APhEx 9, 2014 (ed. Vera Tripodi)
Ricevuto il: 12/11/2013
Accettato il: 03/01/2014
Redattore: Claudio Calosi & Pierluigi Graziani

APhEx
PORTALE ITALIANO DI FILOSOFIA ANALITICA
GIORNALE DI FILOSOFIA
NETWORK
N°9 GENNAIO 2014

R e c e n s i o n i

Scott Aaronson, **Quantum Computing since Democritus,** Cambridge, Cambridge University Press, 2013, pp. 370.

di Jacopo Tagliabue*

1. PERCHÉ LEGGERE QUESTO LIBRO

Quantum Computing since Democritus (QCD) non è un libro di filosofia, non è un libro di fisica e non è nemmeno (strettamente parlando) un libro di *quantum computing* (nonostante il titolo – ma d'altra parte il povero Democrito compare sì e no in tre pagine). Non è un libro di testo per un corso universitario (anche se nasce dalle lezioni dell'autore a Waterloo nel 2006, tuttora disponibili liberamente online),[1] ma non è nemmeno un saggio per il tipico *educated reader* (a meno che non sia *molto* *educated*). QCD è stato descritto da Dave Bacon di Google come un 'poema travestito da appunti di lezione', e non credo potremo fare meglio; tuttavia, se non è chiaro

* L'autore desidera ringraziare Guglielmo Feis, Luca Gasparri, Sarah Songhorian per commenti a una versione precedente di questa recensione.

[1] Si veda *http://www.scottaaronson.com/democritus/default.html*.

nemmeno ad un ontologo come classificarlo nella tassonomia editoriale standard, è chiarissimo quali siano i motivi per leggerlo – tre contenutistici, uno formale:

1) Il libro contiene una visione lucidissima della meccanica quantistica (QM) e del suo posto centrale nella moderna comprensione del mondo fisico e della nozione di computazione. Come ama dire Aaronson [Aaronson 2013, pp. 119-110], la QM è di facile comprensione una volta che si toglie la fisica dalla teoria: in particolare, è una «bellissima generalizzazione della teoria della probabilità» [Aaronson 20013, p. XVII] che porta naturalmente a una nozione di computazione (quella quantistica) che mette in dubbio alcune delle più radicate idee sull'argomento.

2) QCD è un'introduzione eccellente, ancorché *opinionated*, a un campo in continua crescita: come sottolineato nella prefazione [Aaronson 2013, p. XXIII], nei sette anni trascorsi dalle lezioni di Waterloo sono successe moltissime cose. Anche se ovviamente esistono già eccellenti (e voluminosi) manuali, QCD fornisce un'introduzione fresca e moderna alla materia, che non disdegna i particolari formali ma che non si dimentica mai della visione d'insieme che sta tratteggiando pagina per pagina – in altre parole, un'ottima ricognizione del territorio, da cui partire per successive esplorazioni personali.

3) Aaronson ha una sensibilità filosofica non comune per uno scienziato, anche se non sempre grande riverenza per la materia.[2] QCD contiene moltissime discussioni "filosofiche" (*spoiler alert*: l'argomento di Searle-Penrose [Aaronson 2013, Cap. 4], il problema di Newcomb [Aaronson 2013, Cap. 19], il libero arbitrio [Aaronson 2013, Cap. 19], tra gli altri) affrontate con l'occhio critico e formale del teorico della

[2] Vedi ad esempio i commenti sull'argomento di Kripke delle C-fibre [Aaronson 2013, pp. 52-53].

computazione. In particolare, QCD contiene alcune delle riflessioni presenti in Aaronson [2012]: complessivamente, l'immagine che emerge dal libro è che la teoria della complessità possa entrare di diritto nella cassetta degli attrezzi del filosofo di professione, al pari di logica e teoria della computabilità.

4) Seth Lloyd ha descritto il libro come 'ridicolmente spassoso'– e chi siamo noi per contestare Seth Lloyd? QCD è un rarissimo esempio di saggio scientifico/filosofico di alto livello in cui si ride a ogni pagina. Da ogni riga traspare il divertimento e la passione di Aaronson – *unapologetic nerd* se ce n'è uno – per la materia, e il lettore non può a sua volta che gioire del senso dell'umorismo che pervade l'opera. QCD è un antidoto a tutta quella accademia che si prende sul serio, nonché l'esempio nitido di come la serietà di un argomento *non* sia in alcun modo funzione della serietà con cui lo si può esporre.

In questa recensione presenteremo dapprima la struttura generale del volume (*Sezione II*), utilizzeremo quindi QCD (*Sezione III*) per introdurre alcune nozioni base che ci saranno utili nel valutare la portata filosofica e concettuale delle tesi proposte (*Sezione IV*). Nella sezione finale torneremo invece a parlare del progetto generale dell'autore per fare un bilancio di questo ambizioso volume.

2. La Struttura

QCD conta due prefazioni (una, che è la recensione dell'autore al suo stesso libro, e una "più seria") e 22 capitoli. Tuttavia, per i nostri scopi possiamo riassumere come segue la struttura logica del volume:[3]

Il mondo pre-QM (Cap. 1-8):

i) Logica, teoria degli insiemi, computabilità classica (i.e. cosa possiamo computare?)

ii) Complessità computazionale (i.e. cosa possiamo computare in modo *efficiente*?)

Il mondo quantistico (Cap. 9-12)

i) QM, teoria della probabilità e *quantum computing*

ii) Complessità computazionale e *quantum computing*

Applicazioni (Cap. 13-21)

i) Risposte ai critici

ii) Fisica

iii) Filosofia

3. Complessità e Quantum Computing 101

Ci sono due nozioni che occorre introdurre per poter affrontare con profitto l'analisi degli argomenti di QCD. La prima è la nozione di complessità computazionale, la seconda di *quantum computing*.

[3] In realtà i capitoli possono essere letti per la maggior parte in ordine sparso: l'ordine qui presentato è però conveniente dal punto di vista espositivo.

Consideriamo i seguenti "compiti di decisione", ovvero compiti la cui soluzione implica una risposta sì/no:

> C_1) Dato un elenco finito di persone e un nome, restituire SI se il nome è nell'elenco, NO in caso contrario.
>
> C_2) Data una formula in logica al primo ordine, restituire SI se la formula è una tautologia, NO in caso contrario.

Come il lettore informato immaginerà facilmente, (C_1) e (C_2) sono compiti profondamente diversi: (C_1) è a tutti gli effetti un problema risolvibile *tramite un algoritmo*, (C_2) invece no. Il grande – e stupefacente – lascito della teoria della computazione (vedi ad es. Boolos, Burgess, Jeffrey [2007]) è appunto quello di aver chiaramente distinto i compiti in linea di principio risolvibili da una procedura automatica da quelli invece che per sempre eluderanno la computazione. Il grande – e stupefacente – lascito della teoria della complessità computazionale è invece quello di aver distinto, all'interno della prima classe di compiti, quelli risolvibili *in modo efficiente* da quelli risolvibili in modo *non* efficiente (per un manuale eccellente, si veda Moore, Mertens [2011]). Prima di illustrare casi illustri di entrambe le classi, occorre essere chiari sulla nozione di efficienza. Dato un compito C, un algoritmo che risolve le istanze di C è *efficiente* solo se le risorse che utilizza scalano in maniera al massimo polinomiale rispetto alla grandezza dell'input; se esiste tale algoritmo, C è considerato un compito *facile*.

Supponiamo, per esempio, di avere un algoritmo A che impiega quattro passaggi per sommare due numeri di due cifre ciascuno. A è efficiente solo se all'aumentare della grandezza dei numeri, i passaggi aumentano in modo (al più) polinomiale: ad es., se A

somma due numeri di dieci cifre in venti passaggi, possiamo inferire che le risorse utilizzate (i.e. lo spazio sul foglio) scalano all'incirca come:

i) $R = 2(n)$

(dove *n* è il numero di cifre nel numero in input). Invece, se A somma due numeri di dieci cifre in 2^{10} passaggi, le risorse scalano all'incirca come:

ii) $R = 2^n$

La differenza è evidente: nel primo caso A permette di sommare praticamente 1234567890 e 2143658709: infatti la somma di questi numeri occuperà solo venti righe sul foglio; nel secondo caso, A ci richiederebbe di usare 1024 righe per risolvere il problema! E ovviamente, più i numeri crescono più la differenza tra (i) e (ii) – tra polinomiale ed esponenziale, tra efficiente e inefficiente – cresce: con numeri di ottanta cifre, (ii) richiede più passaggi del numero di atomi nell'intero universo! Ovviamente, l'identificazione tra "polinomiale" ed "efficiente" può essere contestata su molte basi: tuttavia, nel corso dei decenni tale distinzione si è rivelata sorprendentemente accurata in tutti i casi di interesse – solo per fare un esempio, il fatto che possiamo comprare QCD su Amazon con una carta di credito è *interamente* dovuto alla asimmetria tra moltiplicare 73 per 97, trovando 7081, e risalire a 73 e 97, avendo 1081: moltiplicare è facile, scomporre difficile [Aaronson 2013, pp. 102-106]. In altre parole, anche se "efficiente" e "polinomiale" non sono davvero perfettamente sinonimi, sono comunque concetti in larga parte sovrapponibili.

Quali sono casi famosi di problemi risolvibili in modo efficiente/non efficiente all'interno della nozione "classica" di computazione? Nella prima categoria abbiamo ad esempio:

E_1) Dato un numero intero P, determinare se P è primo o no.

E_2) Date due sequenze di DNA, sapere quante inserzioni/cancellazioni sono necessarie per passare da una sequenza all'altra.

E_3) Date una stringa e una grammatica *context-free*, scoprire se la stringa appartiene o meno alla grammatica.

Nella seconda abbiamo ad esempio:

NE_1) Dato un numero intero F, trovare la (unica) scomposizione in numeri primi di F.

NE_2) Data una formula Booleana in forma normale congiuntiva con al massimo 3 letterali per clausola determinare se esiste un'assegnazione che la soddisfa (i.e. il famoso problema "3-SAT").

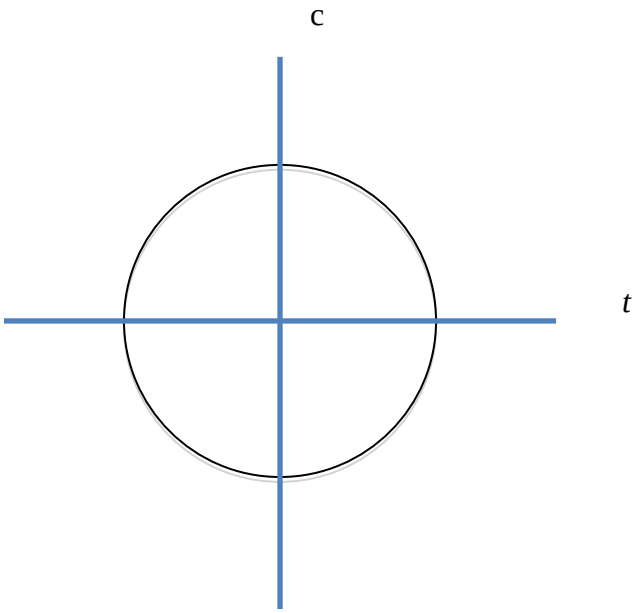
NE_3) Dato un grafo G con N vertici e K colori, determinare se G può essere colorato in modo tale che non ci siano due vertici adiacenti con identico colore.

In termini familiari a un informatico, i problemi del primo gruppo appartengono alla classe di complessità P (come "polinomiale"), quelli del secondo alla classe NP (tecnicamente, i problemi per cui trovare una soluzione è esponenzialmente difficile, ma per cui *verificare* una soluzione proposta è facile). *En passant*, quale sia l'esatta relazione tra queste due classi di complessità è uno dei Sette Problemi Matematici del Millennio e il più grande quesito aperto di tutta l'informatica teorica contemporanea.

Ora che abbiamo acquisito una certa familiarità con le basi concettuali della teoria della complessità, è giunto il momento di affrontare la computazione quantistica. Per contestualizzare il problema, ricordiamo brevemente cosa comporta una computazione classica. Prendiamo l'input del problema (ad es., due numeri da sommare), lo trasformiamo in un formato conveniente per la computazione (ad es., due stringhe di *bit*, ovvero di valori binari 0/1), applichiamo un elenco adeguato di operazioni logiche (ad es., operatori Booleani come AND, NOT, ecc.) alle nostre stringhe e quindi, ottenuto il risultato in *bit*, lo riconvertiamo in un formato facilmente leggibile (ad es., un numero in formato decimale). In computazione quantistica tutto avviene in modo molto simile, ma i *bit* sono sostituiti dai *qubit*, gli operatori logici standard dai *quantum gates*, e leggere il risultato di una computazione diviene a tutti gli effetti misurare un sistema quantistico. Qual è la differenza, dunque? *Bit* e *qubit* sono oggetti profondamente diversi e, seguendo Aaronson, proveremo a spiegare perché. Come anticipato, per Aaronson la QM non è una teoria fisica nel senso "classico" del termine: anche se è stata sviluppata per tentare di spiegare una serie di "anomalie" sperimentali ad inizio Novecento, sarebbe potuta in realtà venire scoperta dagli antichi Greci *senza nessun riferimento al mondo fisico*. La storia è andata diversamente, ma il cuore concettuale della QM altro non è che una naturale estensione della teoria della probabilità.

Consideriamo una moneta, che ha probabilità t di dare testa, c di dare croce: cosa possiamo dire di t e c *a priori*? Le due probabilità devono essere numeri reali maggiori o uguali a 0 e la somma dei loro valori assoluti deve essere 1 (ecco perché spesso in casi di esito binario le probabilità si indicano con p e $1 - p$). Ovviamente, ci sono molte somme diverse che potremmo decidere di volere mantenere: ad esempio, potremmo

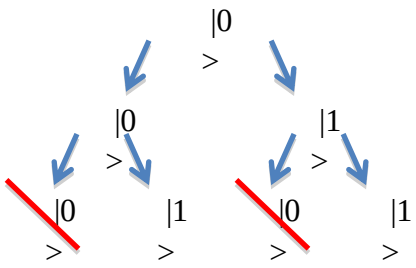
richiedere che la somma *dei quadrati* di *t* e *c* sia uno – in altre parole, ora vogliamo un vettore (*t*, *c*) tale che $t^2 + c^2 = 1$. L'insieme di tutti questi vettori è un cerchio [Aaronson 2013, p. 112]:



Rimane ora il problema di associare la teoria all'osservazione: nel nostro caso, dobbiamo garantire che quando *misuriamo* il nostro sistema, o *t* o *c* è ciò che viene osservato. Come passare dal vettore (*t*, *c*) a due numeri che, sommati, facciano 1? Semplice, prendiamo il quadrato di *t* e *c*! In effetti, questo strano oggetto che abbiamo "creato" è ovviamente il *qubit*, dove il vettore (*a*, *b*) generalmente viene indicato come $a|0\rangle + b|1\rangle$: *a* e *b* sono *amplitudini* e la probabilità di osservare 0 quando misuriamo il *qubit* è uguale ad a^2. Mentre un *bit* classico può essere solo in due stati (in due punti del cerchio), un *qubit* può avere per stato un qualsiasi punto della circonferenza.

Con questa semplicissima generalizzazione siamo già in grado di osservare uno dei fenomeni caratteristici della QM, l'interferenza: prendiamo lo stato $|0\rangle$ e applichiamo a

questo stato una trasformazione che ruoti il vettore nel piano di 45 gradi in senso antiorario; quello che otteniamo è l'equivalente di prendere una moneta non truccata e lanciarla, ovvero otterremo uno stato "random" in cui i due esiti sono equiprobabili (50/50). Ora applichiamo la stessa operazione un'altra volta: il risultato è |1>, cioè applicare una trasformazione di "randomizzazione" a uno stato "random" produce un risultato deterministico [Aaronson 2013, pp. 114-115]! Nonostante infatti ci siano due "strade" che portano al risultato |0>, una ha amplitudine positiva, l'altra negativa, le quali si cancellano[4] e fanno sì che il risultato non |0> non venga mai osservato:



È proprio nell'introduzione delle basi della QM che si evidenzia uno degli aspetti più eccitanti e "filosofici" della metodologia dell'autore – quella di chiedersi il perché di fatti apparentemente *bruti*: non basta sapere che gli esperimenti contraddicono la nostra intuizione, occorre chiederci come modificare le nostre intuizioni, cioè domandarsi «come avrei dovuto ragionare perché il risultato degli esperimenti non fosse in realtà una sorpresa?» [Aaronson 2013, p. XVI]. A tal proposito, un espediente retorico più volte utilizzato dall'autore è il seguente: supponendo di essere una divinità alle prese con la creazione del mondo, perché mai dovrei decidere di costruire un mondo fatto esattamente così [Aaronson 2013, pp. 116]?

[4] Le amplitudini in gioco in questo fenomeno divengono chiare se si scrive per intero la trasformazione in oggetto: per chi volesse approfondire, si veda Aaronson [2013], p. 114.

Nel nostro caso, la prima domanda da porsi è ovviamente: perché mai dovremmo prendere il quadrato e non il cubo (o qualsiasi altra cosa) come base per il nostro vettore? La seconda domanda è ancora più sottile: anche se abbiamo lavorato fin qui usando numeri reali, sappiamo che le amplitudini nella QM sono numeri complessi – per quale motivo? Rimandiamo il lettore curioso alle eccellenti spiegazioni dell'autore in merito [Aaronson 2013, pp. 116-123].

Prima di concludere la nostra ricognizione in questo stimolante territorio, siamo pronti per ricondurre le nuove nozioni alla teoria della computabilità e alla teoria della complessità: cosa possiamo fare *in più* e cosa possiamo fare più velocemente grazie ai *qubit*? Prima la cattiva notizia: con abbastanza tempo e risorse, un computer classico può emulare un qualsiasi computer quantistico, anche se con lentezza esponenziale (un computer quantistico può essere in una sovrapposizione qualsiasi di 2^n stati diversi, mentre un computer standard può essere solo in uno di questi 2^n stati in ciascun istante). In altre parole, la computazione quantistica è soggetta alle stesse limitazioni di quella classica quando parliamo di funzioni computabili e non. La buona notizia – nonché una delle scoperte scientifiche più sorprendenti della fine del secolo scorso – è che esistono problemi di grande rilevanza pratica in cui un computer quantistico sarebbe significativamente superiore rispetto a un computer classico. Abbiamo prima affermato che scomporre un numero in fattori primi è un problema non risolvibile in modo efficiente con la computazione classica: proprio questo problema – al centro della moderna crittografia informatica – è risolvibile in modo efficiente (cioè polinomiale) da un computer quantistico, grazie all'algoritmo scoperto dal matematico Peter Shor nel 1994.

4. APPLICAZIONI

L'apparato teorico sviluppato per spiegare i principi della computazione quantistica è funzionale a due cose: in primo luogo, in considerazione di risultati come l'algoritmo di Shor, appare sempre più chiaro che se davvero riuscissimo a costruire un computer quantistico generale e scalabile – ovvero in grado di gestire un numero significativo di *qubit* – le ricadute pratiche per l'umanità sarebbero enormi. Non troppo sorprendentemente, tuttavia, non c'è quasi traccia di questa tensione ingegneristica in QCD:[5] l'Autore è per abilità, storia personale e temperamento molto più interessato ad esplorare le conseguenze concettuali della rivoluzione quantistica piuttosto che seguirne i pur interessanti sviluppi pratici. In più punti, Aaronson dichiara che sarebbe molto più affascinato se risultasse un domani che i computer quantistici *non* si possono costruire: «tale fallimento implicherebbe che c'è qualcosa di sbagliato o incompleto nella nostra comprensione base della meccanica quantistica: una rivoluzione per la fisica!» [Aaronson 2013, p. XXIII].

Avendo delegato agli ingegneri e ai fisici il compito di produrre risultati pratici, Aaronson è così libero di utilizzare le nozioni che ha introdotto per illuminare, spesso in modo ardito e provocante, nodi concettuali a cavallo tra matematica, fisica, Intelligenza Artificiale e filosofia. Per ragioni di spazio, ci limiteremo a illustrare la metodologia dell'autore in due argomenti chiave, affrontati nelle due sotto-sezioni seguenti.

[5] Il problema principale è la cosiddetta *decoherence*, poiché i computer quantistici mantengono le loro proprietà solo se vengono "isolati" dal mondo esterno. Allo stato attuale sono stati costruiti solo computer con un piccolo numero di *qubit*: per citare alcuni risultati, nel 2001 si è riusciti a scomporre in fattori primi 15 [Vandersypen et al., 2001], nel 2011 si è riusciti a scomporre 143 [Xu et al., 2011]. A onore del vero, un recente risultato sperimentale si deve proprio all'implementazione di un'idea di Aaronson [Broome et al., 2013]. Per la controversia su D-Wave, l'azienda privata che sostiene di aver prodotto il primo computer quantistico commerciale, si veda il blog personale di Scott Aaronson.

4.1 Computazione e filosofia della mente

Il primo argomento è quello della Stanza Cinese, proposto da John Searle nel tentativo di dimostrare che un computer non potrà mai davvero *pensare* [Searle 1980]. L'argomento è molto famoso ma è utile riprenderlo brevemente:

> S_1) Supponiamo che Scott sia chiuso in una stanza.
>
> S_2) Supponiamo che Scott riceva dall'esterno un foglio contenente una frase in cinese. Scott non sa il cinese, ma ha a disposizione un libro di regole che associano a simboli altri simboli. Scott segue le regole e trasmette all'esterno un foglio compilato con questo sistema.
>
> S_3) Se il libro delle regole è corretto, possiamo supporre che un osservatore esterno alla stanza riceva una comunicazione in cinese corretta. Tuttavia, possiamo dire che Scott conosce il cinese?

La risposta di Searle ovviamente è *no*: nonostante Scott produca un comportamento indistinguibile da quello di un parlante nativo, non c'è nessuna comprensione del cinese. Il parallelo con il computer è ovvio: anche se un computer "parlasse" cinese correttamente, questo non implicherebbe una qualche sorta di comprensione poiché la macchina non farebbe altro che seguire delle regole, come Scott nell'esempio.

Ovviamente, non è questo il luogo per ricapitolare più di venti anni di letteratura in risposta all'argomento di Searle. Tuttavia, il ragionamento usato da Aaronson è interessante (sviluppato maggiormente in Aaronson [2012]). Supponiamo di incontrare un sistema fisico (ad esempio, un vero essere umano o un robot indistinguibile da un essere umano) che parla correttamente la nostra lingua: come facciamo a sapere che

davvero comprenda la lingua e non stia "semplicemente" seguendo un enorme libro di regole (che dicono, ad esempio, di rispondere 'Bene, tu?' a 'Come stai oggi?')? Supponiamo di stabilire che si possa mettere un limite *n* al numero di informazioni che possiamo scambiare con questo sistema fisico prima di decretare se davvero il sistema *comprende* o semplicemente stia "simulando" un comportamento intelligente. Dato che *n* è un numero finito, la tavola di regole necessarie per collegare le nostre frasi con le risposte del sistema, ancorché astronomica, deve per forza risultare finita: tuttavia – questo è il trucco – se davvero un sistema risulta "intelligente" dopo lo scambio di *n* informazioni, non potrebbe essere intelligente perché possiede un libro di regole come quello che Searle ci chiede di immaginare poiché quel libro non riuscirebbe ad essere salvato in un hard disk grande come il pianeta Terra. Questo significa che qualsiasi sistema fisico che "comprenda una lingua" in qualche modo deve avere un sistema efficiente di organizzazione dell'informazione – cioè deve essere qualcosa che definiremmo *intelligente*: se mai un computer riprodurrà la performance della Stanza Cinese di Searle, non potrà certo farlo con il metodo "stupido" che Searle ci chiede di immaginare. In altre parole, l'esperimento mentale è truccato fin dall'inizio perché nasconde magistralmente l'aspetto di complessità computazionale: Searle vuole convincerci che si possa simulare di "comprendere una lingua" attraverso una gigantesca tabella di traduzione, ma sappiamo che sarebbe impossibile costruire tale tabella – pertanto, qualsiasi conclusione possiamo trarre dall'esperimento mentale è viziata fin dall'origine da un'intuizione scorretta sulla complessità computazionale del compito richiesto.

4.2 Computazione e libero arbitrio

Lee Harvey Oswald preme il grilletto e uccide John Kennedy il 22 novembre del 1963 (perlomeno, se rimaniamo nella versione ufficiale della storia). Il problema del libero arbitrio è semplice: dato che Cesare ha passato il Rubicone secoli prima, Oswald era libero di agire altrimenti o il suo atto è stato determinato fin dai tempi di Cesare dalle leggi della fisica?

Come acutamente riconosciuto dall'autore, parte rilevante di questo dilemma metafisico è nella definizione stessa del problema: cosa significa essere liberi? La proposta di Aaronson è di affrontare la questione ponendo l'accento sull'aspetto di *prevedibilità* delle azioni di un sistema fisico *S*: se posso prevedere a t_0 l'azione (o la distribuzione di probabilità sulle azioni) di *S* a t_n, allora *S* non è libero – se vogliamo un sistema libero, dobbiamo quindi rivolgerci a qualcosa "non prevedibile". Il primo pensiero va ovviamente alla teoria della computabilità, dato che non esiste un algoritmo per decidere a t_0 se un sistema Turing universale si fermerà a t_n avendo ricevuto un certo input; ma, sottolinea Aaronson, esiste un modo teorico di prevedere il comportamento di *S*: simularlo con un altro computer più veloce. Se davvero vogliamo qualcosa di "più libero", la computazione classica non basta poiché l'informazione classica è duplicabile infinitamente.

Qui fa il suo ingresso sulla scena la computazione quantistica, che, con il teorema di *no-cloning* quantistico, afferma che sia impossibile copiare perfettamente uno stato quantistico sconosciuto *a priori*. In altre parole, mentre l'informazione classica è in linea di principio sempre duplicabile, questo non è vero per l'informazione quantistica. Se sappiamo che il libero arbitrio non può dunque nascondersi in sistemi fisici a

informazione classica (poiché essi sono per definizione prevedibili attraverso una *copia perfetta* del loro stato e la simulazione su un hardware più veloce), il teorema di *no-cloning* lascia aperta una porta: dato che un sistema fisico ad informazione quantistica non può essere così copiato, in teoria il suo comportamento potrebbe non essere prevedibile. A questo punto la domanda originale 'abbiamo libero arbitrio?' si è già trasformata in 'possiamo predire il nostro comportamento?', e infine in 'l'informazione nel nostro cervello è di natura classica (quindi copiabile/simulabile) o quantistica (quindi non copiabile)?': nessuno conosce oggi la risposta precisa a questa domanda, ma è importante sottolineare, con Aaronson, che «queste sono, in linea di principio, domande che ammettono una risposta empirica» [Aaronson 2013, p. 301].

5. Giudizio finale

QCD contiene molte discussioni interessanti che non hanno trovato spazio in questa recensione; tuttavia possiamo certamente fornire un bilancio complessivo del volume.

QCD è un libro consigliato all'informatico come al filosofo: come abbiamo avuto modo di sottolineare più volte, Aaronson è per molti aspetti uno scrittore geniale. In particolare, QCD spicca per lo stile, informale e divertente, e l'atteggiamento filosofico che pervade tutte le osservazioni scientifiche; Aaronson raggiunge i suoi migliori momenti filosofici proprio quando non menziona esplicitamente la riflessione filosofica (vedi oltre): non importa quanto un fatto scientifico possa sembrare bruto e non spiegabile, l'autore cerca sempre di connetterlo in modo originale e intuitivo ad una visione del mondo più ampia, coerente e meno arbitraria.

Dal punto di vista squisitamente filosofico, QCD ha luci e ombre: nonostante i filosofi possano certamente ricavare tantissimo materiale dalla lettura, non potranno che essere ogni tanto sorpresi da alcune discussioni di argomenti filosofici classici. A livello metodologico, QCD presenta un pattern ricorrente: l'Autore affronta una domanda antica e venerabile (ad es.: 'Abbiamo il libero arbitrio?"', *trasforma* la vaga domanda iniziale in una domanda simile, ma abbastanza precisa da ammettere una formulazione matematica (ad es.: 'Siamo un sistema fisico che in principio non è predicibile?'), *risolve* il collegato problema matematico. In alcuni casi, la trasformazione della domanda iniziale in un'altra è, non solo innocua, ma illuminante (ad es.: l'intera epistemologia o quasi potrebbe essere certamente riletta con estremo profitto utilizzando gli strumenti della complessità computazionale); in altri, la sensazione che si stia fornendo una risposta a una domanda *diversa* è piuttosto forte. Nonostante la profonda conoscenza filosofica, la sensibilità dell'autore rimane prettamente scientifica; per il lettore filosofo, alcune considerazioni risulteranno piuttosto superficiali. A tal proposito la discussione dell'argomento di Kripke sulla riduzione mente-corpo è significativa: per quanto sia scorretto, ovviamente, equiparare un *teorema* di impossibilità come quello di Gödel a un *argomento* di impossibilità come quello sulla riduzione mente-corpo, Aaronson dovrebbe cogliere il punto concettuale sollevato da Kripke (ovvero, il fatto che nonostante molte persone cerchino di ridurre una cosa all'altra, esistono motivi *a priori* per pensare che questa riduzione non sarà mai possibile) – invece la discussione rimane piuttosto scarna [Aaronson 2013, pp. 52-53].

Infine, occorre sottolineare che QCD non è un libro semplice per il "non iniziato": il lettore che non abbia familiarità con una serie di strumenti concettuali e formali potrà

trovare difficile seguire le peripezie dell'autore in alcuni punti, rischiando di perdersi il divertimento della lettura nel tentativo di ricostruire i passaggi formali necessari (la bibliografia, comodamente accessibile in ogni pagina, è sfortunatamente piuttosto povera). Certamente – come sottolineato fin dalla prefazione – esistono già tantissimi manuali di complessità computazionale, meccanica quantistica, computazione quantistica. L'intento dell'autore non è certamente manualistico: Aaronson sfrutta QCD per presentare la propria prospettiva sulle nozioni fondamentali del mondo (computazione, matematica, mondo fisico), pur non cedendo mai alla tentazione di presentare il proprio pensiero come un sistema nel senso filosofico del termine.

In definitiva, QCD è un volume di difficile classificazione, ma che certamente colpisce nel segno: il lettore con background filosofico troverà senza dubbio che lo sforzo richiesto per completare la lettura sarà ampiamente ricompensato dalle importanti nozioni che QCD magistralmente illustra (e che, purtroppo, sono relativamente poco familiari nel campo) e dalle provocanti intuizioni proposte dall'autore. Difficilmente QCD offre soluzioni definitive a noti problemi filosofici, ma certamente fornisce una quantità eccezionale di nuovi strumenti e nuove intuizioni che il filosofo di professione può importare con profitto nelle proprie ricerche.

BIBLIOGRAFIA

Aaronson S. (2013), *Quantum Computing since Democritus*, Cambridge University Press, Cambridge (MA).

Aaronson S. (2012), "Why Philosophers Should Care About Computational

Complexity", in Copeland B. J., Posy C. J., Shagir O. (a cura di), *Computability: Gödel, Turing, Church, and Beyond*, The MIT Press, Cambridge (MA).

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

Broome M. A., *et al.* (2013), "Photonic Boson Sampling in a Tunable Circuit", *Science*, 6121, pp. 794-798.

Moore C., Mertens S. (2011), *The Nature of Computation*, Oxford University Press, Oxford.

Searle J. (1980), "Minds, Brains and Programs", *Behavioral and Brain Sciences*, 3, pp. 417-57.

VandersypenL. M. K., *et al* (2001), "Experimental Realization of Shor's Quantum Factoring Algorithm Using Nuclear Magnetic Resonance", *Nature*, 414, pp. 883-887.

XuN., *et al*. (2011), "Quantum Factorization of 143 on a Dipolar-Coupling NMR system", *arXiv:1111.3726.*