

LA COMPUTABILITÀ: ALGORITMI, LOGICA, CALCOLATORI

Marcello Frixione e Dario Palladino

[Carocci Editore – Roma, 2011]

Michele Herbstritt

Chi conosce già le *Bussole* di Carocci, sa cosa è lecito aspettarsi da *La computabilità: Algoritmi, Logica, Calcolatori*, di Marcello Frixione e Dario Palladino. Chi non le conosce, d'altra parte, avrà sicuramente di che stupirsi. Dando per scontato che ci si possa stupire tanto negativamente quanto positivamente, si può certamente dire che i modi di stupirsi per questo libro sono (almeno) due, diversi nelle cause e opposti negli effetti. Se, in quanto segue, lo “stupore” sembra un concetto troppo forte, si sostituisca a esso la “sorpresa” o l’“inaspettato”: dovrebbero andar bene ugualmente.

Il modo di dire *in cauda venenum* esercita indubbiamente un certo fascino, ma in questa sede si preferisce lasciarlo da parte, cominciando a trattare (brevemente) proprio a partire dalla “brutta sorpresa”, se così si può chiamare, che aspetta il lettore digiuno di *Bussole*, e magari, già parzialmente satollo di computabilità. I temi affrontati nelle cento pagine e poco più sono tra quelli che ci si aspetterebbe: introduzione agli algoritmi, esposizione del modello di Turing, definizione delle funzioni ricorsive, tesi di Church. E poco altro. Niente gradi di risolubilità e niente teoremi di Kleene. Chi di computabilità sa (o cerca) qualcosa di più, non lo può certo trovare qui.¹

Fortunatamente, le “brutte sorprese” hanno anche una funzione, si passi l'esagerazione, educativa. Affrontare la lettura di una *Bussola* con aspettative sbagliate non può che portare a delusioni. Ma quali sono, dunque, le giuste aspettative? A questo proposito viene in aiuto la quarta di copertina: «Chiare, essenziali, accurate: le guide Carocci per orientarsi nei principali temi della cultura contemporanea», questa la definizione ufficiale di cosa siano le *Bussole*, e poi, poco più in basso: «Il testo si propone di esporre i concetti fondamentali della computabilità senza presupporre alcuna conoscenza tecnica preliminare [...]». Ebbene, tenendo a mente queste premesse, le aspettative sono giustamente ridimensionate.

Si tratta quindi di ricominciare da capo, come si è detto, con le giuste aspettative. Ed è a questo punto che subentra il secondo motivo di stupore, la “bella sorpresa”, se così la si vuole chiamare. Perché se ci si accontenta della quantità e del grado di approfondimento degli argomenti trattati, non si può non rimanere colpiti dalla chiarezza con cui questi sono esposti. Tutta la trattazione è una sorta di piacevole passeggiata in cui il lettore è accompagnato per

¹Chi cercasse una discussione più completa e approfondita dei temi trattati può consultare (Frixione and Palladino, 2004), manuale che gli autori stessi indicano come fonte principale. In lingua inglese, invece, un manuale di teoria della computabilità sicuramente ottimo ma anche più tecnico e impegnativo è (Bools et al., 2007).

mano (talvolta addirittura portato in lettigia) verso la comprensione di temi, va detto, non sempre semplicissimi.

Il primo capitolo ha un carattere del tutto informale e introduttivo: il lettore che avesse anche solo una vaga idea del significato della parola *algoritmo* ha a sua disposizione una dozzina abbondante di pagine per farsene un'idea più precisa e rigorosa. Gli esempi, di carattere matematico e non, sono chiari e funzionali; la sezione relativa ai diagrammi di flusso molto utile in particolare per assimilare il concetto di ciclo (finito e non); l'accento ai metodi di codifica dei dati, facendo riferimento al mondo (più quotidiano) dei calcolatori elettronici, risulta efficace. Forse, gli autori avrebbero potuto mettere l'accento un po' di più sul fatto che i concetti di cui si tratta rimangono esplicitati solo a livello intuitivo e, *quindi*, che sarà necessario renderli rigorosi in seguito.

Come è naturale, il concetto (informale) di algoritmo viene collegato a quello di funzione (calcolabile), nel secondo capitolo. Qui una qualche abitudine al ragionamento e alla notazione matematici risulta senz'altro di una certa utilità, ma la trattazione è comunque portata avanti per gradi, a partire dalla definizione generale di funzione, raffinata poi in quella di funzione calcolabile (totale o parziale), per giungere con un'accelerata finale a mostrare addirittura che l'insieme delle funzioni calcolabili non esaurirà mai l'insieme delle funzioni aritmetiche (quest'ultima dimostrazione, fondata sul ragionamento diagonale *a là* Cantor, è effettivamente più impegnativa, perché presuppone un certo grado di familiarità con alcune nozioni più avanzate di teoria degli insiemi, come quella di cardinalità).

Una volta introdotti i concetti fondamentali della disciplina, non resta che “svelare l'assassino”, rivelando come sia possibile rendere rigorosi i concetti di algoritmo e di funzione calcolabile. Il terzo capitolo ha precisamente questo scopo, essendo dedicato alle macchine di Turing. Dopo una concisa introduzione storica, che dà un'idea del contesto scientifico in cui inserire il famoso articolo di Turing (1936), gli autori forniscono un'esposizione piuttosto tradizionale di che cosa siano le macchine di Turing e in cosa consista il modello computazionale che si basa su di esse (le funzioni T-computabili). Anche in questo caso, gli esempi non mancano e la trattazione risulta molto chiara. Il cosiddetto *problema della fermata*, in cui certamente risiede una parte del merito scientifico di Turing, viene rimandata al quinto capitolo. Prima di allora, è necessario presentare almeno un “complice”, un altro esempio di sostituito formale: le funzioni ricorsive.

Il quarto capitolo si occupa di funzioni ricorsive, e risulta essere il più impegnativo del libro. La classe delle funzioni ricorsive è definita a partire da quella delle funzioni ricorsive primitive, le quali sono sì funzioni calcolabili, ma non certo sufficienti a esaurire la classe di queste ultime (e due controesempi sono forniti a riguardo: la cosiddetta *funzione diagonale* e la funzione di Ackermann. Per ovviare ai problemi delle funzioni ricorsive primitive, si introduce il famoso operatore di minimalizzazione, ottenendo la classe delle funzioni ricorsive generali, vere candidate allo scopo di sostituire il concetto intuitivo di funzione calcolabile.

Come si è già detto, il quinto capitolo si occupa del problema della fermata. Ma non solo, ovviamente. Anzi, prima di trattare di problemi non risolvibili per mezzo di un algoritmo, risulta necessario citare la tesi di Church: i modelli rigorosi elaborati per sostituire il concetto intuitivo di algoritmo, così ci si può esprimere, colgono nel segno; in altri termini: le funzioni calcolabili sono tutte e sole quelle T-computabili/ricorsive generali. A favore della tesi di Church gli autori forniscono tre argomenti: quello di *evidenza euristica* (così la chiamano), relativo sostanzialmente al fatto che tutte le funzioni calcolabili a noi note sono effettivamente T-computabili/ricorsive generali; quello di carattere più logico, relativo al fatto che *tutte* le sistemazioni formali elaborate con lo scopo di sostituire il concetto intuitivo di algoritmo risultano equivalenti fra loro (si citano, fra gli altri, la lambda-ricorsività, Herbrand-Gödel

ricorsività e gli algoritmi di Markov); quello che fa perno sulla natura indipendente e sostanzialmente non matematica del modello di Turing. Una volta stabilita l'alta plausibilità della tesi di Church, non resta agli autori che introdurre la macchina di Turing universale e il noto procedimento diagonale che porta alla definizione di un problema insolubile per mezzo di macchine di Turing (il problema della fermata), e dunque, modulo tesi di Church, insolubile in maniera algoritmica.

Giunto in fondo al quinto capitolo, il lettore (precedentemente) digiuno avrà sicuramente un bel po' di materiale su cui riflettere e, in fin dei conti, potrebbe decidere di fermarsi, almeno temporaneamente, avendo appreso i basilari di teoria della computabilità. Per chi decidesse poi di continuare la lettura, il libro garantisce altri due capitoli, di carattere più generale, dedicati a mostrare alcuni dei collegamenti vigenti fra la teoria in questione e altri interessanti temi.

Il sesto capitolo, in una dozzina di pagine, vuole essere un'introduzione al problema dei Fondamenti della Matematica, a partire dalle geometrie non euclidee fino al teorema di incompletezza di Gödel (di cui è addirittura presente una dimostrazione). *L'excursus* storico è interessante, ma risulta un po' frettoloso e, forse, sarebbe stato più utile all'inizio del libro. Il teorema di incompletezza di Gödel ha senza dubbio un suo valore intrinseco che ne rende interessante l'esposizione a prescindere, ma la decisione di fornirne una dimostrazione (per quanto informale) in un libro introduttivo come questo forse non è stata delle migliori.

L'ultimo capitolo del libro, decisamente più azzeccato, è dedicato innanzitutto a mostrare come la teoria della computabilità sia strettamente in relazione con l'informatica, fino a potersi caratterizzare come fondamento teorico di quest'ultima. A questo scopo vengono introdotti i concetti informatici fondamentali (unità di *input* e di *output*, *CPU*, programma memorizzato, linguaggio di programmazione) e viene mostrato in che modo l'architettura di von Neumann (sulla quale si basano i calcolatori contemporanei) possa essere messa in relazione con la macchina di Turing universale, di cui si accennava nel quinto capitolo. Infine, non manca un accenno ad alcune questioni di complessità computazionale.

La seconda parte del capitolo si occupa di mostrare in che rapporto stia la teoria della computabilità con la scienza cognitiva. Quest'ultima viene introdotta da un punto di vista storico e concettuale come una reazione al comportamentismo fondata sui concetti e sugli strumenti della teoria della computabilità: il funzionalismo che caratterizzava la prima impostazione cognitivista considerava possibile una descrizione dei processi mentali in termini di computazioni, facendo astrazione dal supporto fisico che realizza i processi mentali e concentrando l'attenzione sulle proprietà logiche, i rapporti reciproci e funzionali degli stati mentali. L'impostazione funzionalista ha lasciato il posto a nuovi sviluppi, più attenti alla dimensione neurobiologica della mente (gli studi sul cervello) e a quella ecologica (*embodied mind*, rapporto fra mente e ambiente), ma gli autori sostengono che i recenti sviluppi non siano necessariamente in contraddizione con un'impostazione, in fondo, computazionale, citando a sostegno alcune interessanti riflessioni metodologiche dello scienziato cognitivo David Marr.

Riferimenti bibliografici

- Boolos, G. S., J. P. Burgess, and R. C. Jeffrey (2007). *Computability and Logic*. Cambridge University Press. 131
- Frixione, M. and D. Palladino (2004). *Funzioni, Macchine, Algoritmi*. Carocci. 131
- Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*. 132