Le parole dell'informatica: modello di calcolo, complessità e trattabilità

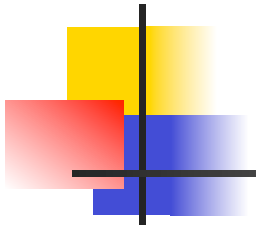
Angelo Montanari

Dipartimento di Matematica e Informatica

Università degli Studi di Udine

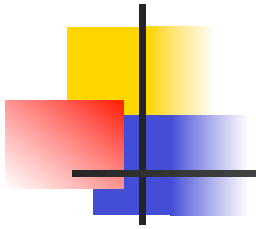
Ciclo di seminari su un Vocabolario Filosofico dell'Informatica

Udine, 5 dicembre, 2014



Le parole dell'informatica

- algoritmi (teoria degli)
- decidibilità/indecidibilità
- **modelli di calcolo**
- **complessità (teoria della)**
- **trattabilità**



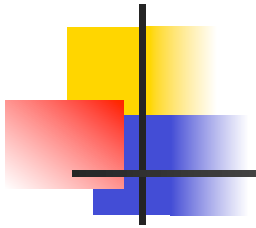
La macchina di Turing

Un modello di calcolo, o di computazione, è una macchina in grado di eseguire algoritmi

Il modello di calcolo di riferimento in informatica è un modello molto semplice detto **macchina di Turing**

Alcuni semplici esempi di macchina di Turing:

- complemento bit a bit
- ordinamento di stringhe binarie

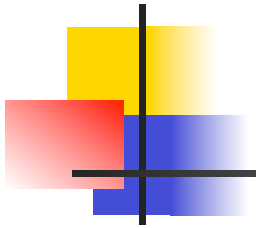


La tesi di Church

Quali problemi algoritmici possono essere risolti con una macchina di Turing?

La **tesi di Church** afferma che le macchine di Turing sono in grado di risolvere tutti i problemi algoritmici effettivamente risolubili

E' una tesi, non un teorema (la nozione di risolubilità effettiva non è formalizzata), ma da tutti accettata



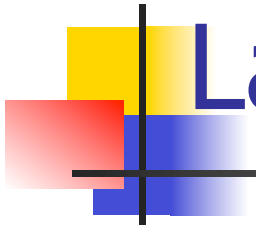
Modelli di calcolo alternativi

Tutti i modelli di calcolo proposti per catturare la nozione di computabilità effettiva a partire dagli anni '30 del secolo scorso, quali

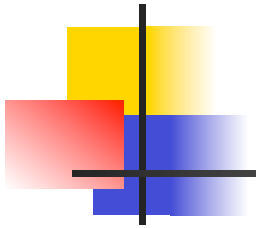
- lambda calcolo di Church
- regole di produzione di Post
- classe delle funzioni ricorsive di Kleene

sono fra loro equivalenti ed equivalenti alla macchina di Turing

La macchina di Turing universale



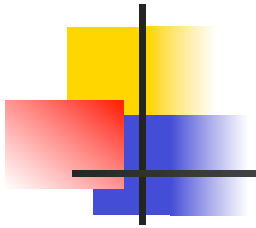
- Turing mostrò anche come creare una singola macchina di Turing (**macchina di Turing universale**) in grado di fare tutto ciò che può essere fatto da una qualsiasi macchina di Turing
- Una tale macchina di Turing riceve in ingresso una macchina di Turing MT e un input per essa e si comporta esattamente come si comporterebbe la macchina MT su tale input



Il calcolatore universale

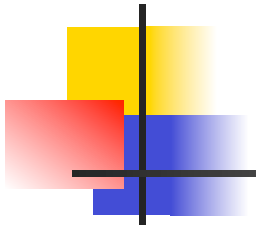
- E' il modello matematico del **calcolatore universale**

- MT e il suo input sono trattati come dati dalla macchina di Turing universale che si comporta come un interprete degli attuali linguaggi di programmazione



Nuovi modelli di calcolo

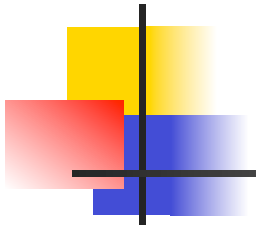
- Negli ultimi anni hanno acquistato un certo rilievo nuovi modelli di calcolo, incluse la computazione quantistica (**quantum computing**) e la computazione molecolare (**DNA computing**)
- La tesi di Church rimane valida
- Ad esempio, un computer quantistico è in grado di emulare ogni computazione effettuata con uno dei modelli di calcolo classici, ma vale anche il viceversa: un modello di calcolo classico è in grado di simulare qualsiasi computazione quantistica



Complessità (Teoria della)

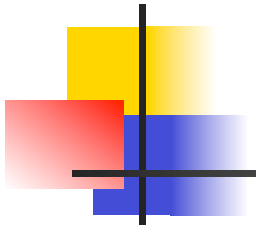
- La distinzione tra problemi decidibili e indecidibili non è l'unica distinzione di interesse in informatica

- Ci sono problemi computabili/decidibili la cui **soluzione** risulta **troppo costosa** dal punto di vista delle risorse di tempo di calcolo e/o di spazio di memoria necessarie ad un algoritmo per risolverli



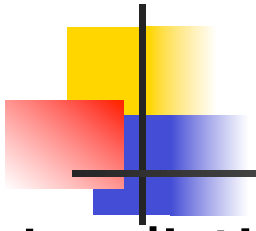
Complessità computazionale

- La classificazione dei problemi sulla base dell'ammontare di risorse (**tempo** e/o **spazio**) richiesto per la loro soluzione da parte di un qualche strumento di calcolo (ad esempio, una macchina di Turing) prende il nome di (teoria della) **complessità computazionale**



Alcuni elementi fondamentali

- In generale, tempo e spazio necessari dipendono (sono funzione di) dalla dimensione dell'input: **analisi asintotica** (al crescere della dimensione dell'input)
- A parità di dimensione, la complessità può variare al variare dell'input; di norma, **analisi del caso peggiore** (non è l'unica possibile, né sempre la più ragionevole)

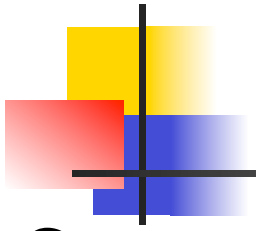


Complessità dei problemi

La distinzione tra

- **complessità degli algoritmi**
- **complessità dei problemi**

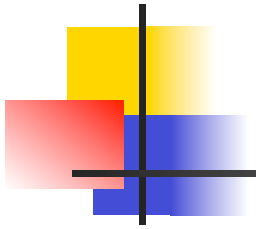
Limiti superiori e inferiori alla complessità dei problemi (ad esempio, il problema della calcolo del massimo di un insieme di numeri naturali distinti): se coincidono, **soluzioni ottimali**



Misura della complessità

Come **misurare** la complessità temporale (di un algoritmo)?

- Il tempo si misura calcolando il numero di azioni elementari effettuate durante l'esecuzione di un programma espresso in funzione della dimensione dell'input
- A seconda della struttura di tale funzione, si parla di tempo logaritmico, di tempo polinomiale, di tempo esponenziale nella dimensione dell'input

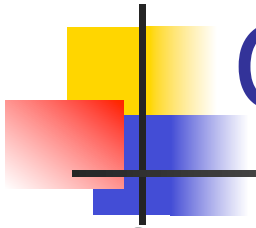


Trattabilità e intrattabilità

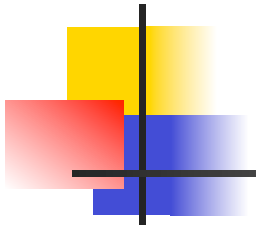
Distinguiamo tra algoritmi (buoni) che richiedono un tempo polinomiale, o inferiore, e algoritmi (cattivi) che richiedono un tempo esponenziale, o superiore (trascuriamo ciò che sta nel mezzo)

Un problema per il quale esiste una soluzione algoritmica buona è detto **trattabile**; un problema che ammette solo soluzioni algoritmiche cattive è detto **intrattabile**

Complessità e modelli di calcolo



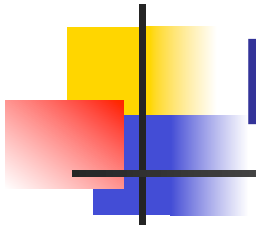
- Secondo la tesi di Church, la computabilità / decidibilità di un problema non dipende dal modello (classico) di calcolo di riferimento
- E' possibile mostrare che un risultato analogo vale per la complessità (trattabilità) di un problema: i modelli di calcolo sono **correlati polinomialmente** (un problema che può essere risolto in un dato modello, può essere risolto in qualsiasi altro modello e la differenza rispetto al tempo di computazione può essere espressa con una funzione polinomiale)



Una questione

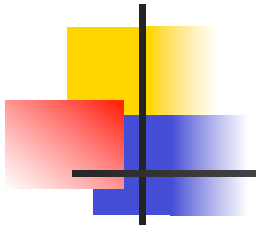
Ciò non sembra necessariamente valere per la computazione quantistica..

Ciò differenzia la questione circa la decidibilità/ indecidibilità di un problema da quella sulla trattabilità/intrattabilità



Problemi fortemente intrattabili

- Non tutti i problemi che ammettono solo soluzioni algoritmiche cattive sono uguali
- Vi sono, ad esempio, problemi che hanno una complessità doppiamente esponenziale (è questo il caso dell'aritmetica di Presburger)
- In generale, un problema si dice **elementarmente decidibile** se può essere risolto da un algoritmo che, su un input di dimensione n, esegue un numero di operazioni superiormente limitato da una torre di esponenziali, con un numero prefissato di esponenti



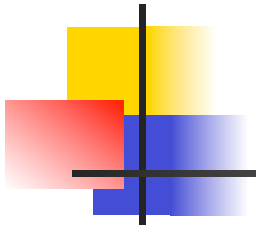
Gerarchia classi di complessità

Esistono problemi decidibili che **non** sono

elementarmente decidibili

E' questo il caso del problema della (in)soddisfacibilità delle formule della logica monadica al second'ordine di un successore WS1S (lo stesso vale per il suo frammento al prim'ordine)

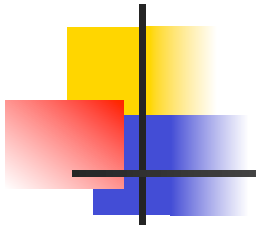
In analogia con quanto fatto con i problemi indecidibili, è possibile organizzare gerarchicamente i problemi decidibili sulla base della loro complessità (**gerarchie di classi di complessità**)



Problemi NP-completi

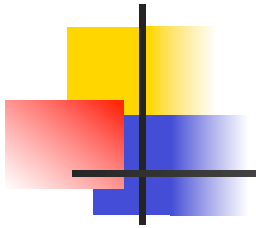
- **Problemi NP**. Vi è una classe di problemi decidibili che possiedono limiti inferiori polinomiali (lineari o al più quadratici), per i quali sono note soluzioni esponenziali, ma non soluzioni polinomiali

- **NP-completezza**. I problemi in tale classe sono fra loro equivalenti (nel senso della riducibilità)



Problemi NP-completi

- Tali problemi sono caratterizzati dal fatto che **verificare** se una possibile soluzione è veramente tale richiede un **tempo polinomiale** (“E’ più facile criticare che fare" – Moshe Vardi)
- Inoltre, è possibile mostrare che essi possono essere risolti in tempo polinomiale da un **algoritmo non deterministico** (in ogni punto di scelta tale algoritmo fa la scelta giusta, se una tale scelta esiste)



P versus NP

- Se indichiamo con P la classe dei problemi la cui soluzione si può ottenere in tempo polinomiale, ovviamente P è un sottoinsieme di NP

- Stabilire se

P = NP

è un problema aperto