Informatica teorica

Indice

1. Introduzione **2**
1.1. Definizione 2
1.2. Cosa e come 2
2. Richiami matematici **3**
2.1. Funzioni 3

1. Introduzione

1.1. Definizione

L'**informatica teorica** è quella branca dell'informatica che si "contrappone" all'informatica applicata: in quest'ultima, l'informatica è usata solo come uno *strumento* per gestire l'oggetto del discorso, mentre nella prima l'informatica diventa l'*oggetto* del discorso, di cui ne vengono studiati i fondamenti.

1.2. Cosa e come

Analizziamo i due aspetti fondamentali che caratteristicano ogni materia:

1. il **cosa**: l'informatica è la scienza che studia l'informazione e la sua elaborazione automatica mediante un sistema di calcolo. Ogni volta che ho un *problema* cerco di risolverlo automaticamente scrivendo un programma. *Posso farlo per ogni problema? Esistono problemi che non sono risolubili?* Possiamo chiamare questo primo aspetto con il nome di **teoria della calcolabilità**, quella branca che studia cosa è calcolabile e cosa non lo è, a prescindere dal costo in termini di risorse che ne deriva. In questa parte utilizzeremo una caratterizzazione molto rigorosa e una definizione di problema il più generale possibile, così che l'analisi non dipenda da fattori tecnologici, storici…
2. il **come**: è relazionato alla **teoria della complessità**, quella branca che studia la quantità di risorse computazionali richieste nella soluzione automatica di un problema. Con *risorsa computazionale* si intende qualsiasi cosa venga consumato durante l'esecuzione di un programma, ad esempio:
   - elettricità;
   - numero di processori;
   - tempo;
   - spazio di memoria.

   In questa parte daremo una definizione rigorosa di tempo, spazio e di problema efficientemente risolubile in tempo e spazio, oltre che uno sguardo al famoso problema `P = NP`.

Possiamo dire che il *cosa* è uno studio **qualitativo**, mentre il *come* è uno studio **quantitativo**.

Grazie alla teoria della calcolabilità individueremo le funzioni calcolabili, di cui studieremo la complessità.

2. Richiami matematici

2.1. Funzioni

Una **funzione** da un insieme A ad un insieme B è una *legge*, spesso indicata con f, che spiega come associare agli elementi di A un elemento di B.

Abbiamo due tipi di funzioni:
- **generale**: la funzione è definita in modo generale come $f : A \longrightarrow B$, in cui A è detto **dominio** di f e B è detto **codominio** di f;
- **locale/puntuale**: la funzione riguarda i singoli valori a e b:

  $$f(a) = b \quad | \quad a \overset{f}{\longmapsto} b$$

  in cui b è detta **immagine** di a rispetto ad f e a è detta **controimmagine** di b rispetto ad f.

Possiamo categorizzare le funzioni in base ad alcune proprietà:
- **iniettività**: una funzione $f : A \longrightarrow B$ si dice *iniettiva* se e solo se:

  $$\forall a_1, a_2 \in A \quad a_1 \neq a_2 \Longrightarrow f(a_1) \neq f(a_2)$$

  In poche parole, non ho *confluenze*, ovvero *elementi diversi finiscono in elementi diversi.*
- **suriettività**: una funzione $f : A \longrightarrow B$ si dice *suriettiva* se e solo se:

  $$\forall b \in B \quad \exists a \in A \ | \ f(a) = b.$$

  In poche parole, *ogni elemento del codominio ha almeno una controimmagine.*

Se definiamo l'**insieme immagine**:

$$\mathrm{Im}_f = \{b \in B \ | \ \exists a \in A \text{ tale che } f(a) = b\} = \{f(a) \ | \ a \in A\} \subseteq B$$

possiamo dare una definizione alternativa di funzione suriettiva, in particolare una funzione è *suriettiva* se e solo se $\mathrm{Im}_f = B$.

Infine, una funzione $f : A \longrightarrow B$ si dice **biiettiva** se e solo se è iniettiva e suriettiva, quindi vale: $\forall b \in B \quad \exists! a \in A \ | \ f(a) = b$.

Se $f : A \longrightarrow B$ è una funzione biiettiva, si definisce **inversa** di f la funzione $f^{-1} : B \longrightarrow A$ tale che:

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

Per definire la funzione inversa f^{-1}, la funzione f deve essere biiettiva: se così non fosse, la sua inversa avrebbe problemi di definizione.

Un'operazione definita su funzioni è la **composizione**: date $f : A \longrightarrow B$ e $g : B \longrightarrow C$, la funzione f *composto* g è la funzione $g \circ f : A \longrightarrow C$ definita come $(g \circ f)(a) = g(f(a))$.

La composizione *non è commutativa*, quindi $g \circ f \neq f \circ g$ in generale, ma è *associativa*, quindi $(f \circ g) \circ h = f \circ (g \circ h)$.

La composizione f *composto* g la possiamo leggere come *prima agisce* f *poi agisce* g.

Dato l'insieme A, la **funzione identità** su A è la funzione $i_A : A \longrightarrow A$ tale che:

$$i_A(a) = a \quad \forall a \in A,$$

ovvero è una funzione che mappa ogni elemento su se stesso.

Grazie alla funzione identità, possiamo dare una definizione alternativa di funzione inversa: data la funzione $f : A \longrightarrow B$ biiettiva, la sua inversa è l'unica funzione $f^{-1} : B \longrightarrow A$ che soddisfa:

$$f \circ f^{-1} = f^{-1} \circ f = \mathrm{id}_A \,.$$

Possiamo vedere f^{-1} come l'unica funzione che ci permette di *tornare indietro*.