riprendere alcune posizioni di Turing in relazione ai metodi matematici di dimostrazione, all'intelligenza e alle macchine di calcolo, per mostrare l'influenza fondamentale che ebbero sul suo pionieristico programma di ricerca sull'intelligenza artificiale.

PAROLE CHIAVE: Computabilità, incomputabilità, macchina di Turing, tesi di Church-Turing, intelligenza artificiale

Inhabiting the uncomputable. Some reflections on Alan Turing's research program

Towards the end of the 1940s, Alan Turing had developed the idea that digital computers could reproduce human thinking and, in order to measure their intelligence, he proposed the famous "imitation game", which later became known as the "Turing test". Although it has sometimes been considered an operational definition of thinking or intelligence, in Turing's view, the test only provided a sufficient condition for a machine's behaviour to be considered intelligent. Turing arrived at these conclusions based on his work, which began in the 1930s, in the field of mathematical logic to demonstrate the unsolvability of David Hilbert's *Entscheidungsproblem*. The present article aims to revisit some of Turing's ideas on mathematical methods of proof, intelligence and computing machines, in order to demonstrate the fundamental influence they had on his pioneering research programme on artificial intelligence.

KEYWORDS: Computability, uncomputability, Turing machine, Church-Turing thesis, artificial intelligence

