

Fondamenti di informatica

UniVR - Dipartimento di Informatica

Fabio Irimie

1° Semestre 2025/2026

Indice

1	Introduzione	2
1.1	Cos'è l'informatica?	2
1.2	Origini dell'informatica	2
1.2.1	Calcolabilità	2

1 Introduzione

1.1 Cos'è l'informatica?

È una scienza che studia la calcolabilità, cioè cerca di capire che problemi si possono risolvere con un programma. Nasce dall'unione di matematica, ingegneria e logica. Il computer è solo uno strumento, mentre la matematica è il linguaggio con cui si creano algoritmi che permettono di risolvere i problemi.

1.2 Origini dell'informatica

Hilbert, nel 1900, si pose l'obiettivo di formalizzare tutta la matematica con un insieme finito e non contraddittorio di assiomi. Nel 1931, invece, Gödel dimostrò che l'informatica non potrà mai rappresentare tutta la matematica, perché ci saranno sempre proposizioni vere ma non dimostrabili tramite il calcolo. Ci si iniziò a chiedere se esistessero modelli di calcolo meccanici in grado di risolvere tutti i problemi. Nel 1936, Turing propose la macchina di Turing, una **sola** macchina programmabile in grado di risolvere tutti i problemi risolvibili:

$$\text{Int}(P, x) = \begin{cases} P(x) & \text{se } P(x) \text{ termina} \\ \uparrow & \text{se } P(x) \text{ non termina} \end{cases}$$

dove P è un programma e x è un input. La macchina di Turing è un modello teorico di calcolatore, che non esiste fisicamente, ma è in grado di simulare qualsiasi altro calcolatore. Da questo modello deriva la concezione di calcolabilità, cioè se un problema è intuitivamente calcolabile, allora esiste un programma in grado di risolverlo.

Altri modelli di calcolo che sono stati proposti sono:

- Lambda-calcolo
- Funzioni ricorsive
- Linguaggi di programmazione (Turing-completi)

Definizione utile 1.1. La Turing-completezza è la proprietà di un linguaggio di programmazione di essere in grado di simulare una macchina di Turing, cioè di poter risolvere qualsiasi problema risolvibile.

1.2.1 Calcolabilità

Un programma è calcolabile se termina, ma non è detto che termini in un tempo ragionevole. Non esistono algoritmi che possono dire se un programma termina o meno. Questo è un esempio di problema non calcolabile.

I problemi non calcolabili sono infinitamente più numerosi di quelli calcolabili