# Elementi di calcolabilità e complesità

*Matteo Giorgi*

*a.a. 2020*

# Note (._. )

Queste note sono state prese dal sottoscritto durante le lezioni del corso di *Elementi di calcolabilità e complessità* tenuto dal prof. Pierpaolo Degano con l'ausilio del dott. Giulio Masetti durante l'anno accademico 2019-2020.

Il materiale usato per la stesura, oltre alle note distribuite dal professore, comprende, in maniera più o meno estensiva, i seguenti testi:

- Introduction to the theory of computation - Sipser
- Computability, introduction to recursive function theory - Cutland
- Introduzione alla complessità computazionale - Bernasconi, Codenotti
- Linguaggi, modelli, complessità - Ausiello, d'Amore, Gambosi

---


**Pierpaolo Degano**

- ufficio 285, dipartimento di Informatica
- `di.unipi.it/~degano`
- `degano@unipi.di.it`
- 050 221 2757

**Giulio Masetti**

- ufficio I-25, edificio B, CNR
- `giulio.masetti@isti.cnr.it`

**Matteo Giorgi (me)**

- `matteogiorgi.github.io`
- `matteo.giorgi@protonmail.com`

# Lezione 01 2020.02.17

## Idea intuitiva di algoritmo

Il punto di partenza della teoria della calcolabilità è l'esigenza di formalizzare l'idea intuitiva di *funzione calcolabile* da un algoritmo, ovvero di *funzione algoritmica*[1].

Si può dire che un algoritmo sia un procedimento di calcolo che consenta di pervenire alla soluzione di un problema, numerico o simbolico, mediante una sequenza finita di operazioni, completamente e univocamente determinate: una serie di istruzioni la cui esecuzione consenta di trasformare l'insieme finito di dati simbolici che descrivono il problema, nella soluzione del problema stesso.

### Caratteristiche distintive

1. l'insieme delle istruzioni che definisce l'algoritmo è finito
2. l'insieme delle informazioni (istruzioni) che rappresentano il problema
   - ha cardinalità finita
   - ha un effetto limitato su dati discreti
   - è descritta da dati finiti
3. esiste un agente di calcolo in grado di eseguire le istruzioni
4. il procedimento di calcolo (o *computazione*) è suddiviso in passi discreti e non fa uso di dispositivi analogici
5. la computazione è deterministica: la sequenza dei passi computazionali è determinata senza ambiguità

In conclusione, una procedura algoritmica riceve in ingresso una descrizione finita dei dati del problema e restituisce, dopo un tempo finito, una descrizione finita del risultato. La sua natura deterministica fa sì che l'algoritmo fornisca sempre lo stesso risultato ogni volta che riceve in ingresso gli stessi dati: così facendo l'algoritmo stabilisce una relazione funzionale tra l'insieme dei dati e quello dei risultati.

[1] la nozione di funzione algoritmica sarà poi formalizzata dal concetto di *funzione ricorsiva*

# Macchina di Turing

Esattamente come *funzioni ricorsive* e *λ-calcolo*, le Macchine di Turing sono un modello di calcolo. Nella sua versione più tradizionale una *MdT* si presenta come un dispositivo che accede ad un nastro potenzialmente illimitato diviso in celle, ciascuna contenente un simbolo appartenente ad un dato alfabeto Σ (alfabeto della macchina), comprendente i simboli # e ▷ che denotano rispettivamente l'assenza di informazione nella cella e la marca di inizio stringa:

- la MdT opera tramite un cursore, che può scorrere sul nastro in entrambe le direzioni e scrivere i caratteri $\sigma_i$ appartenenti all'alfabeto Σ.
- gli stati $q_i$ della macchina appartengono all'insieme finito degli stati $Q$ e identificano le configurazioni istante per istante della macchina (posizione cursore, carattere letto, informazioni memoria).

  $\triangleright\ a\ b\ ...\ a\ \underline{a}\ b\ \#\ \#$
- il meccanismo che fa evolvere la computazione della macchina è detta funzione di transizione $\delta$ e permette, partendo dallo stato $q_x$ attuale e dal carattere $\sigma_x$ presente sulla cella puntata dala testina, di portare la macchina in un'altro stato $q_y$, scrivere un carattere $\sigma_y$ su tale cella precedentemente occupata da $\sigma_x$ ed eventualmente spostare la testina (←, →, ↮).

Supponiamo di avere un impiegato diligente ma stupido (che Turing chiamava *the computer*) che esegue a pedissequamente le istruzioni fornitegli: è la nostra Macchina di Turing (MdT), che posso schematizzare con una quadrupla:

$$M = (Q, \Sigma, \delta, q_0)$$

$Q$ è l'insieme finito degli stati

$\Sigma$

$\delta$

$q_0$

I razionali sono isomorfi ai naturali (ne esiste una corrispondenza biunivoca): verifica a coda di rondine che mi crea una funzione bigettiva $N^2N$

Ricorda la definizione di funzione: sottoinsieme del prodotto cartesiano $AxB$ tale che $f : A - B$

# Lezione 02 `2020.02.18`

## Hello

Tristique sed, semper in, hendrerit non, sem. Vivamus dignissim massa in ipsum. Morbi fringilla ullamcorper ligula. Nunc turpis. Mauris vitae sapien. Nunc luctus bibendum velit.

Morbi faucibus volutpat sapien. Nam ac mauris at justo adipiscing facilisis. Nunc et velit. Donec auctor, nulla id laoreet volutpat, pede erat feugiat ante, auctor facilisis dui augue non turpis. Suspendisse mattis metus et justo. Aliquam erat volutpat. Suspendisse potenti. Nam hendrerit lorem commodo metus laoreet ullamcorper. Proin vel nunc a felis sollicitudin pretium. Maecenas in metus at mi mollis posuere. Quisque ac quam sed massa adipiscing rutrum. Vestibulum ipsum. Phasellus porta sapien. Maecenas venenatis tellus vel.