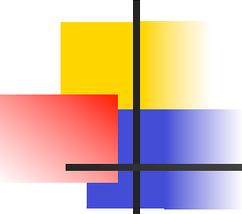


# Per un vocabolario filosofico dell'informatica

---

Angelo Montanari  
Dipartimento di Matematica e Informatica  
Università degli Studi di Udine

Udine, 11 maggio, 2015

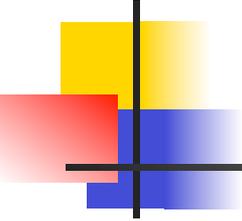


# Obiettivi del corso

---

In un ciclo di seminari, abbiamo messo a fuoco

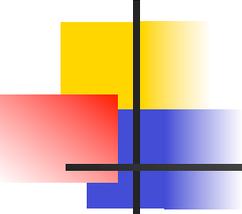
- **declinazioni particolari** (in ambito informatico) di termini da sempre appartenenti al vocabolario filosofico, quali intelligenza (artificiale), conoscenza, rappresentazione, ontologia, linguaggio, determinismo, tempo, infinito
- **termini caratteristici** dell'informatica ``filosoficamente'' rilevanti, quali algoritmo, decidibilità, modello di calcolo, complessità, trattabilità



# Informatica e filosofia

---

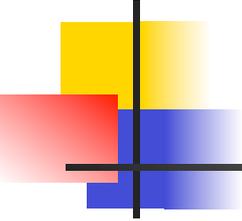
- Obiettivo del ciclo di seminari era stimolare una riflessione **multidisciplinare** sul tema proposto
- Necessità di **colmare** i numerosi **vuoti** della riflessione sulla valenza filosofica dell'informatica
- Esistono numerosi libri di divulgazione riguardanti vari aspetti dell'informatica, esistono anche libri di **storia** dell'informatica, ma mancano libri che ne analizzino in modo sistematico le **implicazioni filosofiche** (i contributi fondamentali dell'informatica non fanno parte della cultura condivisa)



# Struttura del corso

---

- **Intelligenza artificiale (IA)** e dintorni (conoscenza, rappresentazione, ragionamento, ontologia, intelligenza, intenzionalità, apprendimento, pianificazione)
- Le **parole dell'informatica** (algoritmo, decidibilità, modello di calcolo, complessità computazionale, trattabilità)
- **Parole note, nuovi significati** (linguaggio, determinismo, tempo, infinito)



# Un vocabolario antropomorfo

---

L'uso di un **vocabolario antropomorfo** nella descrizione delle caratteristiche e del funzionamento dei sistemi informatici

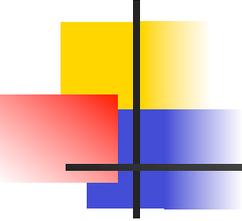
- è particolarmente evidente nel caso dei sistemi di IA (intelligenza, conoscenza, apprendimento, ragionamento)
- ma si è verificato in misura più o meno rilevante in molti altri casi (memoria, comunicazione, interrogazione)

**Ragioni** dell'uso di un vocabolario antropomorfo:

- uomo/animale come modello in cibernetica (N. Wiener, *Cybernetics or Control and Communication in the Animal and the Machine*, MIT Press, 1962) e successivamente in molti ambiti dell'informatica (IA, robotica, bionica, ..)

# Norbert Wiener





# Il rapporto uomo/macchina

---

- Osservazione: ogni **discorso** sulle proprietà "antropomorfiche" delle macchine/calcolatori non riguarda tanto la macchina (il calcolatore) in sé, ma il modo in cui noi vediamo la macchina, e indirettamente noi stessi (Minsky rivendica la legittimità/utilità dell'uso di termini antropomorfici)
- Ciò vale, ad esempio, per la questione relativa alla **intelligenza/intenzionalità** delle **macchine**: parlare delle macchine è un modo (indiretto) per parlare di noi stessi (è una questione antropologica)

# Intelligenza e macchine

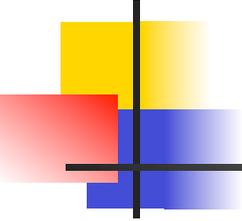
Il test di Turing (o **gioco dell'imitazione**): una  
può essere definita intelligente se riesce a convincere  
una persona che il suo comportamento, dal punto di  
vista intellettuale, non è diverso da quello di un essere  
umano medio



# Alan Turing

---





# La stanza cinese di Searle

---

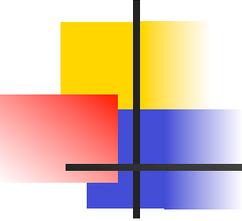
L'obiezione di Searle: l'esecuzione di un programma su un dato input (**processo** nel linguaggio informatico comune) non è mai di per se stessa una condizione sufficiente per l'intenzionalità

**La "dimostrazione"** (un esperimento mentale)

Searle immagina di sostituire un agente umano al calcolatore nel ruolo di esecutore di una specifica istanza di un programma e mostra come tale esecuzione possa avvenire alcuna forma significativa di intenzionalità

# John Searle





# La nozione di algoritmo

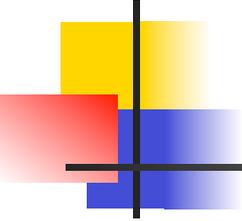
---

Un **algoritmo** è una descrizione finita e non ambigua di una sequenza di passi che consente di risolvere un determinato **problema** (esempi: il problema della ricerca del massimo di un insieme di numeri naturali distinti, il problema dell'ordinamento)

## **Caratteristiche**

Uniformità: un algoritmo si applica a tutte le istanze di un dato problema, potenzialmente infinite, e non ad una singola istanza, e la sua formulazione non dipende dalla singola istanza, ma è la stessa per ogni istanza

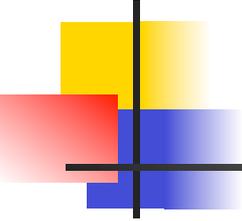
Effettività: la raggiungibilità della soluzione voluta in un tempo finito (equivalentemente, in un numero finito di passi)



# Problemi indecidibili

---

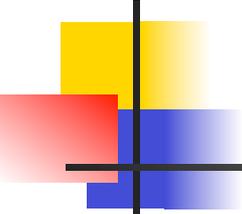
- Uno dei risultati più importanti (della logica matematica e) dell'informatica è stato la scoperta dell'esistenza di problemi per i quali non esistono algoritmi/programmi in grado di risolverli (**problemi indecidibili**)
- E non si tratta di problemi marginali o irrilevanti, ma, in molti casi, di problemi di importanza fondamentale
- Ad esempio, non esiste un algoritmo in grado di stabilire se, dati un programma e un suo possibile input, l'esecuzione di tale programma sullo specifico input termina o meno (**problema della terminazione**)



# Problemi intrattabili

---

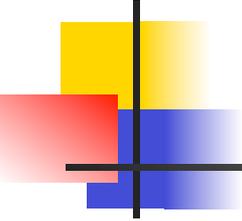
- La situazione è, in verità, ancora "peggiore": vi sono problemi decidibili la cui soluzione risulta troppo onerosa dal punto di vista delle risorse di tempo di calcolo e/o di spazio di memoria necessarie ad un algoritmo per risolverli (**problemi intrattabili**)
- E' questo il caso degli **scacchi**: un programma che volesse esaminare gli effetti di tutte le mosse possibili in modo da poter scegliere, ad ogni passo, la miglior mossa in assoluto richiederebbe l'analisi di un numero di mosse molto maggiore del numero totale di protoni presenti nell'universo (necessità di **euristiche**)



# Infinito: da problema a risorsa

---

- Programmi standard: a fronte di un dato input, devono produrre un determinato output
- L'**infinito** come **problema**: dobbiamo garantire che il programma produca l'output atteso in un tempo finito, ossia che il programma sia terminante
- Altra importante classe di programmi, detti **programmi reattivi** (esempio, i sistemi operativi), la cui funzione è quella di mantenere nel tempo una certa modalità di interazione con l'ambiente in cui operano
- L'**infinito** come **risorsa**: i programmi reattivi sono inerentemente non terminanti (**computazioni infinite**)



## Per saperne di più ...

---

- I lucidi del corso possono essere scaricati all'indirizzo:

<https://users.dimi.uniud.it/~angelo.montanari/digressions.php>