Computabilità, Complessità e Logica

Lezione 18

Un problema PSPACE-completo

Supponiamo di avere una formula $\phi(x_1, \ldots, x_n)$ in n variabili $x_1, \ldots, x_n$. Per semplicità supponiamo n pari.

Possiamo definire una formula in cui tutte le variabili sono quantificate:

$$\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n \phi(x_1, \ldots, x_n)$$

Cosa significa?

Un problema PSPACE-completo

- $\exists x_i \phi(x_i)$ è valido se almeno uno tra $x_i = \text{true}$ e $x_i = \text{false}$ rende vera la formula $\phi(x_i)$
- $\forall x_i \phi(x_i)$ è valido se sia $x_i = \text{true}$ che $x_i = \text{false}$ rendono vera la formula $\phi(x_i)$
- Quindi © è valida se
    - Esiste un valore di x_1 tale per cui..
        - Per ogni valore di x_2 è vero che...
            - Esiste un valore di x_3 tale per cui...
                - $\cdots$
                    - $\phi(x_1, \ldots, x_n)$ è vera

Un problema PSPACE-completo

- Il problema di decisione di **quantified SAT (QSAT)** o SAT quantificato chiede se
- data una formula $\exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n \phi(x_1, \ldots, x_n)$ in cui ogni variabile è quantificata
- la formula è valida
- QSAT è un problema PSPACE-completo
- Non è immediato da vedere il collegamento tra quantificatori e PSPACE, quindi introduciamo la nozione di *macchina alternante*

Macchine alternanti

- Una macchina di Turing **alternante** è una macchina non deterministica definita come$M = (Q, \Sigma, \Gamma, \delta, q_0, g)$

- Dove $g : Q \to \{ \wedge , \ \vee , \text{accept}, \text{reject}\}$ è una funzione che assegna a ogni stato della macchina una tra quattro etichette:

  - $\text{accept}, \text{reject}$ indicano che lo stato è accettante o rifiutante

  - $\wedge$ indica che una configurazione che contiene questo stato è accettante se **tutte** le configurazioni raggiungibili in un passo sono accettanti

  - $\vee$ indica che una configurazione che contiene questo stato è accettante se **esiste** una configurazione raggiungibile in un passo che è accettante

Macchine alternanti

- Queste macchine generalizzano le condizioni di accettazione delle macchine usate per definire NP e coNP
- Per la condizione di accettazione "esiste una computazione accettante", tutti gli stati non accettanti o rifiutanti sono etichettati con $\vee$
- Per la condizione di accettazione "ogni computazione è accettante", tutti gli stati non accettanti o rifiutanti sono etichettati con $\wedge$

Macchine alternanti

- AP è la classe dei linguaggi riconosciuti da macchine alternanti che lavorano in tempo polinomiale
- Chandra, Kozen e Stockmeyer nel 1981 dimostrano che $\mathrm{AP} = \mathrm{PSPACE}$
- In questo caso verificare se una formula quantificata è vera diventa una alternanza tra stati $\vee$ (quando si incontra un quantificatore $\exists$) e stati $\wedge$ (quando si incontra un quantificatore $\forall$)

Approssimazione

- Possiamo riformulare molti problemi di decisione come problemi di ottimizzazione:
  - Minimizzare la lunghezza del percorso (per TSP)
  - Massimizzare il guadagno senza superare la capacità dello zaino
  - Etc.
- Se anche non otteniamo la risposta ottima possiamo arrivarci vicino?

Approssimazione

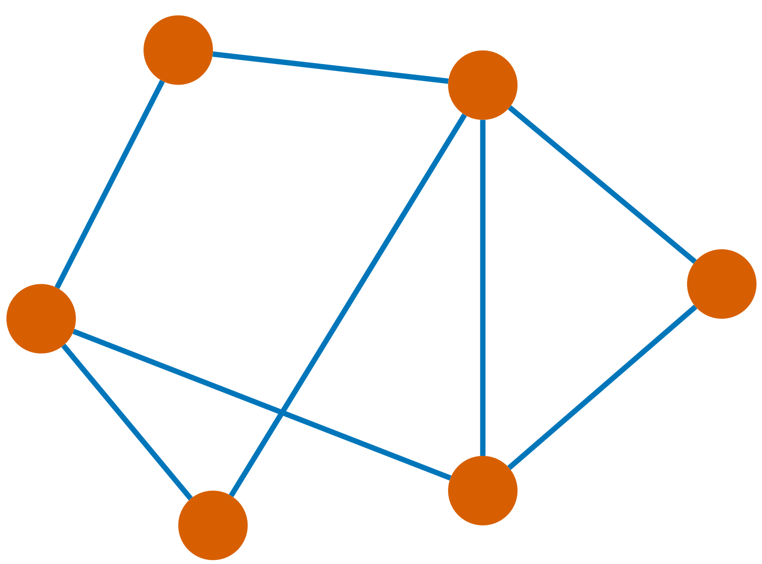
- Si consideri il problema di massimizzare una funzione $f(x)$
- Sia OPT il valore massimo raggiunto da f
- Diciamo che un algoritmo è in grado di approssimare l'ottimo con errore relativo $c \geq 0$ se la soluzione trovata dall'algoritmo è al più $(1 - c)\mathrm{OPT}$
- Similmente, per minimizzazione, se la soluzione trovata è al più $(1 + c)\mathrm{OPT}$

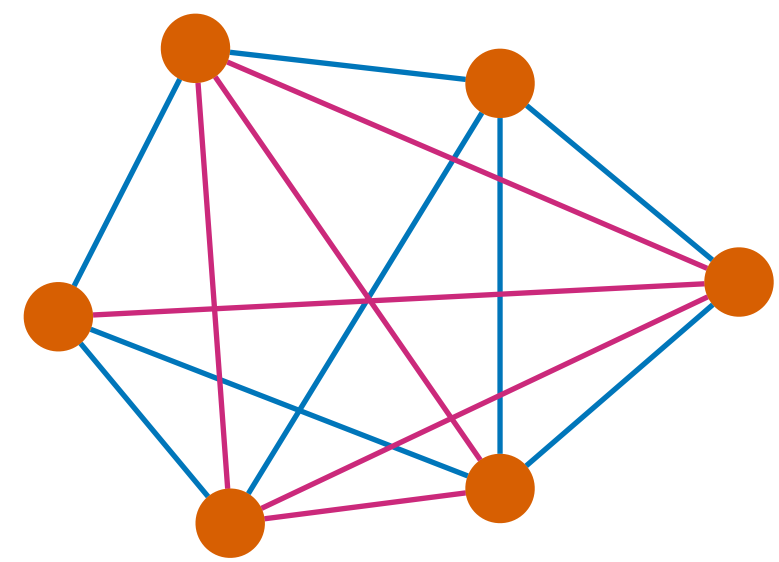
Approssimazione

- Possiamo approssimare bene ogni problema in NP usando algoritmi che lavorano in tempo polinomiale deterministico?
- Dipende dal problema!
- In particolare, approssimare per una qualsiasi costante c alcuni problemi (come TSP) mostrerebbe che $\mathrm{P} = \mathrm{NP}$
- Supponiamo esiste un algoritmo in grado di trovare un percorso al più $(1 + c)\mathrm{OPT}$ per TSP
- Mostriamo che allora che sapremmo risolvere il ciclo Hamiltoniano in tempo polinomiale

Approssimazione

Ipotesi: esiste un algoritmo che lavora in tempo polinomiale
che ci approssima la soluzione ottima di TSP con una costante $c \geq 0$.
Ovvero la lunghezza del ciclo ritornato è al più $(1+c)$ volte quella del ciclo ottimo





Input: un grafo $G = (V, E)$ di n nodi
Output: esiste un ciclo Hamiltoniano?

Se l'arco esiste in G allora ha peso 1
Se l'arco non esiste in G allora ha peso $(1+c)n$

Esiste un ciclo Hamiltoniano in G
L'ottimo per TSP ha valore n

La soluzione approssimata è
minore o uguale a $(1+c)n$

Non esiste un ciclo Hamiltoniano in G
L'ottimo per TSP ha valore maggiore di $(1+c)n$

La soluzione approssimata è
maggiore di $(1+c)n$

Saremmo in grado di dire
se esiste un ciclo Hamiltoniano
in G solo guardando
la soluzione approssimata!