# Informatica Teorica

Manuel Pagliuca

29 ottobre 2021

## Indice

# 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 cosi 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 si, 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 il 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**:

$$f : A \to B$$

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 uan notazione che permette di stabilire localmente l'operato della funzione, essa rappresenta l'operato della legge $f$ sull'elemento $a$ che porta all'elemento $b$.

$$a \xrightarrow{\text{f}} b$$

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:

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

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

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

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

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

*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 perchè $\forall m \in \mathbb{N}, m = \lfloor \sqrt{m^2} \rfloor = f(m^2)$. Sostanzialmente, posso tornare con facilità al dominio perchè 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 perchè 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 $f$ definitiamo 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).

$$f : A \to B \textbf{ sse}$$

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

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.

$$f : A \to B$$

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

Per esempio l'inversa di $f(n)$ 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 $f : A \to B$ e $g : B \to C$, 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 $g \circ f : A \to C$ come la funzione che va da 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.

$$i_A : A \to A : i_A(a) = a \; \forall a \in A$$

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

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

$$oppure$$

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

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 $f : A \to B$ 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:

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

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:

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

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

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:

$$Dom_f \nsubseteq A \implies f \text{ parziale}$$

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

Alcuni esempi:

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

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

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

### 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.

$$f : A \to B \text{ parziale} \implies \widetilde{f} : A \to B \cap \{\perp\}$$

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

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

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:

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

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*).

$$\pi_i : A_1 \times ... \times A_n \to A_i$$

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

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

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

Alcuni esempi:

$C = \{(x, y) \in \mathbb{R}^2 : x^2 + y_2 = 1\} \implies$ punti che si trovano lungo la circonferenza

$I = \{(x, y) \in \mathbb{R}^2 : x^2 + y_2 < 1\} \implies$ punti che si trovano all'interno della circonferenza

$E = \{(x, y) \in \mathbb{R}^2 : x^2 + y_2 > 1\} \implies$ punti che si trovano all'esterno della circonferenza

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

#### 3.6.6 Insieme di funzioni

L'insieme delle funzioni che vanno dall'insieme $A$ a $B$ viene indicato con:

$$B^A = \{f : A \to B\} = \text{ insieme delle funzioni da } A \text{ a } B$$

L'insieme delle funzioni *parziali* che vanno da $A$ a $B$:

$$B_\perp^A = \{f : A \to B_\perp\} = \text{ insieme delle funzioni parziali che va da A a B}$$

### 3.6.7 Funzione di valutazione

Dati due insiemi $A, B$ e $B_{\perp}^{A}$ si definisce una funzione di valutazione come:

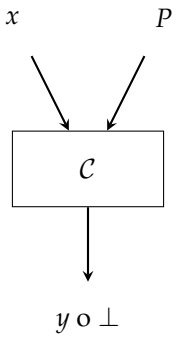
$$\omega : B_{\perp}^{A} \times A \to B \text{ con } \omega(f, a) = f(a)$$

Essa è una funzione che valuta il punto del codominio $A$ in termini di $B$, ovvero restituisce un $f(a)$, quindi il suo compito è strettamente quello di valutare.

- Tenendo fisso $a$ e facendo variare $f$, è come se $a$ fosse un *benchmark* su cui testiamo una serie di funzioni.
- Tenendo fisso $f$ e facendo variare $a$ ottengo il grafico di $f$.

## 3.7 Modellare teoricamente un sistema di calcolo

Un sistema di calcolo è architettato come un architettura di Von Neumann, è un sistema che dato in input un *dato* $x$ ed un *programma* $P$. L'output può essere indicato con $y$ quando è definito, mentre con $\perp$ quando va in *loop*.




$P \in PROG$: Un programma è una **sequenza di regole** scritte in un certo linguaggio che prende un input e lo trasforma in un output (o in un loop). Essenzialmente un programma è la definizione di una funzione scritta a mano, esso realizza una funzione che parte dai dati in input ed ottiene i dati in output.

$P \in DATI_{\perp}^{D}ATI$ è una funzione in un linguaggio di programmazione.

Quindi, ribadendo per l'ennesima volta un programma è la realizzazione di una funzione, che va da un insieme di dati ad un altro.

Il sistema di calcolo in se prende in input dei dati ed una funzione (il programma), che mi da un output, esso è definito come una **funzione di valutazione** $\mathcal{C}$.

$$\mathcal{C} : DATI_{\perp}^{DATI} \times DATI \to DATI_{\perp}$$

Quindi $\mathcal{C}$ è una funzione di valutazione, e $\mathcal{C}(P, x)$ è la funzione di valutazione del programma $P$ sul dato $x$.

Per esempio, consideriamo il seguente programma:

**Input:** $x$
**if** $\langle x \rangle_2 == 0$ **then**
  return $x$;
**else**
  **while** $1 < 2$ **do** ;
  return 1;
**end**

**Algorithm 1:** Semantica di $P$

Voglio capire la semantica di questo programma è per capirlo voglio definire una funzione che mi rappresenti il legame input-output rispetto ad un dato elemento.

Il programma prende in ingresso un numero intero, se il numero è pari restituisce esattamente il numero intero, altrimenti entra in un loop (non termina).

$$\mathcal{C}(P, \_) : \mathbb{N} \to \mathbb{N}_\perp$$

La **semantica** è la seguente :

$$\mathcal{C}(P, n) = \begin{cases} n & \text{se } n \text{ è pari} \\ \perp & \text{altrimenti} \end{cases}$$

### 3.7.1 Potenza computazionale

I sistemi di calcolo servono per calcolare le funzioni, ogni programma che immetto nel mio sistema di calcolo mi viene calcolato. Indico con potenza computazionale del mio sistema di calcolo è *tutto quello che sa fare* il mio sistema di calcolo, o meglio, tutte quelle funzioni che il mio sistema di calcolo può calcolare.

Tutte le funzioni che può calcolare il mio sistema di calcolo è un sottoinsieme di tutte le funzioni immaginabili che possono andare da dati in dati. I programmi in generale sono delle funzioni da dati in dati, quindi il mio sistema di calcolo è ovvio che calcoli questo tipo di funzioni, è un sottoinsieme perchè non so quali funzioni è in grado di risolvere il mio sistema di calcolo (e questa è la domanda a *cosa* a cui risponde l'informatica teorica).

$$F(\mathcal{C}) = \{\mathcal{C}(P, \_) : P \in PROG\} \subseteq DATI_\perp^{DATI}$$

Questo significa che esistono delle funzioni che non possono essere risolte dal mio sistema di calcolo, e che quindi **non sono automatizzabili**.

$$F(\mathcal{C}) \nsubseteq DATI_\perp^{DATI}$$

Sono presenti funzioni che l'informatica può risolvere e fare tutto.

$$F(\mathcal{C}) = DATI_{\perp}^{DATI}$$

Quindi abbiamo ridotto il quesito *che cosa sono in grado di fare i programmi*, ad un quesito matematico di inclusione propria ed impropria.

### 3.7.2 Importanza del calcolo di funzione

Calcolare una funzione significa risolvere problemi in *generale*. Questo perchè ad ogni problema posso associare una *funzione soluzione*, non è altro che quella funzione che ad un dato input associa una determinata *soluzione*. Per esempio, il *problema del calcolo del determinante*:

DET: **Input:** $M \in \mathbb{R}^{n \times n}$
**Output:** Determinante di $M$

La rispettiva funzione soluzione:

$$f_{DET} : \mathbb{N}^{n \times n} \to \mathbb{Z}$$

$$f_{DET}(M) = Det(M) :$$

Vediamo che risolvere il dato problema consiste nel calcolare la funzione soluzione, o risolvere il problema significa sostanzialmente avere un programma in grado di risolvere quella funzione.

Find-Replace: **Input:** Testo, parola da cercare, parola da sostituire
**Output:** Testo in cui ogni occorrenza della parola da cercare è sostituita dall'altra

La funzione soluzione:

$$f_{Find-Replace} : TESTI \times PAROLE \times PAROLE \to TESTI$$

$$f_{Find-Replace}(\text{La nostra vita,nostra,vostra}) = \text{La vostra vita}$$

### 3.7.3 Cardinalità degli insiemi - I°

Siamo arrivati al primo punto fondamentale, il *cosa* dell'informatica, ed abbiamo visto che si concretizza nell'interrogarsi sulla relazione tra l'insieme della potenza computazionale e quello di tutte le funzioni che vanno da dati in dati possibili.

$$F(\mathcal{C}) \ ? \ DATI_{\perp}^{DATI}$$

Per cercare di dare una dimostrazione sul carattere dell'inclusione, è molto utile fare riferimento al concetto matematica di **cardinalità**.

Dato un insieme $A$, la sua cardinalità si indica con $|A|$. Intuitivamente che la cardinalità di un insieme è il numero di elementi da cui è formato l'insieme. Questa idea intuitiva è abbastanza corretta quando si ha a che fare con insiemi finiti (la cardinalità mi può aiutare a capire quale insieme è più grande degli altri).

Purtroppo però quando tiriamo in ballo insiemi di cardinalità infinita, questo potrebbe portare a conclusioni più complicate da elaborare. Si potrebbe *erroneamente* pensare $|\mathbb{N}| = \infty = |\mathbb{B}|$, ma questo sappiamo che non è assolutamente vero, perchè l'insieme dei numeri reali è molto più fitto di quello dei numeri interi.

Quindi dobbiamo aggiungere dei nuovi concetti, visto che *"l'infinito di* $\mathbb{R}$*"* è diverso da quello di $\mathbb{N}$. Il concetto da introdurre è quello di **relazione**.

### 3.7.4 Relazione

Consideriamo un insieme $A$, si definisce **relazione binaria** $R$ su $A$, il sottoinsieme del prodotto cartesiano di $A$ su se stesso.

$$R \subseteq A^2$$

Gli elementi $a, b \in A$ stanno in una relazione $R$ se e solo se $(a, b) \in R$. Sono presenti due notazioni infisse per denotare l'esistenza della relazione tra i due elementi:

$$a \text{ R } b \text{ oppure } a \not\text{R } b$$

Consideriamo per esempio la relazione $R$ come la relazione che agisce sui numeri naturali, tale che un numero divida l'altro.

$$R \equiv \text{divide} : 3 \text{ R } 6, 5 \text{ R } 45, ..., 3 \not\text{R } 10$$

$$R = \{(a, b) \in \mathbb{N}^2 : \langle b \rangle_a = 0\}$$

Introduciamo anche il concetto di *equivalenza in modulo k*, la quale mi descrive una relazione del tipo:

$$a \equiv_k b \text{ sse } \langle a \rangle_k = \langle b \rangle_k$$

Per esempio: $5 \equiv_2 7, 4 \equiv_2 16, ...$ (quando il resto dato dal divisore $k$ è uguale su entrambi gli operandi).
Parliamo di **relazione di equivalenza** se e solo se soddisfa le seguenti proprietà:

- **Riflessiva**: $\forall a \in A, a \text{ R } a$
- **Simmetrica**: $\forall a, b, a \text{ R } b \Leftrightarrow b \text{ R } a$
- **Transitiva**: $\forall a, b, c, a \text{ R } b \wedge b \text{ R } c \implies a \text{ R } c$

Ora considerando le precedenti relazioni, vediamo che la relazione $R \equiv$ *"divide"* non è una relazione di equivalenza:

- È riflessiva.
- **Non è simmetrica**: $3 \text{ R } 6$ ma $6 \not\text{R } 3$.
- È transitiva.

Mentre per la seconda relazione $\equiv_k$ possiamo dire che essa è una relazione di equivalenza:

- È riflessiva.
- È simmetrica, vengono valutati i modulo non direttamente gli operandi.
- è transitiva (per lo stesso motivo ancora).

Con una relazione di equivalenza è possibile effettua un **partizionamento delle classi di equivalenza**.

### 3.7.5 Relazioni di equivalenza e partizioni

Considerando una relazione di equivalenza $R \subseteq A^2$ induce una **partizione** su $A$. Le partizioni sono dei sottoinsiemi $A_1, A_2, A_3, ... \subseteq A$ tali che:

- $A_i \neq \emptyset$ (non sono vuoti).
- $i \neq j \implies A_i \cap A_j = \emptyset$ (non sono sovrapponibili).
- $\bigcup_{i=1} A_i = A$ (la loro unione ricompone $A$).

Una **classe di equivalenza** di un elemento $a \in A$ è l'insieme di tutti gli elementi che sono in relazione con $a$, notazione:

$$[a]_R = \{b \in A : a \,\mathrm{R}\, b\}$$

Si dimostra facilmente che :

- Non esistono classi di equivalenza vuote, questo per via della proprietà *riflessiva* delle relazioni di equivalenza.
- Se prendo due elementi diversi del dominio le classi di equivalenza relative o sono disgiunti o sono la stessa classe di equivalenza: $a, b \in A$ vale che $[a]_R \cap [b]_R = \emptyset$ o $[a]_r = [b]_R$.
- $\bigcup_{a \in A} [a]_R = A$

Quindi la partizione indotta da una relazione di equivalenza non è altro che l'insieme delle classi di equivalenza relative a quella relazione.
L'insieme delle classi di equivalenza di $R$ è la partizione indotta da $R$ su $A$.


Figura 1: Insieme $A$ partizionato

L'insieme $A$ partizionato rispetto alla relazione $R$ è detto **insieme quoziente**:

$$A/R$$

Quindi il quoziente di $A$ rispetto a $R$ è lo *"spezzettamento"* di $A$ in classi di equivalenza.

Consideriamo il seguente esempio di insieme quoziente, consideriamo la relazione di equivalenza $\equiv_4 \subseteq \mathbb{N}$. *Quali classi di equivalenza ammette questa relazione ?*

$$[0]_4 = 4k \text{ tutti i multipli di 4 hanno lo stesso resto di 1}$$

$$[1]_4 = 4k+1 \text{ tutti i multipli di 4 aumentati di 1 hanno lo stesso resto 1}$$

$$[2]_4 = 4k+2; [3]_4 = 4k+3; ...$$

*Ne esistono altre ?* No, non esistono altre classi di equivalenza perché ogni caso successivo a quelli *"base"* si riconduce a quelli *"base"*.

$$\langle n \rangle_2 \in \{0, 1, 2, 3\}$$

Voglio vedere se queste classi di equivalenza può rappresentare una partizione:

- Nessuna classe è vuota.
- Sono mutuamente disgiunte, poiché se prendo un numero esso ricadrà solamente in una di queste classi di equivalenza.
- L'unione di queste 3 classi di equivalenza restituisce l'insieme originale $\bigcup_{i=0}^{3} [i]_4 = \mathbb{N}$.

Figura 2: Insieme quoziente di $\equiv_4$

$\{[i]_4 : 0 \leq i \leq 3\}$ è una partizione di $\mathbb{N}$ indotta dalla relazione $\equiv_4$, tale per cui mi dia il rispettivo insieme quoziente (a volte impropriamente chiamato $\mathbb{Z}_4$):

$$N/\equiv_4 \ = \{[i]_4 : 0 \leq i \leq 3\} = \mathbb{Z}_4$$

Adesso abbiamo in mano gli strumenti per definire in maniera molto più fine il concetto di cardinalità.

### 3.7.6 Cardinalità degli insiemi - II°

Sia $\mathcal{U}$ (insieme universo) la classe di tutti gli insiemi, definiamo la relazione $\sim \subseteq \mathcal{U}^2$ detta relazione di *equi numerosità* (hanno la stessa dimensione numerica), tra le coppie degli insiemi, se e solo se esiste una **biezione** tra $A$ e $B$ (ovvero, se riesco ad esibire una funzione iniettiva e suriettiva che va da $A$ in $B$).

Questa relazione tra insiemi è una relazione di equivalenza, poiché:

- $\sim$ è riflessiva, se utilizzo la funzione identità $i_A$.
- $\sim$ è simmetrica, se esiste una biezione $A \to B$ allora esiste una biezione anche $B \to A$. Ovvero $A \sim B$ e $B \sim A$.
- $\sim$ è transitiva, se compongo funzioni biettive ottengo ancora una funzione biettiva.

Due insiemi che stanno in questa relazione vengono detti *equi numerosi*. Ora considerando l'insieme quoziente del nostro universo $\mathcal{U}$ rispetto alla relazione di equi numerosità $\sim$ (quindi stesso numero di elementi in entrambi i due insiemi), esso mi rappresenta il concetto di **cardinalità** di un insieme.

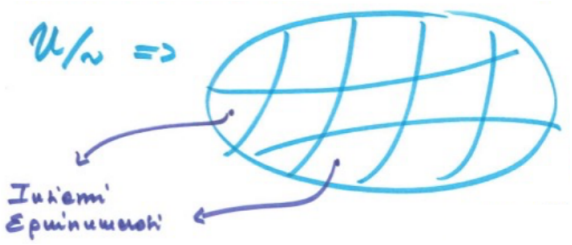



Figura 3: Insieme quoziente $\mathcal{U}/\sim$

Quindi spezzettando l'insieme $\mathcal{U}$ in classi di equivalenza in questa classe di equivalenza ci sono tutti gli insiemi che sono equi numerosi (seppur diversi).

Questo concetto permette di parlare in maniera molto precisa anche di insiemi di cardinalità infinita. Questi insiemi possono avere lo stesso numero

Per esempio, consideriamo $n \in \mathbb{N}^+$, si consideri l'insieme $J_n = \{1, 2, , ..., n\}$. Per questo insieme è ovvio quale sia il concetto di cardinalità perchè è finito.

Allora diciamo che un insieme $A$ ha cardinalità **finita** se è equi numeroso con $J_n$ (ovvero $A \sim J_n$) per un dato $n$ ed in quel caso scriviamo che $|A| = n$.

Quindi nella classe di equivalenza $J_1$ troviamo tutti gli insiemi con un elemento, nella classe di equivalenza di $J_2$ troveremo tutti gli insiemi con due elementi, ecc.

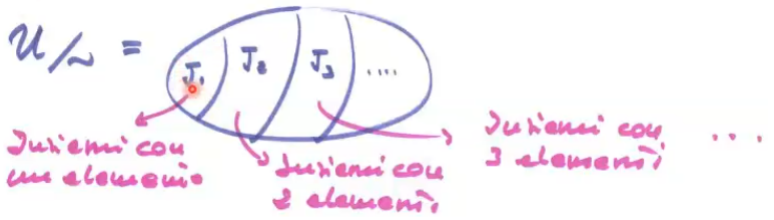



Figura 4: Esempio $\mathcal{U}/\sim$ rispetto all'insieme $J_n$

Un insieme che non ha cardinalità si dice banalmente che ha cardinalità **infinità**. Fin qui abbiamo parlato ancora di cardinalità finita, ma adesso faremo un esempio con cardinalità infinita.

### 3.7.7 Insiemi numerabili

$A$ si dice **numerabile** se è nella stessa classe di equivalenza dell'insieme dei numeri naturali $\mathbb{N}$, ovvero se esiste una *biezione* tra l'insieme $A$ e l'insieme $\mathbb{N}$.

Siccome stanno in relazione di equi numerosità vuol dire che è presente una biezione fra i due elementi, quindi due elementi non possono collidere.

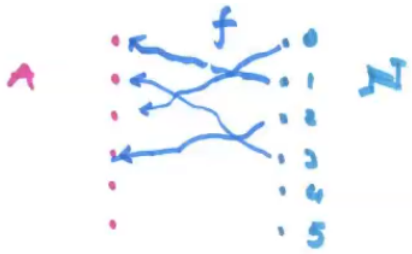



Figura 5: $A$ è un insieme numerabile

La presenza di questa biezione significa che come $A$ può essere **listato** con $f(0), f(1), f(2), ...$ su $\mathbb{N}$ senza perdere un elemento, anche l'insieme $\mathbb{N}$ può essere listato a sua volta su $A$.

Alcuni esempi di insiemi numerabili:

- I numeri pari $f(n) = 2n$ , essi sembrerebbero la metà dei numeri naturali ma sono tanti quanto i numeri naturali perchè esiste questa biezione (vale anche per i dispari $f(n) = 2n + 1$).
- L'insieme $\mathbb{Z}$, possono essere presenti diverse funzioni, per esempio posso mappare i negativi sui numeri pari ed i positivi sui numeri dispari.
- L'insieme $\mathbb{Q}$, vedremo più avanti.

- L'insieme delle stringhe binarie che incominciano con 1, ovvero $1\{0,1\}^*$, dove una funzione $f(n) = bin(n)$ è in grado di associare un numero naturale (utilizzando le potenze in posizione) ad una stringa binaria.

## 3.8 Insiemi non numerabili

Esistono degli insiemi che non sono listabili, dette in altre parole sono più fitti di $\mathbb{N}$, è un altro tipo di categoria di infinito.

$\mathbb{R}$ non è numerabile

**Dimostrazione**
L'idea consiste nel dimostra per assurdo che non esiste una biezione perchè sono presenti dei "buchi" tra le associazioni. Ordine della dimostrazione:

- Dimostro che $\mathbb{R} \sim [0,1]$ (si dice che è *isomorfo/equi numeroso* all'intervallo $[0,1]$), ovvero che è fitto quanto l'intervallo citato.
- Dimostro che $\mathbb{N} \nsim [0,1]$
- $\mathbb{R} \sim [0,1] \nsim \mathbb{N} \implies \mathbb{R} \nsim \mathbb{N}$

**Dimostrazione** $R \sim [0,1]$
Significa riuscire a rappresentare una funzione che mappa gli elementi tra i due insiemi in maniera che sia suriettiva ed iniettiva.

Prima di tutto abbiamo una retta cartesiana con un origine fissata e che copre tutti i numeri reali. Pongo l'intervallo $[0,1]$ in modo che si trovi in corrispondenza del punto mediano della retta 0, successivamente prendo un compasso punto il centro nel mediano dell'intervallo e traccio la semi circonferenza rispetto alla metà dell'intervallo.

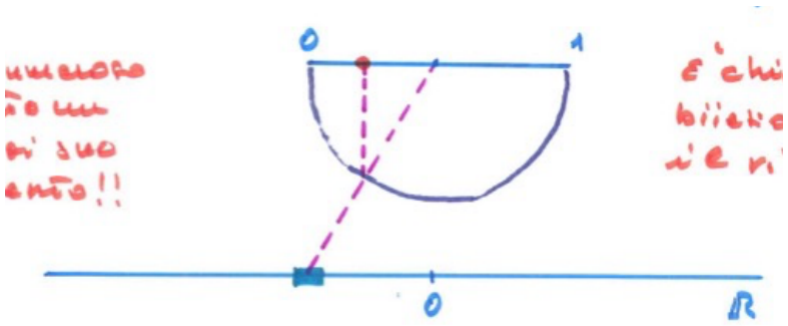



Figura 6: Dimostrazione $R \sim [0,1]$

Gli elementi dell'insieme $\mathbb{R}$ vengono mappati tracciando una retta in direzione del punto mediano di $[0,1]$, nel punto di intersezione si traccia la retta

perpendicolare che interseca l'intervallo ed in quel punto si trova il valore corrispondente. Si può fare lo stesso in maniera opposta partendo da $[0,1]$, trovando l'intersezione si parte dal punto mediano di quest'ultima e si attraversa l'intersezione toccando $\mathbb{R}$.

Questo dimostra che tutti i valori di $\mathbb{R}$ sono associabili su $[0,1]$, quindi $\mathbb{R} \sim [0,1]$.

**Dimostrazione** $\mathbb{N} \nsim [0,1]$ **(per diagonalizzazione)**
Supponiamo per assurdo che $\mathbb{N} \sim [0,1] \implies [0,1]$, ovvero che sia listabile (*elencabile*) in maniera esaustiva così come lo è $\mathbb{N}$. Siccome sto ipotizzando che sia elencabile, allora creo una lista di tutti gli elementi in $[0,1]$, i numeri all'interno di questo numero seguono il formato "0.", quindi:

$$\begin{array}{ccccc} \underline{0.a_{11}} & 0.a_{12} & 0.a_{13} & 0.a_{14} & ... \\ 0.a_{21} & \underline{0.a_{22}} & 0.a_{23} & 0.a_{24} & ... \\ 0.a_{31} & 0.a_{32} & \underline{0.a_{33}} & 0.a_{34} & ... \\ 0.a_{41} & 0.a_{42} & 0.a_{43} & \underline{0.a_{44}} & ... \\ ... & ... & ... & ... & ... \end{array}$$

Se riesco a costruire un numero che non fa parte di questa lista infinita (elusivo alla biezione), allora vuol dire che la lista non è elencabile, e quindi $[0,1]$ non è numerabile.

Per costruire questo numero elusivo che non si trova in nessuno di questi elementi della lista, vado a guardare le cifre sulla *diagonale*.
Costruiamo il numero elusivo alla lista

$$0.c_1c_2c_3c_4$$

Tale che rispetti la seguente regola rispetto alla diagonale del elenco dei numeri $\in [0,1]$

$$c_i = \begin{cases} a_{ii} + 1 \text{ se } a_{ii} < 9 \\ a_{ii} - 1 \text{ se } a_{ii} = 9 \end{cases}$$

Ora usando questa regola non riuscirò mai a posizionare il mio nuovo numero tra quelli elencati, questo perchè ovviamente cambio le cifre (il perché questo accade probabilmente è collegato al fatto che $\mathbb{R}$ è più fitto di $\mathbb{N}$, proprio quello che vogliamo dimostrare). Il numero $0.c_1c_2... \in [0,1]$, ma non appartiene nelle liste poiché:

- Differisce dal primo perché $c_1 \neq a_{11}$
- Differisce dal secondo perché $c_2 \neq a_{22}$
- Differisce da qualunque numero presente sulla diagonale

Quindi la lista non è esaustiva $\implies \mathbb{N} \nsim [0,1]$, significa che non cattura tutto l'intervallo $[0,1]$, quindi non può esistere una biezione tra questo ed $\mathbb{N}$, ovvero $[0,1]$ **non è numerabile**.

**Conclusione**

$$\mathbb{R} \sim [0,1] \nsim \implies \mathbb{R} \nsim \mathbb{N}$$

- $\mathbb{R}$ non è numerabile.
- Esso è più *"fitto"* di $\mathbb{N}$.
- Qualsiasi tentativo di listare anche solo un segmento non è esaustivo.
- $\mathbb{R}$ è un insieme **continuo**, e tutti gli insiemi equi numerosi $\mathbb{R}$ si dicono continui.
- Altri esempi di insiemi non numerabili:

### 3.8.1 Insieme delle parti di $\mathbb{N}$

Un altro insieme non numerabile è quello costituito dalla famiglia di sottoinsiemi possibili di $\mathbb{N}$, quindi mettendo assieme uno dopo l'altro tutti i sottoinsiemi di differenti cardinalità (talvolta chiamato **insieme potenza** o **booleano di** $\mathbb{N}$). Per esempio su un insieme $S = \{a, b, c\}$, l'insieme delle parti è $\mathcal{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, S\}$.

$$\mathcal{P}(\mathbb{N}) = 2^{\mathbb{N}} = \{\text{sottoinsiemi di } \mathbb{N}\} \nsim \mathbb{N}$$

Anche questa dimostrazione avviene per diagonalizzazione, per assurdo suppongo che esista una biezione che mi permetta di elencare tutti i sottoinsiemi di $\mathbb{N}$.

Posso rappresentare i sotto insiemi di $\mathbb{N}$ utilizzando un **vettore caratteristico** $A \subseteq \mathbb{N}$, questo è un vettore dove metto il bit di appartenenza rispetto alla corrispondenza dell'elemento.

$$\begin{array}{lcccccccc} \mathbb{N} \to & 0 & 1 & 2 & 3 & 4 & 5 & 6 & \ldots \\ A \to & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \ldots \end{array}$$

Allora se io suppongo per assurdo che $\mathcal{P}(\mathbb{N}) \sim \mathbb{N}$, ovvero che è numerabile rispetto all'insieme dei numeri naturali, allora posso elencare in maniera esaustiva i sotto insiemi di $\mathbb{N}$, questo elencandone i vettori caratteristici.

$$\begin{array}{ccccc} \underline{0.b_{11}} & 0.b_{12} & 0.b_{13} & 0.b_{14} & \ldots \\ 0.b_{21} & \underline{0.b_{22}} & 0.b_{23} & 0.b_{24} & \ldots \\ 0.b_{31} & 0.b_{32} & \underline{0.b_{33}} & 0.b_{34} & \ldots \\ 0.b_{41} & 0.b_{42} & 0.b_{43} & \underline{0.b_{44}} & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{array}$$

Se riesco a costruire un sottoinsieme di $\mathbb{N}$ (vettore caratteristico) che non fa parte da di questa lista, allora l'insieme delle parti non è un insieme numerabile poiché non è equi-numeroso con $\mathbb{N}$. Questo è possibile procedendo per *diagonalizzazione*, per esempio se considerassi il seguente vettore negato:

$$\begin{array}{cccc} \overline{b_{01}} & \overline{b_{12}} & \overline{b_{23}} & \overline{b_{34}} \quad \ldots \end{array}$$

riesco a constatare che esso non appartiene alla lista nonostante sia un sottoinsieme di $\mathbb{N}$ in quanto composto solo da valori booleani.

$$\mathcal{P}(\mathbb{N}) \nsim \mathbb{N}$$

### 3.8.2 Insieme $\mathbb{N}^{\mathbb{N}}$

Consideriamo l'insieme di tutte le funzioni possibili funzioni che vanno da $\mathbb{N}$ in $\mathbb{N}$.

$$\mathbb{N}^{\mathbb{N}} = \{f : \mathbb{N} \to \mathbb{N}\}$$

Ipotizzando che $\mathbb{N}^{\mathbb{N}} \sim$ , consideriamo la seguente tabella dove nella prima colonna si trova l'elenco esaustivo di tutte le funzioni $\in \mathbb{N}$, e nelle colonne successive tutto il dominio completo di $\mathbb{N}$. Significa che per ogni riga sara presente l'insieme dei valori che fornisce il grafico di una funzione $f_i$.

| Elenco delle funzioni di $\mathbb{N}$ | **0** | **1** | **2** | **3** | **...**$\in \mathbb{N}$ |
|---|---|---|---|---|---|
| $f_0$ | $\underline{f_0(0)}$ | $f_0(1)$ | $f_0(2)$ | $f_0(3)$ | ... |
| $f_1$ | $f_1(0)$ | $\underline{f_1(1)}$ | $f_1(2)$ | $f_1(3)$ | ... |
| $f_2$ | $f_2(0)$ | $f_2(1)$ | $\underline{f_2(2)}$ | $f_2(3)$ | ... |
| $f_3$ | $f_3(0)$ | $f_3(1)$ | $f_3(2)$ | $\underline{f_3(3)}$ | ... |
| ... | ... | ... | ... | ... | $\underline{...}$ |

Allora anche in questo caso per diagonalizzazione voglio costruire una funzione $\in \mathbb{N}^{\mathbb{N}}$ ma elusiva all'elenco delle funzioni della tabella:

$$\varphi : \mathbb{N} \to \mathbb{N} \text{ come } \varphi(n) = f_n(n) + 1$$

Siamo d'accordo che $\varphi$ sia ben definita, visto che per ogni immagine corrisponderà un immagine numero naturale, ma notiamo che una qualsiasi $\varphi(n)$ presa sulla lista avrà per forza una immagine della diagonale discordante rispetto a quelle elencate nella tabella.
Quindi abbiamo una funzione elusiva all'elenco, e la nostre ipotesi si rivela assurda:

$$\varphi(0) = f_0(0) + 1 \neq f_0(0)$$

$$\mathbb{N}^{\mathbb{N}} \nsim \mathbb{N}$$

N.B.: Gli insiemi $\mathcal{P}(\mathbb{N})$ e $\mathbb{N}^{\mathbb{N}}$ sono detti **insiemi continui**, e sono equinumerosi a $\mathbb{R}$.

## 3.9 Cosa è calcolabile?

Sappiamo che la potenza computazionale di un sistema di calcolo $\mathcal{C}$ corrisponde a:

$$F(\mathcal{C}) = \{\mathcal{C}(P, \_) : P \in PROG\} \subseteq DATI_{\perp}^{DATI}$$

Il nostro problema consiste nel comprendere se questa inclusione sia propria o impropria, per dimostrare ciò posso utilizzare il concetto di cardinalità precedentemente appreso.
Se riesco a dimostrare che il primo insieme ha una cardinalità numerabile ed il secondo ha una cardinalità continua allora il secondo insieme sarà molto più grande del primo.
Prendiamo alcune **assunzioni** che verranno dimostrate successivamente nel corso:

- $PROG \sim \mathbb{N}$, i programmi sono tanti quanti i numeri naturali, questo perché un programma viene digitalizzato in una fila di bit (finiti). Questo vuol dire che per ogni programma corrisponderà un numero che fa parte dei numeri naturali.
- $DATI \sim \mathbb{N}$, i dati sono tanti quanto i numeri, questo per lo stesso motivo di digitalizzazione (o traduzione) in binario delle informazioni.

Possiamo dire che la potenza computazionale sia equinumerosa rispetto al numero di programmi, al variare del programma cambio la funzione di potenza computazionale. A questo punto agganciamo la prima assunzione.

$$F(\mathcal{C}) \sim PROG \sim \mathbb{N}$$

Sapendo che l'insieme dei dati in dati definito come $DATI_{\perp}^{DATI}$, grazie alla seconda assunzione che abbiamo fatto allora possiamo asserire:

$$DATI_{\perp}^{DATI} \sim \mathbb{N}_{\perp}^{\mathbb{N}}$$

Abbiamo visto precedentemente che $\mathbb{N}^{\mathbb{N}} \nsim \mathbb{N}$, allora unendo le precedenti dimostrazioni possiamo confermare:

$$F(\mathcal{C}) \sim PROG \sim \mathbb{N} \nsim \mathbb{N}^{\mathbb{N}} \sim DATI_{\perp}^{DATI}$$

$$F(\mathcal{C}) \nsim DATI_{\perp}^{DATI}$$

Ovvero, che l'insieme dei programmi (corrispondente all'insieme delle funzioni di potenza di calcolo) non è equinumeroso (o *isomorfo*) all'insieme delle funzioni che vanno da dati in dati. In parole povere ho un numero di programmi nettamente inferiore (fitto quanto i numeri naturali) rispetto al numero di funzioni (o problemi) calcolabili (fitto quanto i reali $\mathbb{N}_{\perp}^{\mathbb{N}}$).