

# Informatica Teorica

Manuel Pagliuca

13 ottobre 2021

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Cosa studia l'informatica ? . . . . .	2
1.2	Come l'informatica risolve i problemi? . . . . .	2
<b>2</b>	<b>Syllabus</b>	<b>3</b>
<b>3</b>	<b>Il nostro linguaggio: la matematica</b>	<b>4</b>
3.1	Funzione . . . . .	4
3.2	Classi di funzioni . . . . .	4
3.2.1	Funzioni iniettive . . . . .	4
3.2.2	Funzioni suriettive . . . . .	5
3.3	Insieme immagine di una funzione . . . . .	5
3.4	Funzioni biettive . . . . .	6
3.5	Inversa di una funzione . . . . .	6
3.6	Composizione di funzioni . . . . .	6
3.6.1	Funzione identità . . . . .	7
3.6.2	Definizione alternativa di funzione inversa . . . . .	7
3.6.3	Funzioni totali e parziali . . . . .	7
3.6.4	Totalizzazione di una funzione parziale . . . . .	8
3.6.5	Prodotto cartesiano . . . . .	8

# 1 Introduzione

Nei corsi di informatica applicata come quelli di: Sistemi Operativi, Basi di dati, ecc... l'oggetto di studio è definito dal corso, e l'informatica è lo strumento per studiare questo oggetto.

Nel corso di informatica teorica l'oggetto di studio è l'informatica stessa, si studiano i fondamenti dell'informatica (come un corso di sistemi operativi effettua uno studio sui fondamenti dei sistemi operativi).

Per eseguire lo studio di questa disciplina ci si pone le due domande fondamentali nei confronti dell'informatica *cosa* e *come*.

## 1.1 Cosa studia l'informatica ?

L'informatica è una disciplina che studia l'informazione e l'elaborazione automatica mediante sistemi di calcolo che eseguono programmi.

Ma sappiamo che tutti i problemi sono risolubili per via automatica, e questo porta proprio alla nostra domanda, quali problemi sono in grado di risolvere *automaticamente*?

Ciò che studia il *cosa* dell'informatica si chiama **teoria della calcolabilità**, mostreremo nelle successive lezioni che non è una domanda così facile da rispondere, poichè esistono delle cose che non sono calcolabili. Quindi dato che non ci sono cose calcolabili, la teoria della calcolabilità si domanda *che cosa è calcolabile?*.

Nella teoria della calcolabilità vogliamo una risposta generale, non cerchiamo dei casi particolari per dire questo è calcolabile o meno, esistono delle proprietà che accomunano tutto ciò che è calcolabile ? La risposta è sì, la nozione di calcolabilità è denotabile attraverso la matematica e quindi posso affrontare l'insieme di cose fattibili dell'informatica con gli strumenti della matematica.

## 1.2 Come l'informatica risolve i problemi?

La branca dell'informatica che si chiama **teoria della complessità** risponde alla domanda *come è risolubile questo problema ?*. Essa studia la quantità di risorse computazionali richieste dalla *soluzione automatica* dei problemi. Una *risorsa computazionale* è qualsiasi cosa venga sprecata per l'esecuzione dei programmi:

Le principali risorse su cui ci concentreremo sono il **tempo** e **spazio di memoria**, dovremo quindi dare una definizione rigorosa di queste risorse computazionali. Successivamente si potrà porre delle domande ovvie : *quale è la classe di problemi che vengono risolti efficientemente in termini di tempo e di memoria ?*. Notare che nella teoria della complessità non si parla solo di **risolubilità** (come nella teoria della calcolabilità) ma anche dell'**efficienza** con cui risolvo questo problema.

Esistono problemi che si trovano in una zona grigia, che non sappiamo se hanno una risoluzione efficiente, ma sono problemi molto importanti, nessuno è riuscito a dimostrare se avranno soluzioni efficienti ma nemmeno il contrario, ovvero nessuno è riuscito a dimostrare che saranno risolubili efficientemente.

Questa classe di problemi è la classe di problemi NP, vedremo poi di cosa si tratta.

## 2 Syllabus

- Teoria della calcolabilità: individuare la qualità della calcolabilità dei problemi, quali sono le categorie di problemi calcolabili e distinguerla da quella dei problemi non calcolabili.
- Teoria della complessità: studio quantitativo dei problemi, dopo aver delimitato i confini di ciò che è calcolabile cercare un sotto insieme di problemi **efficientemente calcolabili**.

## 3 Il nostro linguaggio: la matematica

### 3.1 Funzione

Una funzione dall'insieme  $A$  all'insieme  $B$  è una **legge**, che chiamiamo solitamente  $f$ , che spiega come associare ad ogni elemento di  $A$  un elemento di  $B$ .

Dal punto di vista formale l'espressione della funzione viene definita **globalmente**:

$$f : A \rightarrow B$$

Dove  $A$  viene chiamato **dominio** e  $B$  **codominio**, questa notazione dice che ogni elemento del dominio è associato attraverso una legge  $f$  ad un elemento del codominio. Esiste anche una notazione che permette di stabilire localmente l'operato della funzione, essa rappresenta l'operato della legge  $f$  sull'elemento  $a$  che porta all'elemento  $b$ .

$$a \xrightarrow{f} b$$

La notazione comunemente più utilizzata (in particolare nei libri di testo, ma anche nei corsi di matematica), è la seguente:

$$f(a) = b$$

Solitamente  $b$  è l'**immagine** di  $a$  secondo  $f$ , e meno usualmente si dice che  $a$  è la **controimmagine** di  $b$  (sempre secondo  $f$ ).

Per esempio:

$$f : \mathbb{N} \rightarrow \mathbb{N}$$

Dove  $\mathbb{N}$  è l'insieme dei numeri naturali  $0, 1, 2, 3, \dots$ , e per utilizzi futuri denotiamo con  $\mathbb{N}^+$  l'insieme dei numeri naturali positivi (zero escluso)  $1, 2, 3, 4, \dots$ . Ora vediamo la specifica della funzione  $f$

$$f(n) = \lfloor \sqrt{n} \rfloor$$

Considerando  $n = 5$  l'immagine di quest'ultima sarà  $f(5) = \lfloor \sqrt{5} \rfloor = 2$ .

Quindi possiamo dedurre che per più elementi del dominio una **funzione** effettua un mapping ad uno ed un solo valore del codominio (nel caso in cui un valore del dominio venga mappato a più valori del codominio allora si parla di relazione, ma non più di funzione).

### 3.2 Classi di funzioni

#### 3.2.1 Funzioni iniettive

Una funzione è **iniettiva** se e solo se elementi diversi del dominio vengono mappati in elementi diversi del codominio.

$$f : A \rightarrow B \text{ è iniettiva sse } \forall a_1, a_2 \in A \text{ dove } a_1 \neq a_2 \implies f(a_1) \neq f(a_2)$$

Esempio 1

$$f(n) = \lfloor \sqrt{n} \rfloor$$

Abbiamo visto precedentemente che questa funzione non è iniettiva, perché avviene un mapping per più elementi del dominio ad un unico elemento del codominio.

Esempio 2

$$f(n) = [n]_2$$

Questa funzione è fortemente non iniettiva, le due metà dell'insieme dei numeri naturali vengono mappate solamente su due numeri  $f(2k) = 0$  e  $f(2k+1) = 1$ .

Esempio 3

$$f(n) = n^2$$

Questa è una funzione iniettiva, poiché ad ogni controimmagine corrisponde una immagine distinta.

### 3.2.2 Funzioni suriettive

Una funzione è suriettiva quando tutti gli elementi del codominio hanno una corrispondenza con un elemento del dominio.

$$f : A \implies B \text{ sse } \forall b \in B, \exists a \in A : f(a) = b$$

Esempio 1

$f(n) = \lfloor \sqrt{n} \rfloor$  è suriettiva, questo perché  $\forall m \in \mathbb{N}, m = \lfloor \sqrt{m^2} \rfloor = f(m^2)$ . Sostanzialmente, posso tornare con facilità al dominio perché mi basta elevare al quadrato l'immagine, e questo è fattibile per tutto l'insieme dei numeri naturali.

Esempio 2

$f(n) = [n]_2$  non è una funzione suriettiva, questo perché per esempio 3 non è immagine di niente rispetto a  $f$  (il codominio è tutto).

### 3.3 Insieme immagine di una funzione

$$Im_f = \{ b \in B : \exists a, f(a) = b \} = \{ f(a) : a \in A \}$$

Data  $f$  definiamo l'**insieme immagine di  $f$**  come gli elementi del codominio  $\in B$  che sono immagine di un elemento del dominio  $A$ .

La relazione tra questo insieme  $Im_f$  ed il codominio stesso di  $f$  quale è  $B$ , consiste in:

$$Im_f \subseteq B$$

Allora possiamo dire che una funzione è suriettiva quando:

$$Im_f = B$$

*Esempi*

$$Im_{\lfloor \sqrt{n} \rfloor} = \mathbb{N} \implies f(n) = \lfloor \sqrt{n} \rfloor \text{ è suriettiva}$$

$$Im_{[n]_2} = 0, 1 \subseteq \mathbb{N} \implies f(n) = [n]_2 \text{ non è suriettiva}$$

### 3.4 Funzioni biettive

Una funzione si dice biettiva quando è sia suriettiva che iniettiva, devono valere entrambe le due condizioni (questo due condizioni è possibile fonderle in un'unica condizione).

$$f : A \rightarrow B \text{ sse}$$

$$\forall a_1, a_2 \in A, a_1 \neq a_2 \implies f(a_1) \neq f(a_2) \wedge \forall b \in B, \exists a \in A : f(a) = b$$

Che converge in un'unica definizione:

$$\forall b \in B, \exists! a \in A : f(a) = b$$

Per esempio:  $f(n) = n$ , oppure considerando gli insiemi dei numeri reali  $f(x) = x^3$ . Solo per questa tipologia di funzioni esiste il concetto di **funzione inversa**.

### 3.5 Inversa di una funzione

Data una funzione  $f$  biettiva si definisce l'inversa come  $f^{-1}$  la funzione tale che crei un mapping tra l'immagine del codominio rispetto alla controimmagine del dominio.

$$f : A \rightarrow B$$

$$f^{-1} : B \rightarrow A \text{ tale che } f^{-1}(b) = a \iff f(a) = b$$

Per esempio l'inversa di  $f(n) = n$  è  $f^{-1} = n$ , oppure l'inversa di  $f(x) = x^3$  è  $f^{-1} = \sqrt[3]{x}$  (considerando l'insieme dei numeri reali).

### 3.6 Composizione di funzioni

Date due funzioni  $f : A \rightarrow B$  e  $g : B \rightarrow C$ , notiamo che queste funzioni hanno una caratteristica in comune, ovvero che il codominio di una è il dominio dell'altra. Definiamo la composizione di funzione  $g \circ f : A \rightarrow C$  come la funzione che va dal dominio di  $f$  al codominio di  $g$ , definita come  $g(f(a))$ .

$$g \circ f = g(f(a))$$

Per esempio  $f(n) = n + 1$  e  $g(n) = n^2$ :

- $f$  composto  $g : g \circ f(n) = (n+1)^2$
- $g$  composto  $f : f \circ g(n) = n^2 + 1$

N.B. L'operazione di composizione restituisce una funzione, e l'operatore  $\circ$  non è commutativo, però quando dominio e codominio lo permettono è **associativo**:  $(f \circ g) \circ h = f \circ (g \circ h)$ .

### 3.6.1 Funzione identità

La funzione identità sull'insieme  $A$  è una funzione che effettua un mappaggio ricorsivo sullo stesso elemento.

$$i_A : A \rightarrow A : i_A(a) = a \quad \forall a \in A$$

Per esempio la funzione identità sull'insieme  $\mathbb{N}$  è  $i_{\mathbb{N}}(n) = n$ .

### 3.6.2 Definizione alternativa di funzione inversa

Data una funzione  $f : A \rightarrow B$  biettiva, la sua inversa è l'unica funzione  $f^{-1} : B \rightarrow A$  che soddisfa:

$$f^{-1}(b) = a \iff f(a) = b$$

oppure

$$f^{-1} \circ f = i_A \wedge f \circ f^{-1} = i_B$$

Infatti considerando  $f^{-1} \circ f(x) = \sqrt[3]{x^3} = x = i_{\mathbb{N}}(x)$  e  $f \circ f^{-1}(x) = (\sqrt[3]{x})^3 = x = i_{\mathbb{N}}(x)$

### 3.6.3 Funzioni totali e parziali

Considerando una funzione  $f : A \rightarrow B$  essa è una legge che ad **ogni** elemento di  $A$  si associa un elemento di  $B$ , questo significa che ogni immagine  $f(a)$  è definita per ogni elemento  $a \in A$ . Esiste un apposita notazione:

$$f(a) \downarrow \quad \forall a \in A$$

Una funzione di questo tipo viene chiamata **totale** poiché risulta definita sulla totalità del suo dominio.

Certe funzioni potrebbero *non essere definita* per quale che elemento di  $a \in A$ , e quindi non avere delle immagini corrispondenti, la notazione:

$$f(a) \uparrow$$

Ovvero, che per un elemento  $a$  non esiste immagine nell'insieme  $B$  tramite la funzione  $f$ .

Consideriamo il seguente esempio:

$$f : \mathbb{N} \rightarrow \mathbb{N}$$

$$f(n) = \lfloor \frac{1}{n} \rfloor \text{ non è definita su } n = 0 \implies f(0) \uparrow \forall n \in \mathbb{N} \setminus 0, f(n) \downarrow$$

Una funzione viene definita **parziale** se a *qualche* elemento di  $A$  si associa un elemento di  $B$ . Si amplia un nuovo concetto che è quello del **dominio** (o campo di esistenza) della funzione, ovvero quell'insieme costituito da tutti gli elementi di  $A$  tali per cui è definita una immagine appartenente a  $B$ .

$$Dom_f = \{a \in A : f(a) \downarrow\} \subseteq A$$

Allora vale precisare le due seguenti regole:

$$Dom_f \not\subseteq A \implies f \text{ parziale}$$

$$Dom_f \equiv A \implies f \text{ totale}$$

Alcuni esempi:

$$f(n) = \left\{ \frac{1}{n} + \frac{1}{(n-1)(n-2)} \implies Dom_f = \mathbb{N} \setminus \{0, 1, 2\} \right\}$$

$$f(n) = \lfloor \log n \rfloor \implies Dom_f = \mathbb{N} \setminus \{0\}$$

$$f(n) = \lfloor \sqrt{-n} \rfloor \implies Dom_f = \{0\}$$

### 3.6.4 Totalizzazione di una funzione parziale

Teniamo conto di una cosa, possiamo convenzionalmente rendere totale una funzione parziale, basta estendere il codominio con un **simbolo convenzionale**  $\perp$  che buttiamo fuori tutte le volte che la funzione non è definita.

$$f : A \rightarrow B \text{ parziale} \implies \tilde{f} : A \rightarrow B \cap \{\perp\}$$

La totalizzazione di  $f$  viene raggiunta con l'aggiunta di questo simbolo.

$$\tilde{f}(a) = \begin{cases} f(a) & \text{se } a \in Dom_f \\ \perp & \text{altrimenti} \end{cases}$$

Quindi per i punti dove il campo di esistenza non è definito verrà restituito  $\perp$ , per convenzione quando una funzione parziale viene totalizzata ovvero  $B \cap \perp$  possiamo utilizzare la seguente notazione  $B_\perp$ .

### 3.6.5 Prodotto cartesiano

$$A \times B = \{(a, b) : a \in A \wedge b \in B\}$$

Il **prodotto cartesiano** di due insiemi è l'insieme di coppie dove il primo elemento della coppia appartiene al primo insieme, ed il secondo elemento della coppia appartiene al secondo insieme.



Il prodotto cartesiano è un'operazione che non commuta.

$$A \times B \neq B \times A$$

Ovviamente l'unico caso dove un prodotto cartesiano è commutativo è quando  $A \equiv B$ . Il prodotto cartesiano può essere esteso al prodotto di ennuple di più insiemi cartesiani, dove questa volta il risultato è costituito da un insieme ordinato (non più di coppie) delle ennuple rispettive agli insiemi di provenienza:

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) : a_i \in A_i\}$$

Associato alla definizione di prodotto cartesiano abbiamo anche quella di **proiettore i-esimo**, essa è una funzione che agisce su un prodotto cartesiano. Ha come dominio l'insieme i-esimo di questo prodotto data una tupla del prodotto cartesiano non fa altro che estrapolare la componente i-esima di quella tupla (*destruttura la tupla*).

$$\pi_i : A_1 \times \dots \times A_n \rightarrow A_i$$

$$\pi_i(a_1, \dots, a_n) = a_i$$

Utilizzeremo la seguente notazione esponenziale per calcolare il prodotto cartesiano di un insieme cartesiano con se stesso:

$$A_1 \times A_2 \times A_3 \dots \times A_n = A^n$$

Alcuni esempi:

$$C = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\} \implies \text{punti che si trovano lungo la circonferenza}$$

$$I = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\} \implies \text{punti che si trovano all'interno della circonferenza}$$

$$E = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 > 1\} \implies \text{punti che si trovano all'esterno della circonferenza}$$

$$C \cap I \cap E = \mathbb{R}^2$$