Riferimenti al libro di testo:

- Capitolo 2: 2.7
- Capitolo 3
- Capitolo 4: 4.1

Indice

1 Introduzione alla materia

Prima di iniziare con la presentazione di alcuni concetti fondamentali, si definisce l'**invariante induttiva**: pensando a qualsiasi linguaggio di programmazione, una generica condizione è sempre vera prima, durante e dopo un ciclo. Questo concetto ritornerà in futuro.

1.1 Cardinalità degli insiemi

Il motivo dell'interesse di un ripasso di un argomento trattato in passato è giustificato dal fatto che i dati manipolati in informatica sono (e)numerabili, ovvero è possibile metterli in corrispondenza biunivoca con i numeri naturali, essendo essi stessi rappresentati da numeri (binari).

Di seguito viene mostrato un richiamo ai concetti di base in relazione alla cardinalità di insiemi:

- ✰ **Cardinalità.** Se S è un insieme, la sua ***cardinalità*** si rappresenta con il simbolo $|S|$.

- ✰ **Equipotenza.** Due insiemi A e B sono ***equipotenti*** se esiste una funzione biiettiva del tipo $f : A \to B$ (cioè una funzione sia iniettiva che suriettiva; approfondimento: link, oppure qui di seguito).
  La ***rappresentazione*** matematica è la seguente $A \approx B$.
  La relazione $|A| \leq |B|$ è possibile se esiste una funzione iniettiva $f : A \to B$. Si osservi che la funzione f stabilisce una corrispondenza tra gli elementi dei due insiemi. Infatti, l'**iniettività** assicura che la corrispondenza è stabilita elemento per elemento, mentre la **surriettività** assicura che la quantità degli oggetti nei due insiemi coincide.

- ✰ **Insiemi finiti e infiniti.** Negli ***insiemi finiti***, la cardinalità è un numero naturale corrispondente al numero di oggetti contenuti nell'insieme. Invece, negli ***insiemi infiniti*** la $|A|$ rappresenta la collezione degli insiemi Y tale che $Y \approx A$. Questa collezione viene chiamata **cardinalità** di A. Quindi, è vero che se $A \subseteq B$ allora si deduce che $|A| \leq |B|$.

- ✰ **Insieme numerabile.** Un insieme A viene detto ***numerabile*** se è finito o equipotente all'insieme dei numeri naturali $\mathbb{N}$ (ovvero, $A \approx \mathbb{N}$).
  La cardinalità degli insiemi infiniti numerabili è denotata con $\mathfrak{N}_0$.
  Un insieme A è finito se $|A| < \mathfrak{N}_0$. Quindi, un insieme è numerabile se $|A| \leq \mathfrak{N}_0$ (ovvero se è finito, quindi minore, oppure se è un insieme infinito numerabile rappresentato come $\mathfrak{N}_0$, quindi uguale).

1.2 Alcune notazioni

Se un generico ***linguaggio di programmazione*** viene indicato con la lettera $\mathfrak{L}$ e un generico ***algoritmo*** di un programma viene indicato con la lettera A, allora se un *algoritmo viene implementato in un linguaggio di programmazione*, è possibile scrivere la notazione insiemistica $A \in \mathfrak{L}$. In un linguaggio di programmazione è possibile scrivere infiniti programmi, ovvero l'insieme dei numeri naturali $\mathbb{N}$.
Esistono due tipi di ***rappresentazioni***:

- **Rappresentazione intensionale.** Rappresenta solo l'algoritmo, più nello specifico solamente quella specifica parte di codice (esempio a fine elenco).
- **Rappresentazione estensionale.** Rappresenta l'insieme ma tramite una forma più estesa (esempio a fine elenco).

L'***esecuzione*** di un determinato algoritmo si indica con delle parentesi quadre più spesse $[\![A]\!]$. Quindi, la sua *rappresentazione intensionale* è solamente A, mentre la sua *rappresentazione estensionale* è data da $[\![A]\!](i) = o$ (i è input e o è output). La rappresentazione estensionale può essere anche nel seguente modo $[\![M]\!] \in \{f \mid f = [\![M]\!]\}$ con $f = \{(x, f(x)) \mid x \in \mathbb{N}\}$.

Un programma restituisce uno o più risultati come numeri naturali $\mathbb{N}$, prendendo in input dei numeri naturali $\mathbb{N}$. Quindi, più formalmente si può scrivere $\mathbb{N} \longrightarrow \mathbb{N}$. Questa rappresentazione non è altro che la definizione dei ***problemi*** esistenti. Difatti, l'informatica si pone il dubbio che esista una certa soluzione (f), scritta sotto forma di algoritmo appartenente ad un linguaggio di programmazione, tale che la sua esecuzione dia la soluzione. Più formalmente:

$$\mathbb{N} \longrightarrow \mathbb{N} \ni f \qquad \exists A \in \mathfrak{L} : [\![A]\!] = f$$

1.3 Teorema di Cantor

Il seguente teorema ha come conseguenza che **esistono insiemi non numerabili**. Questo risultato si attribuisce a Georg Cantor, matematico tedesco, nel 1874.

La **dimostrazione** è importante da capire. Essa utilizza una tecnica, detta dimostrazione diagonale, che è alla base di gran parte dei risultati principali che stabiliscono i fondamenti dell'informatica come scienza (Dauben, 1979; Official Cambridge article link).

Teorema 1 (Cantor).

$$|\mathbb{N}| < |\mathbb{N} \longrightarrow \mathbb{N}| \tag{1}$$

La cardinalità di $\mathbb{N}$ *(numero di programmi per risolvere problemi) è strettamente più piccolo della cardinalità delle funzioni* $\mathbb{N} \longrightarrow \mathbb{N}$ *(numero di problemi esistenti).*

Dimostrazione. Si supponga per assurdo che $|\mathbb{N}| = |\mathbb{N} \longrightarrow \mathbb{N}|$. Questo implica che esistono funzioni numerabili come per esempio $f_0, f_1, f_2, ..., f_x, ...$.

La genialità di Cantor si manifesta quando pensa ad una funzione $g(x)$ così definita:

$$g(x) = f_x(x) + 1 \qquad \text{con } g : \mathbb{N} \longrightarrow \mathbb{N}$$

Con ovviamente $x \in \mathbb{N}$. La funzione $g(x)$ prende un numero naturale e restituisce un numero naturale, quindi è correttamente identificabile come un problema (definizione di problema a pagina 3) e matematicamente formalizzabile come $\mathbb{N} \longrightarrow \mathbb{N}$.

Dunque, prendendo qualsiasi funzione f numerata x-esima, essa sarà diversa dalla funzione g numerata x-esima poiché sempre aumentata di 1:

$$f_x(x) \neq g(x) \longrightarrow f_x(x) \neq f_x(x) + 1$$

QED

Questo teorema purtroppo non è possibile applicarlo agli algoritmi informatici poiché se al posto della funzione $f_x(x)$ venisse inserito un algoritmo e quest'ultimo non terminasse mai, dunque sostituibile con ∞, la somma $+1$ non verrebbe mai eseguita. Per esempio, quando un programma entra in un loop che non gli consente di eseguire le istruzioni successive.

1.4 Problema decisionale e ipotesi del continuo

Un ***alfabeto*** è una sequenza di simboli con cui è possibile scrivere gli algoritmi risolutivi. L'alfabeto utilizzato nelle realizzazioni tecnologiche è l'**alfabeto binario** $\Sigma = \{0, 1\}$.

Un ***problema decisionale*** è la versione associata ad un dato problema informatico $f \in \mathbb{N} \longrightarrow \mathbb{N}$, ovvero alla funzione:

$$d_f : \mathbb{N} \times \mathbb{N} \longrightarrow \{0, 1\} \tag{2}$$

Definita nel seguente modo:

$$d_f((x, y)) = \begin{cases} 1 & \text{se } y = f(x) \\ 0 & \text{altrimenti} \end{cases}$$

Un problema decisionale non è altro che una funzione con co-dominio $\{0, 1\}$ che è in grado di decidere se una data coppia $(x, y) \in \mathbb{N} \times \mathbb{N}$ appartiene ad f.

Essendo un problema decisionale una funzione associata ai problemi in informatica, allora esiste la relazione:

$$\mathbb{N} \longrightarrow \{0, 1\} \subseteq \mathbb{N} \longrightarrow \mathbb{N}$$

Dunque, sicuramente sarà vera la seguente condizione:

$$|\mathbb{N} \longrightarrow \{0, 1\}| \leq |\mathbb{N} \longrightarrow \mathbb{N}|$$

Ma sarà vera anche la seguente:

$$|\mathbb{N} \longrightarrow \{0, 1\}| = |\mathbb{N} \longrightarrow \mathbb{N}|$$

Dimostrazione. È chiaro che la seguente relazione è vera:

$$|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$$

Allora, vale anche:

$$|\mathbb{N} \longrightarrow \{0, 1\}| = |\mathbb{N} \times \mathbb{N} \longrightarrow \{0, 1\}|$$

Si prenda qualsiasi funzione del tipo $f : \mathbb{N} \longrightarrow \mathbb{N}$. A tale funzione, viene associato l'insieme:

$$S_f = \{(i, o) \mid f(i) = o\} \subseteq \mathbb{N} \times \mathbb{N}$$

In cui i indica l'input e o indica l'output. Viene scritta la sua relativa equazione caratteristica:

$$f_{S_f}(x, y) = \begin{cases} 1 & \text{se } (x, y) \in S_f \\ 0 & \text{altrimenti} \end{cases}$$

Allora si può affermare con certezza:

$$|\mathbb{N}| < |\mathbb{N} \longrightarrow \mathbb{N}| = |\mathbb{N} \longrightarrow \{0, 1\}| = |2^{\mathbb{N}}| = |\mathbb{R}| \tag{3}$$

L'ultima uguaglianza è possibile grazie all'**ipotesi del continuo**. QED