



# Parole note, nuovi significati: linguaggio, determinismo e infinito

---

Angelo Montanari

Dipartimento di Matematica e Informatica

Università degli Studi di Udine

Ciclo di seminari su un Vocabolario Filosofico  
dell'Informatica

Udine, 27 gennaio, 2016



# Parole note, nuovi significati

---

- **linguaggio**
- **determinismo/non determinismo**
- **infinito**
- tempo



# Alfabeto e linguaggio

---

Come costruire insiemi infiniti di parole (**linguaggi**) a partire da un **alfabeto** finito di simboli

Ogni alfabeto finito può essere codificato in un **alfabeto binario**

L'**espressività** di un linguaggio non dipende dal suo alfabeto, ma dalle regole per la costruzione delle parole



# Linguaggi naturali e formali

---

Distinzione fondamentale: linguaggi formali e linguaggi naturali

- Un **linguaggio formale** è un insieme, in generale infinito, di parole (finite o infinite), costruite a partire da un alfabeto finito attraverso un opportuno insieme di regole
- Delle tecniche di elaborazione automatica del **linguaggio naturale** si occupa, invece, un settore importante dell'intelligenza artificiale (se veda la voce intelligenza)



# I linguaggi di programmazione

---

L'applicazione pratica più importante della nozione di linguaggio formale è quella nell'ambito dei linguaggi di programmazione

**I linguaggi di programmazione**, nella varietà dei paradigmi di programmazione (imperativo, funzionale, logico, ad oggetti), definiscono le regole per la stesura di programmi sintatticamente corretti (si veda la voce algoritmo)



# I linguaggi formali

---

Tre punti di vista alternativi sui linguaggi formali:

- linguaggi **generati** da una grammatica (linguaggio = insieme delle parole generate dalla grammatica)
- linguaggi **accettati** da una macchina/automa (linguaggio = insieme delle stringhe accettate dall'automa)
- linguaggi **definiti** da una formula (linguaggio = insieme dei modelli della formula)



# Linguaggi e macchine

---

Il punto di vista più originale è quello che stabilisce un legame tra linguaggi formali e macchine/automati: una **concezione operativa** dei linguaggi (formali)

Un linguaggio (formale) è una macchina che riconosce insiemi di oggetti (parole, ma anche alberi o grafi)



# La gerarchia di Chomsky - 1

---

- La caratterizzazione dei linguaggi formali in termini di grammatiche è essenzialmente dovuta a **Chomsky**
- La **gerarchia di Chomsky** distingue quattro livelli di grammatiche
- Ogni **livello di grammatiche** può essere messo in corrispondenza con una **classe di macchine**





# La gerarchia di Chomsky - 2

---

- le **grammatiche di tipo 0**, che includono tutte le grammatiche formali e consentono di generare tutti i linguaggi riconosciuti da macchine di Turing

## Osservazione

L'interpretazione delle coppie di input/output come parole consente l'identificazione dei problemi risolti da una macchina di Turing coi linguaggi riconosciuti da una macchina di Turing



# La gerarchia di Chomsky - 3

---

- le **grammatiche sensibili al contesto**, che consentono di generare tutti e soli i linguaggi riconosciuti da macchine di Turing (non-deterministiche) limitate linearmente
- le **grammatiche libere dal contesto**, che consentono di generare tutti e soli i linguaggi riconosciuti da automi a pila (non-deterministici)
- le **grammatiche regolari**, che consentono di generare tutti e soli i linguaggi riconosciuti da automi a stati finiti



# Non determinismo

---

- Il non determinismo in informatica può essere visto come uno strumento per **astrarre** parti sconosciute o complesse dei sistemi
- Differenze tra il non determinismo nell'informatica e il principio di indeterminazione nella fisica (che, però, interviene nella computazione quantistica)



# Determinismo/non determinismo

---

- **Macchine** deterministiche e non deterministiche (equivalenza / non equivalenza)

Esempio

Automi a stati finiti che operano su parole finite vs. automi di Buechi che operano su parole infinite

- **Algoritmi** deterministici e non deterministici (le istruzioni di scelta non deterministica)
- Determinismo/non determinismo e teoria della **complessità**



# MdT non deterministica - 1

---

La proprietà distintiva di una macchina di Turing non deterministica è che, dati il simbolo contenuto nella cella corrente e lo stato corrente del controllo finito, essa non è obbligata ad eseguire un'unica azione univocamente determinata, ma può scegliere fra diverse possibili azioni

A fronte di un dato input, una macchina di Turing non deterministica ammette **più computazioni** alternative



## MdT non deterministica - 2

---

Condizione di accettazione di una parola da parte di una macchina di Turing non deterministica definita in termini dell'**esistenza di una computazione** di successo

Il comportamento di una macchina di Turing non deterministica può essere simulato da una macchina di Turing deterministica con una perdita esponenziale di efficienza (la questione circa l'inevitabilità o meno di tale degrado delle prestazioni è una formulazione equivalente del problema: "P = NP?")



# Dall'infinito come problema

---

Programmi sin qui considerati: a fronte di un dato input, devono produrre un determinato output

L'infinito si presenta come un problema: dobbiamo garantire che il programma produca l'output atteso in un tempo finito, ossia che il **programma** sia **terminante**



# all'infinito come risorsa

---

- Vi è un'altra classe di programmi/sistemi di fondamentale importanza, detti **programmi/sistemi reattivi** (ad esempio, i sistemi operativi), la cui funzione è quella di mantenere nel tempo una certa modalità di interazione con l'ambiente in cui operano
- Tali **programmi** sono inerentemente **non terminanti** (computazioni infinite)





# Le logiche temporali

---

Come specificare e verificare in modo automatico le proprietà attese dei programmi/sistemi reattivi:

- proprietà di sicurezza (safety)
- proprietà di vitalità (liveness)
- proprietà di precedenza

Il ruolo della **logica temporale**



# Finito e infinito

---

- Rappresentazione finita di **oggetti infiniti** (computazioni infinite, spazio degli stati infinito)
- Sistemi a stati finiti e sistemi a **stati infiniti**
- Il ruolo dell'**astrazione** (di nuovo un termine con valenza filosofica che già abbiamo incontrato, ad esempio, nella gerarchia dei linguaggi di programmazione e nella definizione di non determinismo)