

Pensiero computazionale

Rappresenta un atteggiamento e un insieme di competenze universalmente applicabili che tutti, non solo gli informatici, sarebbero desiderosi di apprendere e utilizzare.

Il pensiero computazionale si basa sulla potenza e sui limiti dei processi di elaborazione, siano essi eseguiti da un essere umano o da una macchina. I metodi e i modelli computazionali ci danno il coraggio di risolvere problemi e progettare sistemi che nessuno di noi sarebbe in grado di affrontare da solo. Il pensiero computazionale affronta l'enigma dell'intelligenza artificiale:

Cosa possono fare gli esseri umani meglio dei computer? E Cosa possono fare i computer meglio degli esseri umani? Fondamentalmente, affronta la domanda: *Cosa è computabile?* Oggi conosciamo solo una parte delle risposte a tali domande.

Il pensiero computazionale è un'abilità fondamentale per tutti, non solo per gli informatici. Alla lettura, alla scrittura e all'aritmetica, dovremmo aggiungere il pensiero computazionale alla capacità analitica di ogni bambino.

Proprio come la stampa ha facilitato la diffusione delle tre R¹, ciò che è opportunamente incestuoso in questa visione è che l'informatica e i computer facilitano la diffusione del pensiero computazionale.

Il pensiero computazionale implica la risoluzione di problemi, la progettazione di sistemi e la comprensione del comportamento umano, attingendo ai concetti fondamentali dell'informatica. Il pensiero computazionale comprende una gamma di strumenti mentali che riflettono l'ampiezza del campo dell'informatica.

Dovendo risolvere un problema specifico, potremmo chiederci: quanto è difficile risolverlo? e qual è il modo migliore per risolverlo? L'informatica si basa su solide basi teoriche per rispondere a tali domande con precisione.

Stabilire la difficoltà di un problema tiene conto della potenza di calcolo della macchina, ovvero del dispositivo di elaborazione che eseguirà la soluzione. Dobbiamo considerare il set di istruzioni della macchina, i suoi vincoli di risorse e il suo ambiente operativo.

Per risolvere un problema in modo efficiente, potremmo inoltre chiederci se una soluzione approssimativa sia sufficientemente buona, se possiamo usare la randomizzazione a nostro vantaggio e se siano ammessi falsi positivi o falsi negativi. Il pensiero computazionale consiste nel riformulare un problema apparentemente difficile in uno che sappiamo come risolvere, magari tramite riduzione, incorporamento, trasformazione o simulazione.

Il pensiero computazionale è pensare in modo ricorsivo. È elaborazione parallela. È interpretare il codice come dati e i dati come codice. È il controllo dei tipi come generalizzazione dell'analisi dimensionale. Significa riconoscere sia i vantaggi che i pericoli dell'aliasing, ovvero l'attribuzione di più

¹ Le "tre R" a cui si fa riferimento in questo contesto sono:

Reading (lettura)

Writing (scrittura)

Rithmetic (aritmetica)

Questa è una frase comune in inglese che si riferisce alle tre abilità di base dell'istruzione elementare. Le "R" in realtà non sono tre, ma la frase usa questo gioco di parole per enfatizzare l'importanza di queste competenze fondamentali. Equivalente al nostro: *leggere, scrivere e far di conto*.

nomi a qualcuno o qualcosa. Significa riconoscere sia il costo che il potere dell'indirizzamento indiretto e delle chiamate di procedura. Significa valutare un programma non solo in base alla correttezza e all'efficienza, ma anche in base all'estetica, e la progettazione di un sistema in base alla semplicità e all'eleganza.

Il pensiero computazionale utilizza l'astrazione e la decomposizione quando si affronta un compito complesso o si progetta un sistema complesso di grandi dimensioni. È separazione degli interessi. È scegliere una rappresentazione appropriata per un problema o modellarne gli aspetti rilevanti per renderlo trattabile. È usare invarianti per descrivere il comportamento di un sistema in modo succinto e dichiarativo. È avere la certezza di poter utilizzare, modificare e influenzare in modo sicuro un sistema complesso senza comprenderne ogni dettaglio. È modularizzare qualcosa in previsione di più utenti o precaricare e memorizzare nella cache in previsione di un utilizzo futuro.

Il pensiero computazionale è pensare in termini di prevenzione, protezione e ripristino dagli scenari peggiori attraverso ridondanza, contenimento dei danni e correzione degli errori. Significa chiamare uno blocco del traffico² 'deadlock'³ e i contratti 'interfacce'⁴. È imparare a evitare condizioni di gara quando si sincronizzano riunioni tra loro.

Il pensiero computazionale è usare il ragionamento euristico per trovare una soluzione. È pianificazione, apprendimento e programmazione in presenza di incertezza. È ricerca, ricerca e ancora ricerca, che si traduce in un elenco di pagine web, una strategia per vincere una partita o un controesempio. Il pensiero computazionale utilizza enormi quantità di dati per accelerare i calcoli. Fa compromessi tra tempo e spazio e tra potenza di elaborazione e capacità di archiviazione.

Considera questi esempi quotidiani: *quando tua figlia va a scuola la mattina, mette nello zaino le cose di cui ha bisogno per la giornata*; questo è prefetching e caching⁵.

Quando tuo figlio perde i suoi guanti, gli suggerisci di tornare sui suoi passi; questo è backtracking⁶.

A che punto smetti di noleggiare gli sci e te ne compri un paio?; questo è algoritmi online.

In quale fila ti trovi al supermercato?; questa è modellazione delle prestazioni per sistemi multi-server.

Perché il tuo telefono funziona ancora durante un'interruzione di corrente?; questa è indipendenza da errori e ridondanza nella progettazione. *Come fanno i test di Turing pubblici completamente*

² *Gridlock*: nel contesto del traffico, indica una situazione in cui nessuna automobile può muoversi a causa del blocco totale in un incrocio.

³ *Deadlock*: in informatica, è un termine più specifico e tecnico. Descrive una situazione in cui due o più processi si bloccano a vicenda in modo permanente, ciascuno in attesa che l'altro rilasci una risorsa.

⁴ *interfaces*: in informatica, un'interfaccia è un insieme di regole che definiscono come due componenti di un sistema interagiscono tra loro.

⁵ Il *caching* è un processo che consiste nel salvare temporaneamente una copia dei dati a cui si è avuto accesso di recente o che sono usati più frequentemente in una memoria più veloce, detta *cache*. L'obiettivo è che la prossima volta che si richiederanno quegli stessi dati, non sarà necessario recuperarli dalla loro posizione originale (che è più lenta, come un disco rigido o un server remoto), ma si potrà attingere direttamente dalla cache, riducendo significativamente i tempi di caricamento.

Il *prefetching* (pre-caricamento) è una tecnica *proattiva*. Consiste nel *prevedere* quali dati saranno necessari in futuro e caricarli nella cache *prima ancora che vengano richiesti*. Questo si basa su algoritmi che analizzano i modelli di utilizzo per fare previsioni. Se la previsione è corretta, i dati si troveranno già nella cache quando l'utente o il programma ne avrà bisogno, eliminando quasi del tutto il tempo di attesa.

⁶ Il *backtracking* è una strategia di ricerca che esplora sistematicamente l'insieme (l'*albero di ricerca*) delle soluzioni possibili, abbandonando i percorsi che non possono portare a una soluzione valida. È un approccio ricorsivo che si basa sul "provare, e in caso di fallimento, tornare indietro e provare un'altra strada".

automatizzati per distinguere computer e umani, o CAPTCHA⁷, ad autenticare gli esseri umani?; questo significa sfruttare la difficoltà di risolvere problemi di intelligenza artificiale complessi per ostacolare gli agenti informatici.

Il pensiero computazionale sarà diventato radicato nella vita di tutti quando parole come *algoritmo* e *precondizione* faranno parte del vocabolario di tutti, quando *non determinismo* e *garbage collection* assumeranno i significati usati dagli informatici; e quando gli *alberi* saranno disegnati al contrario. Abbiamo assistito all'influenza del pensiero computazionale su altre discipline. Per esempio, l'apprendimento automatico ha trasformato la statistica. L'apprendimento statistico viene utilizzato per problemi su una scala, in termini di dimensioni e taglia dei dati, inimmaginabile solo pochi anni fa. I dipartimenti di statistica di ogni tipo di organizzazione stanno assumendo informatici. Le facoltà di informatica stanno integrando dipartimenti di statistica esistenti o ne stanno creando di nuovi.

Il recente interesse degli informatici per la biologia è motivato dalla convinzione che i biologi possano trarre beneficio dal pensiero computazionale. Il contributo dell'informatica alla biologia va oltre la capacità di esaminare enormi quantità di sequenze di dati alla ricerca di pattern. La speranza è che le strutture dati e gli algoritmi – le nostre astrazioni e i nostri metodi computazionali – possano rappresentare la struttura delle proteine in modi che ne chiariscano la funzione. La biologia computazionale sta cambiando il modo in cui i biologi pensano. Allo stesso modo, la teoria computazionale dei giochi sta cambiando il modo in cui gli economisti pensano; il nanocomputing, il modo in cui i chimici pensano; e l'informatica quantistica, il modo in cui i fisici pensano.

Questo tipo di pensiero farà parte delle competenze non solo degli altri scienziati, ma di tutti gli altri. L'informatica ubiqua sta al presente come il pensiero computazionale sta al futuro. L'informatica ubiqua era il sogno di ieri che è diventato la realtà di oggi; il pensiero computazionale è la realtà di domani.

COS'È, E COSA NON È

L'informatica è lo studio della computazione - cosa può essere calcolato e come calcolarlo.

Il pensiero computazionale ha quindi le seguenti caratteristiche:

- **Concettualizzare, non programmare.** L'informatica non è programmazione informatica. Pensare come un informatico significa più che essere in grado di programmare un computer. Richiede di pensare su più livelli di astrazione;

- **Un'abilità fondamentale, non un'abilità meccanica.** Un'abilità fondamentale è qualcosa che ogni essere umano deve conoscere per funzionare nella società moderna. Meccanica significa una routine meccanica. Ironicamente, solo quando l'informatica avrà risolto la Grande Sfida dell'IA di far pensare i computer come gli umani, il pensiero sarà meccanico;

- **Un modo in cui pensano gli umani, non i computer.** Il pensiero computazionale è un modo in cui gli umani risolvono i problemi; non cerca di far pensare gli umani come i computer. I computer sono noiosi e monotoni; gli esseri umani sono intelligenti e fantasiosi. Noi esseri umani rendiamo i computer entusiasmanti. Dotati di dispositivi informatici, usiamo la nostra intelligenza per affrontare problemi che non oseremmo affrontare prima dell'era dell'informatica e costruiamo sistemi con funzionalità limitate solo dalla nostra immaginazione;

- **Integra e combina il pensiero matematico e ingegneristico.** L'informatica attinge intrinsecamente al pensiero matematico, dato che, come tutte le scienze, i suoi fondamenti formali si basano sulla matematica. L'informatica attinge intrinsecamente al pensiero ingegneristico, dato che costruiamo

⁷CAPTCHA: **C**ompletely **A**utomated **P**ublic **T**uring-test-to-tell **C**omputers and **H**umans **A**part

sistemi che interagiscono con il mondo reale. I vincoli del dispositivo informatico sottostante costringono gli informatici a pensare in modo computazionale, non solo matematico. Essere liberi di costruire mondi virtuali ci consente di progettare sistemi oltre il mondo fisico;

- Idee, non artefatti. Non saranno solo gli artefatti software e hardware che produciamo a essere fisicamente presenti ovunque e a influenzare le nostre vite in ogni momento, ma anche i concetti computazionali che useremo per affrontare e risolvere i problemi, gestire la nostra vita quotidiana e comunicare e interagire con le altre persone; e

- per tutti, ovunque. Il pensiero computazionale diventerà una realtà quando sarà così parte integrante delle attività umane da scomparire come filosofia esplicita.

Molte persone associano l'informatica alla programmazione informatica. Alcuni genitori vedono solo una ristretta gamma di opportunità di lavoro per i loro figli che si laureano in informatica. Molti pensano che la ricerca fondamentale in informatica sia completata e che rimanga solo l'ingegneria. Il pensiero computazionale è una grande visione che guida insegnanti, ricercatori e professionisti dell'informatica mentre agiamo per cambiare l'immagine che la società ha di questo settore. Dobbiamo soprattutto raggiungere il pubblico pre-universitario, inclusi insegnanti, genitori e studenti, inviando loro due messaggi principali:

- problemi scientifici intellettualmente stimolanti e coinvolgenti restano da comprendere e risolvere. Il dominio del problema e quello della soluzione sono limitati solo dalla nostra curiosità e creatività; e

- si può laureare in informatica e fare qualsiasi cosa. Si può laureare in inglese o matematica e intraprendere una moltitudine di carriere diverse. Lo stesso vale per l'informatica. Si può laureare in informatica e intraprendere una carriera in medicina, giurisprudenza, economia, politica, qualsiasi tipo di scienza o ingegneria, e persino nelle arti.

I professori di informatica dovrebbero tenere un corso chiamato "Modi per pensare come un informatico" alle matricole universitarie, e renderlo disponibile anche a chi non ha una laurea, non solo a chi si laurea in informatica. Dovremmo esporre gli studenti pre-universitari a metodi e modelli computazionali. Invece di lamentarci del calo di interesse per l'informatica o del calo dei finanziamenti per la ricerca in questo campo, dovremmo cercare di stimolare l'interesse del pubblico per l'avventura intellettuale di questo campo. Diffonderemo così la gioia, lo stupore e la potenza dell'informatica, con l'obiettivo di rendere il pensiero computazionale un luogo comune.

Jeannette M. Wing (wing@cs.cmu.edu) is the President's Professor of Computer Science in and head of the Computer Science Department at Carnegie Mellon University, Pittsburgh, PA.