

Informatica Teorica

Mauro Tellaroli

Indice

0	Introduzione	2
1	Prerequisiti matematici	3
2	Teoria della calcolabilità	5
2.1	Sistema di calcolo \mathcal{C}	5
2.2	Potenza computazionale di \mathcal{C}	5
2.3	Cardinalità di insiemi infiniti	5
2.3.1	Relazione binaria	6
2.3.2	Relazione di equivalenza	6
2.3.3	Classe di equivalenza	6
2.3.4	Insiemi isomorfi	6
2.3.5	Insiemi numerabili	7
2.3.6	Insiemi non numerabili	7
2.4	Cosa è calcolabile?	8

0 Introduzione

L'informatica è la disciplina che studia l'informazione e la sua elaborazione **automatica**. L'elaborazione in questione non è legata a nessun mezzo, si tratta quindi di una qualsiasi elaborazione che può avvenire con o senza un computer.

Obiettivo di questo corso è rispondere a due domande:

1. Cosa è calcolabile automaticamente? → Teoria della calcolabilità
2. Quanto “costa” risolvere un problema? → Teoria della complessità

1 Prerequisiti matematici

Classi di funzioni $f : A \rightarrow B$

Iniettive

f è iniettiva se $\forall a_1, a_2 \in A : a_1 \neq a_2 \Rightarrow f(a_1) \neq f(a_2)$

Suriettive

f è suriettiva se $\forall b \in B \exists a \in A : f(a) = b$

Biettive

f è biettiva se è sia iniettiva che suriettiva.

Composizione di funzioni

Date $f : A \rightarrow B$ e $g : B \rightarrow C$, si definisce f composto g come la funzione $g \circ f : A \rightarrow C$ come:

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

La composizione non è un operatore commutativo.

Funzioni parziali e totali

La notazione $f(a) \downarrow$ indica che la funzione è definita su a , ovvero che esiste un valore b del codominio tale che $f(a) = b$.

Al contrario, la notazione $f(a) \uparrow$ indica che la funzione **non** è definita su a .

Una funzione $f : A \rightarrow B$ definita su tutto il suo dominio è detta totale. Se invece esistono dei valori del dominio nei quali f non è definita, f è detta parziale:

$$f \text{ è } \mathbf{totale} \text{ se } \forall a \in A \quad f(a) \downarrow$$

$$f \text{ è } \mathbf{parziale} \text{ se } \exists a \in A : f(a) \uparrow$$

Campo di esistenza

Dalla definizione di funzione parziale si intuisce come l'insieme di tutti i valori nel quale la funzione $f : A \rightarrow B$ è definita, non sempre coincide con il dominio A . Questo insieme è detto **campo di esistenza di f** e si denota con Dom_f :

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

Totalizzazione di una funzione parziale

Presa una funzione $f : A \rightarrow B$ parziale, la si può totalizzare, ovvero rendere totale, aggiungendo al codominio un valore \perp che rappresenta il caso indefinito:

$$f : A \rightarrow B \xrightarrow{\text{totalizzazione}} f : A \rightarrow B \cup \{\perp\}$$

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

L'insieme $B \cup \{\perp\}$ viene abbreviato con B_\perp .

Prodotto cartesiano

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

L'operatore \times non gode della proprietà commutativa.

$$\underbrace{A \times A \times \cdots \times A}_{n \text{ volte}} = A^n$$

Insiemi di funzioni

Tutte le funzioni che vanno da A a B è detto B^A :

$$B^A = \{f : A \rightarrow B\}$$

$$B_{\perp}^A = \{f : A \rightarrow B_{\perp}\}$$

Funzione di valutazione

Si definisce funzione di valutazione $w : B_{\perp}^A \times A \rightarrow B$ con:

$$w(f, a) = f(a)$$

- Fissando a provo tutte le funzioni su a ;
- Fissando f ottengo il suo grafico.

2 Teoria della calcolabilità

2.1 Sistema di calcolo \mathcal{C}

Si vuole modellare matematicamente un calcolatore o sistema di calcolo \mathcal{C} :

$$\begin{array}{l} x \in \text{DATI} \longrightarrow \\ P \in \text{PROG} \longrightarrow \end{array} \boxed{\mathcal{C}} \longrightarrow y / \perp$$

La figura mostra il sistema di calcolo \mathcal{C} che, preso un programma P su input x , restituisce in output il risultato y o il valore \perp se il programma va in loop.

DATI è l'insieme di tutti i possibili dati di input e PROG l'insieme di tutti i possibili programmi.

Il sistema di calcolo \mathcal{C} non fa altro che eseguire il programma P su input x ricavandone il risultato y :

$$\mathcal{C} : \text{PROG} \times \text{DATI} \rightarrow \text{DATI}_{\perp} \quad (1)$$

Quello che fa il programma P è trasformare il dato di input x in un dato di output y ; si può quindi dire che un programma non è altro che una funzione che agisce da DATI in DATI:

$$\begin{array}{c} P : \text{DATI} \rightarrow \text{DATI}_{\perp} \\ \Downarrow \\ \text{PROG} = \text{DATI}_{\perp}^{\text{DATI}} \end{array} \quad (2)$$

La funzione associata al programma P è detta **semantica di P** .

Da (1) e (2) si ottiene che:

$$\mathcal{C} : \text{DATI}_{\perp}^{\text{DATI}} \times \text{DATI} \rightarrow \text{DATI}_{\perp}$$

\mathcal{C} è una funzione di valutazione; $\mathcal{C}(P, x)$ è infatti la semantica di P .

2.2 Potenza computazionale di \mathcal{C}

Si definisce potenza computazionale di \mathcal{C} :

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

$F(\mathcal{C})$ **contiene tutto ciò che un qualsiasi sistema di calcolo \mathcal{C} può calcolare**. Quindi, per stabilire cosa l'informatica può risolvere, basta stabilire il carattere dell'inclusione:

- $F(\mathcal{C}) \subset \text{DATI}_{\perp}^{\text{DATI}} \Rightarrow$ esistono problemi che l'informatica non può risolvere;
- $F(\mathcal{C}) = \text{DATI}_{\perp}^{\text{DATI}} \Rightarrow$ l'informatica può risolvere tutto.

2.3 Cardinalità di insiemi infiniti

Per riuscire a capire se l'inclusione $F(\mathcal{C}) \subseteq \text{DATI}_{\perp}^{\text{DATI}}$ sia propria o meno, si confronterà la cardinalità dei due insiemi. Infatti dalla cardinalità si può ricavare che:

- Se $|F(\mathcal{C})| < |\text{DATI}_{\perp}^{\text{DATI}}| \Rightarrow F(\mathcal{C}) \subset \text{DATI}_{\perp}^{\text{DATI}};$
- Se $|F(\mathcal{C})| = |\text{DATI}_{\perp}^{\text{DATI}}| \Rightarrow F(\mathcal{C}) = \text{DATI}_{\perp}^{\text{DATI}}.$

Il concetto di cardinalità è semplice quando si tratta di insiemi finiti: basta contare il numero di elementi che compongono l'insieme. Tuttavia, in presenza di insiemi infiniti le cose si complicano.

Per esempio, si confrontino \mathbb{N} e \mathbb{R} : entrambi hanno cardinalità infinita ($|\mathbb{N}| = |\mathbb{R}| = \infty$) eppure $\mathbb{N} \subset \mathbb{R}$! Per comprendere quindi meglio la cardinalità di insiemi infiniti si dovrà andare più nel dettaglio.

2.3.1 Relazione binaria

Si definisce relazione binaria R sull'insieme A , un elenco di coppie ordinate di elementi di A : $R \subseteq A^2$. Due elementi $a, b \in A$ sono in relazione R se $(a, b) \in R$. Si usa la notazione:

- $a R b$: a è in relazione R con b ;
- $a \not R b$: a non è in relazione R con b ;

2.3.2 Relazione di equivalenza

$R \subseteq A^2$ è una relazione di equivalenza se gode di:

1. Riflessività: $\forall a \in A \quad a R a$
2. Simmetria: $\forall a, b \in A \quad a R b \Leftrightarrow b R a$
3. Transitività: $\forall a, b, c \in A \quad a R b \wedge b R c \Rightarrow a R c$

2.3.3 Classe di equivalenza

Si definisce classe di equivalenza $[a]_R$ l'insieme degli elementi in relazione R con a :

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

Tutte le classi di equivalenza di R formano una partizione di A . L'insieme A partizionato attraverso le classi di equivalenza di R è detto **quoziente** di A rispetto a R ed è denotato da A/R .

Esempio

Si consideri la relazione $\equiv_4 \subseteq \mathbb{N}^2$ di equivalenza modulo 4. Due numeri sono in relazione di equivalenza modulo 4 se il resto della divisione per 4 è uguale per entrambi.

$$5 \equiv_4 9, \quad 10 \equiv_4 2, \quad \dots$$

Le classi di equivalenza sono:

$$\begin{aligned} [0]_4 &= \{4k\} && \text{(Multipli di 4)} \\ [1]_4 &= \{4k+1\} && \text{(Resto 1)} \\ [2]_4 &= \{4k+2\} && \text{(Resto 2)} \\ [3]_4 &= \{4k+3\} && \text{(Resto 3)} \end{aligned}$$

L'insieme $\{[0]_4, [1]_4, [2]_4, [3]_4\} = \mathbb{N}/\equiv_4$ è una partizione di \mathbb{N} .

2.3.4 Insiemi isomorfi

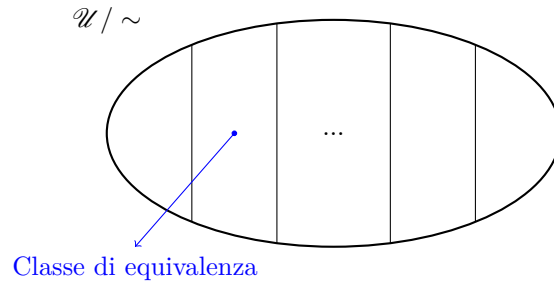
Due insiemi A e B sono **isomorfi** (o equinumerosi) se esiste una funzione biettiva tra essi. Formalmente si indica con:

$$A \sim B$$

La relazione di isomorfismo \sim è una relazione di equivalenza in quanto:

1. Riflessiva: si usi la funzione identità;
2. Simmetrica: se esiste una funzione biettiva allora anche la sua inversa è biettiva;
3. Transitiva: la composizione di due funzioni biettive è una funzione biettiva.

Sia \mathcal{U} l'insieme universo, ovvero l'insieme che contiene tutti gli insiemi. Il quoziente di \mathcal{U} rispetto a \sim (\mathcal{U} / \sim) definisce il concetto di cardinalità:



Ogni partizione di \mathcal{U} / \sim contiene gli insiemi tra loro isomorfi, ovvero che hanno la stessa cardinalità.

Insiemi finiti

Si definisca la famiglia di insiemi:

$$J_n = \begin{cases} \emptyset & n = 0 \\ \{1, \dots, n\} & n > 0 \end{cases}$$

$$J_0 = \{\} , J_1 = \{1\} , J_2 = \{1, 2\} , J_3 = \{1, 2, 3\} , \dots$$

Un'insieme A ha cardinalità finita se $\exists n \in \mathbb{N} : A \sim J_n$ e si può dire che $|A| = n$.

Insiemi infiniti

Un insieme che non è finito ha cardinalità infinita.

2.3.5 Insiemi numerabili

Un insieme A è numerabile se $\mathbb{N} \sim A$ (ovvero $A \in [\mathbb{N}]_{\sim}$). Vuole quindi dire che esiste una biezione $f : \mathbb{N} \rightarrow A$ che permette di listare A come:

$$A = \{f(0), f(1), f(2), \dots\}$$

senza tralasciare nessun elemento.

Esempi

PARI : $f(n) = 2n$
 DISPARI : $f(n) = 2n + 1$
 \mathbb{Z} : mappo i pari nei non-negativi e i dispari nei negativi
 $\{0\} \cup 1\{0, 1\}^*$: converto da binario a decimale

2.3.6 Insiemi non numerabili

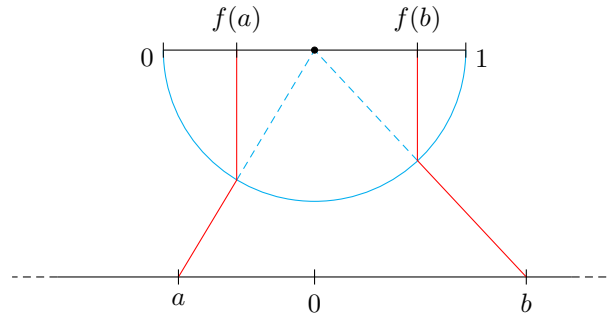
Gli insiemi non numerabili sono insiemi a cardinalità infinita ma non listabili come \mathbb{N} (sono "più fitti"). Il re di questi insiemi è \mathbb{R} .

Teorema 1. \mathbb{R} è un insieme non numerabile:

$$\mathbb{N} \not\sim \mathbb{R}$$

Dimostrazione. Per dimostrarlo dimostro che:

1. $\mathbb{R} \sim (0, 1)$: la biezione è rappresentata graficamente in figura:



(In realtà \mathbb{R} è isomorfo a un suo qualsiasi intervallo).

2. $\mathbb{N} \approx (0, 1)$: dimostrazione per assurdo: assumo che $\mathbb{N} \sim (0, 1)$; Questo vorrebbe dire che tutti i numeri compresi tra 0 e 1 sono numerabili. Elenco tutti i numeri associandoli a un numero naturale:

$0 \mapsto 0.\textcolor{red}{a}_{00} a_{01} a_{02} a_{03} a_{04} \dots$	a_{ij} è la i -esima cifra dopo lo zero del j -esimo numero nella lista.
$1 \mapsto 0.a_{10} \textcolor{red}{a}_{11} a_{12} a_{13} a_{14} \dots$	Se $(0, 1)$ fosse numerabile tutti i suoi numeri dovrebbero far parte della lista.
$2 \mapsto 0.a_{20} a_{21} \textcolor{red}{a}_{22} a_{23} a_{24} \dots$	Si consideri il numero:
$3 \mapsto 0.a_{30} a_{31} a_{32} \textcolor{red}{a}_{33} a_{34} \dots$	
$4 \mapsto 0.a_{40} a_{41} a_{42} a_{43} \textcolor{red}{a}_{44} \dots$	$0.c_0 c_1 c_2 c_3 \dots$
$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots$	con:

$$c_i = \begin{cases} 2 & a_{ii} \neq 2 \\ 3 & a_{ii} = 2 \end{cases}$$

Chiaramente $0.c_0 c_1 c_2 c_3 \dots \in (0, 1)$ ma non appare nella lista:

- Differisce dal primo numero perchè $c_0 \neq a_{00}$;
- Differisce dal secondo numero perchè $c_1 \neq a_{11}$;
- ...
- Differisce da qualunque numero nella lista sulla cifra **diagonale**.

Ho trovato l'assurdo quindi $\mathbb{N} \approx (0, 1)$ (dimostrazione per diagonalizzazione).

Sfruttando la transitività di \sim posso si può affermare quindi che:

$$\mathbb{R} \underset{(1)}{\sim} (0, 1) \underset{(2)}{\approx} \mathbb{N} \Rightarrow \mathbb{R} \approx \mathbb{N}$$

□

Tutti gli insiemi isomorfi a \mathbb{R} sono detti continui. Altri insiemi non numerabili sono:

- Insieme delle parti di \mathbb{N} : $2^{\mathbb{N}} = \{\text{sottoinsiemi di } \mathbb{N}\}$
- Insieme delle funzioni da \mathbb{N} a \mathbb{N} : $\mathbb{N}_{\perp}^{\mathbb{N}} = \{f : \mathbb{N} \rightarrow \mathbb{N}_{\perp}\}$

2.4 Cosa è calcolabile?

Ora che il concetto di cardinalità è più chiaro, si riprenda il concetto di potenza computazionale di un sistema di calcolo \mathcal{C} (paragrafo 2.2):

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

Per definizione $F(\mathcal{C})$ ha la stessa numerosità di PROG:

$$F(\mathcal{C}) \sim \text{PROG}$$

Ragionevolmente, **ma non formalmente**, si può notare che:

- $\text{PROG} \sim \mathbb{N}$: si prenda la stringa binaria con la quale il programma è salvato sul disco e si converta da binario a decimale;
- $\text{DATI} \sim \mathbb{N}$: si applichi lo stesso ragionamento del punto precedente.

Ne segue che:

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