NP completezza

Problemi astratti

- Un problema è un'entità astratta (es. il TSP).
- Una istanza del problema è un suo caso particolare in cui vengono specificati tutti i suoi elementi costitutivi.
- Un programma risolve un problema se può generare una soluzione in corrispondenza di qualunque sua istanza.

Risolvibilitá

Per poter risolvere un problema con un programma è necessario codificare l'istanza da risolvere con una stringa (binaria) comprensibile dal programma.

Codifica: corrispondenza fra l'insieme delle istanze del problema e un insieme di stringhe binarie.

$$e: I \rightarrow \{0,1\}^*$$

Un algoritmo risolve un problema in tempo $O(T(n))$ se, quando gli viene fornita la codifica binaria di una istanza i di lunghezza $n=|i|$, produce una soluzione al più in un tempo $O(T(n))$.

Problemi decisionali

I problemi decisionali sono una classe di problemi dove per ogni possibile ingresso un algoritmo deve scegliere una di due risposte possibili: “si” o “no”.

Si tratta quindi della classe delle funzioni computabili del tipo

f: $\mathbf{N} \rightarrow \{0,1\}$

Problemi decisionali, esempi

- **Problema del sottografo completo.** Dati un grafo G e un intero n, stabilire se il grafo G contiene un sottografo completo con n vertici.
- **Problema del cammino hamiltoniano.** dato un grafo G stabilire se esiste un cammino che tocchi tutti i vertici di G una e una sola volta.
- **Problema del cammino euleriano.** Dato un grafo G stabilire se esiste un cammino che percorra tutti gli archi di G una e una sola volta.

Probl. decisionali, CNF

CNF (*Conjunctive Normal Form*, forma normale congiuntiva): una formula booleana del tipo:

$(x_{1,1} \vee x_{1,2} \vee \ldots \vee x_{1,k1})$ & $(x_{2,1} \vee x_{2,2} \vee \ldots \vee x_{2,k2})$ & ... & $(x_{n,1} \vee x_{n,2} \vee \ldots \vee x_{n,kn})$,

dove $x_{i,j} = v_s$ o $x_{i,j} = \neg v_s$ per un dato insieme di variabili $\{v_1,\ldots,v_m\}$.

- **Problema SAT.** Data una CNF F stabilire se F è soddisfacibile, cioè se esiste un assegnamento di valori 0 e 1 alle variabili in F tale per cui il valore di F per quell'assegnamento è 1.

Probl. decisionali, k-CNF

k-CNF: una formula booleana del tipo:

$(x_{1,1} \vee x_{1,2} \vee \ldots \vee x_{1,k}) \& (x_{2,1} \vee x_{2,2} \vee \ldots \vee x_{2,k}) \& \ldots \& (x_{n,1} \vee x_{n,2} \vee \ldots \vee x_{n,k})$,

ogni congiunto contiene k termini disgiuntivi, e $x_{i,j} = v_s$ o $x_{i,j} = \neg v_s$ per un insieme dato di variabili $\{v_1, \ldots, v_m\}$.

- k-SAT. Data una k-CNF F, stabilire se F è soddisfacibile, cioè se esiste un assegnamento di valori 0 e 1 alle variabili in F, tale per cui il valore di F per quell'assegnamento è 1.

Problemi di ottimizzazione

Spesso il problema non richiede di rispondere si o no, ma di trovare il massimo o il minimo di una funzione (es. TSP, knapsack, scheduling, ...)

Questi sono problemi di ottimizzazione, sono comunque riconducibili a problemi di decisione chiedendosi se esiste una soluzione di costo inferiore (superiore) a una soglia k e istanziando ad es. una ricerca binaria per il minimo k intero.

La complessità di un problema di ottimizzazione e del suo corrispondente problema decisionale è la stessa.

Le classi P ed NP

- Un problema decisionale è nella classe **P** se esiste un algoritmo che *risolve* qualsiasi istanza del problema P in tempo polinomiale.
- Un problema decisionale è nella classe **NP** se esiste un algoritmo che, data una istanza *i* e una sua possibile soluzione s, *verifica la correttezza* della soluzione s in tempo polinomiale (rispetto alla dimensione dell'istanza).

P e NP

Ovviamente $P \subseteq NP$

non è noto se P = NP

la risposta vale 1.000.000 di dollari

(http://www.claymath.org/millennium-problems/millennium-prize-problems)

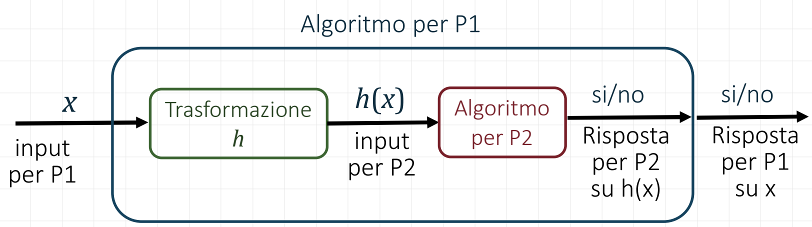
Riducibilitá polinomiale

$g : \mathbf{N} \rightarrow \{0, 1\}$ è riducibile polinomialmente a $f : \mathbf{N} \rightarrow \{0, 1\}$ se esiste una funzione h, calcolabile in tempo polinomiale, tale che per ogni x :

$g(x) = f(h(x))$

Se si sa risolvere f, si sa anche risolvere g !

Notazionalmente: $g \leq_p f$



NP completezza

$f : \mathbf{N} \rightarrow \{0, 1\}$ è NP-completo se e solo se:

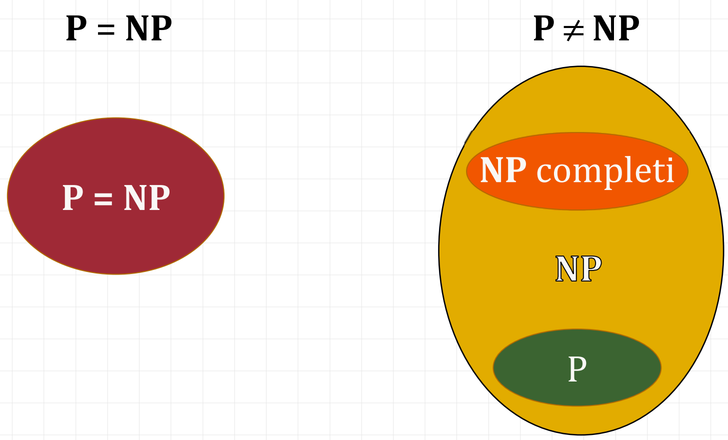
- $f \in \mathbf{NP}$
- per ogni $g \in \mathbf{NP}$ si ha $g \leq_p f$

NPC è la classe dei problemi NP completi.

TEOREMA:

- se un qualunque problema in NPC è risolvibile in tempo polinomiale, allora P=NP.
- *equivalentemente*, se un qualunque problema in NP non è risolvibile in tempo polinomiale, allora tutti i problemi in NPC non sono risolvibili in tempo polinomiale.

P e NP



13

Prove di NP completezza

Difficile: dalla definizione. Si richiede di dimostrare che la funzione è in NP e che qualunque altra funzione in NP è riducibile polinomialmente alla funzione data.

Questo è stato fatto (*Cook 1971, Levin 1973*) per il problema SAT: stabilire se una data formula CNF è soddisfacibile (versione *circuit SAT*).

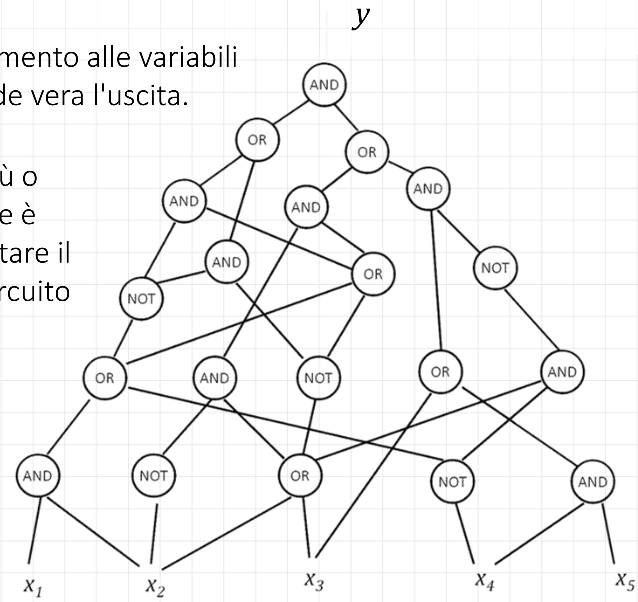
Più facile: mostrare che la funzione f è in NP quindi mostrare che $g \leq_p f$ per qualche problema g che è già noto essere NP completo.

14

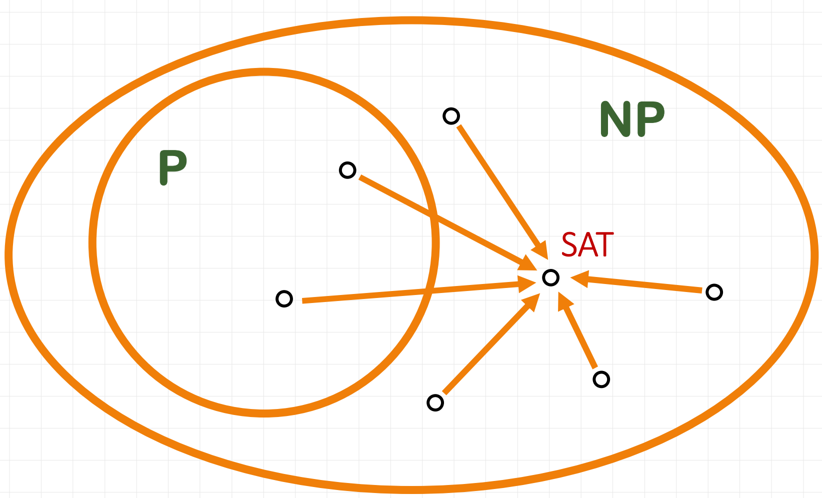
Circuit SAT

Trovare un assegnamento alle variabili in ingresso che rende vera l'uscita.

Intuitivamente, è più o meno come dire che è possibile rappresentare il problema con un circuito elettrico binario.



Ogni problema in NP $\leq_p$ SAT



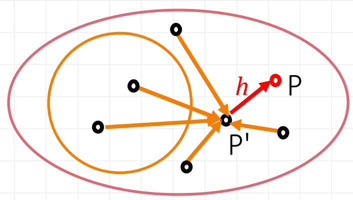
Si può pensare a $\longrightarrow$ come a “è più facile di”.
SAT è il problema più difficile in NP.

Riduzioni: metodologia

Riducendo a P un qualunque problema P' noto essere in NPC, implicitamente si riducono a P tutti i problemi in NP.

Quindi per dimostrare che un problema P è in NPC si può:

1) dimostrare che $P \in NP$
2) selezionare un problema P' in NPC
3) progettare un algoritmo polinomiale che calcola una funzione h che fa corrispondere ad ogni istanza di P' una istanza di P
4) dimostrare che h è tale per cui $x \in P'$ sse $h(x) \in P$, $\forall x$



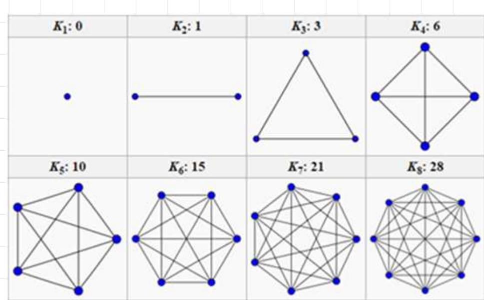
17

NP completezza, esempi di prove

Problema del sottografo completo (max clique). Dati un grafo G e un intero n stabilire se esiste un sottografo completo di G si n vertici.

Prova di **NP**-completezza, si parte da SAT (data una CNF F, stabilire se F è soddisfacibile).

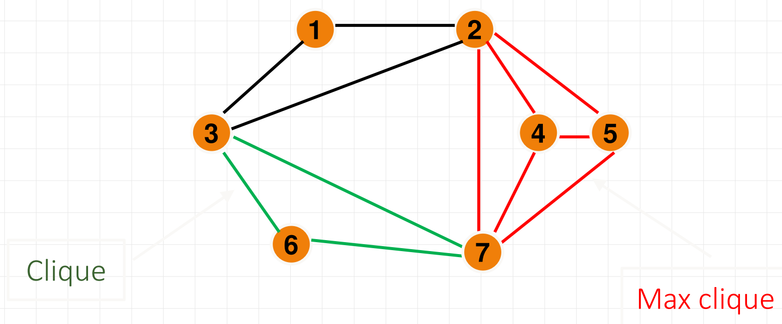
Si assume di sapere già che SAT è **NP**-completo.



18

Clique

Clique: dato un grafo G = (V, E), un sottinsieme S dei suoi vertici forma una clique se ogni coppia di vertici di S è connessa



3-SAT $\leq_p$ CLIQUE

Formula $\Phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$ (k clausole)

3 disgiunti per clausola: $(x_1 \vee x_2 \vee \neg x_3) \wedge \ldots$

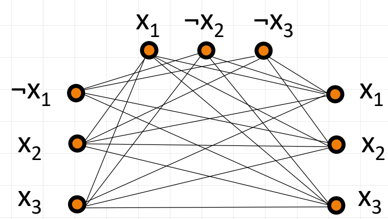
Grafo:

- un vertice per ogni letterale di ogni clausola
- un arco fra due vertici se corrispondenti a:
  1. letterali di clausole diverse e
  2. variabili compatibili

3-SAT $\leq_p$ CLIQUE, esempio

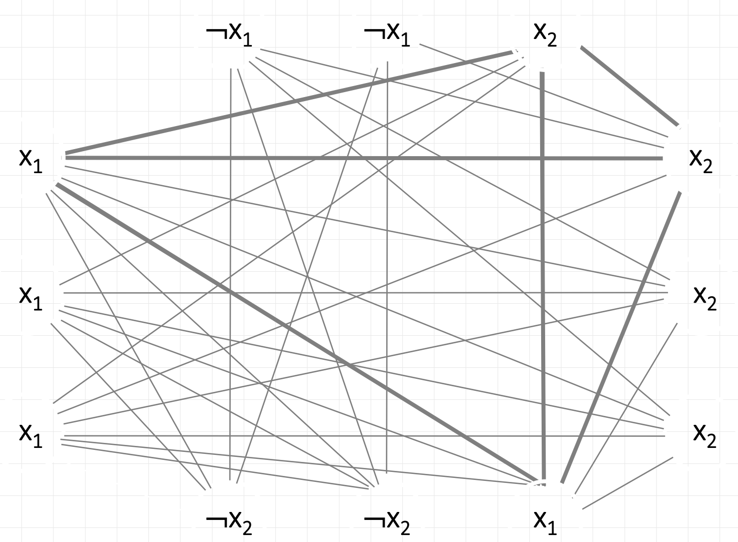
$\Phi = (x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3)$

Grafo:



3-SAT $\leq_p$ CLIQUE, esempio

$(x_1 \lor x_1 \lor x_1) \land (\neg x_1 \lor \neg x_1 \lor x_2) \land (x_2 \lor x_2 \lor x_2) \land (\neg x_2 \lor \neg x_2 \lor x_1)$



3-sat $\leq_p$ Clique

Teorema: Φ è soddisfacibile sse G ha una clique di k vertici

- Φ è soddisfacibile → G ha una clique di k vertici

Se Φ è soddisfacibile allora $\forall C_r$, $\exists \ell_i^r$, un letterale che vale 1.

La ℓ_i^r corrisponde a un vertice v_i^r. Allora $V' = \{v_i^r\}$ è una clique (per $r \neq s$ ℓ_i^r è compatibile con ℓ_j^s).

- G ha una clique di k vertici → Φ è soddisfacibile

Nessun arco in G connette vertici di una tripla (clausola).

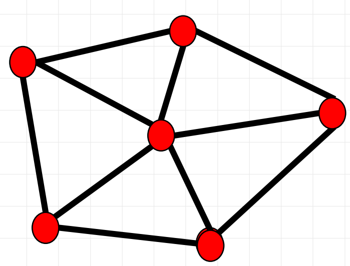
V' ha un vertice per tripla, v_i^r.

Si può porre ℓ_i^r =1.

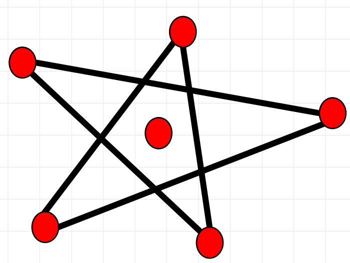
Complemento di un grafo

Dato un grafo G, se G' è il grafo complementare di G allora ogni coppia di nodi è connessa in G' se e solo se non è connessa in G.

G'=(V,E') è complemento di G=(V,E) ↔ $(u,v) \in E'$ sse $(u,v) \notin E$



G



G'

Clique $\leq_p$ Vertex Cover

Vertex Cover:

min $|S|$, $S \subseteq V$ t.c. $\forall (u,v) \in E$ si ha $u \in S$ e/o $v \in S$

(ogni arco del grafo ha almeno un estremo in S)

Input: <G,k> di CLIQUE

sia G' il complemento di G

Output: <G', |V|-k> di vertex cover.

G ha una clique di dimensione k sse G' ha una copertura di dimensione |V|-k.

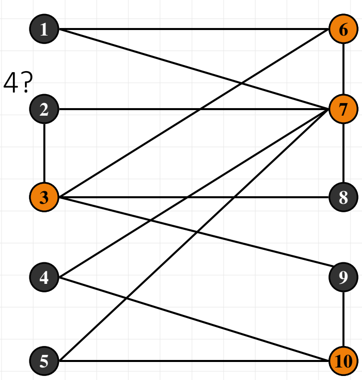
Vertex Cover

VERTEX COVER: Dato un grafo non orientato G = (V, E) e un intero k, c'è un sottinsieme di vertici $S \subseteq V$ tale che $|S| \leq k$, e per cui se $(v,w) \in E$ allora $v \in S$ oppure $w \in S$ oppure entrambi?.

Es.

- C'è un vertex cover di dimensione 4?

  SI.

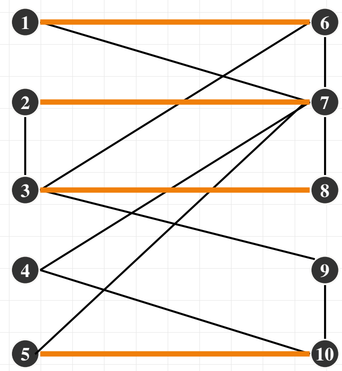


Vertex Cover

VERTEX COVER: Dato un grafo non orientato G = (V, E) e un intero k, c'è un sottinsieme di vertici $S \subseteq V$ tale che $|S| \leq k$, e per cui se $(v, w) \in E$ allora $v \in S$ oppure $w \in S$ oppure entrambi?.

Es.

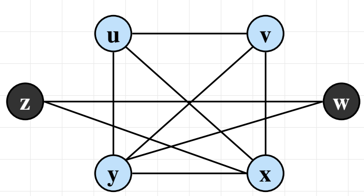
- C'è un vertex cover di dimensione 4?
  SI.
- C'è un vertex cover di dimensione 3?
  NO.



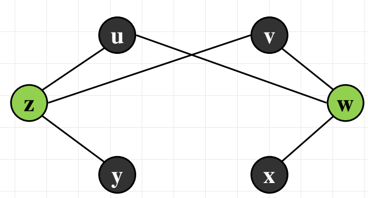
Clique $\leq_p$ Vertex Cover

VERTEX COVER $\leq_p$ CLIQUE.

- Dato un grafo non orientato G = (V, E), e il suo complemento G' = (V, E'), dove E' = { (v, w) : (v, w) $\notin$ E}.
- G ha una clique di dimensione k sse G' ha un vertex cover di dimensione |V| - k.



G
Clique = {u, v, x, y}

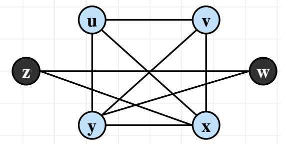


G'
Vertex cover = {w, z}

Clique $\leq_p$ Vertex Cover

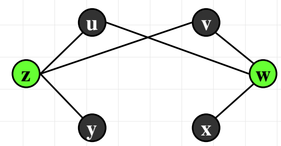
Tesi. VERTEX COVER CLIQUE.

- Dato un grafo non orientato G = (V, E), e il suo complemento G' = (V, E'), dove E' = { (v, w) : (v, w) ∉ E}.
- G ha una clique di dimensione k sse G' ha un vertex cover di dimensione |V|- k.



Prova. ⇒

- Ipotesi: G ha una clique S con |S| = k.
- Considera S' = V – S.
- |S'| = |V| - k.
- Per dimostrare che S' è una cover, considera un qualunque arco (v, w) ∈ E'.
  - (v, w) ∉ E
  - Almeno uno fra v e w non è in S (dato che S è una clique)
  - Almeno uno fra v e w è in S'
  - quindi (v, w) è coperto da S'

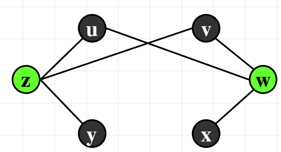


Clique $\leq_p$ Vertex Cover

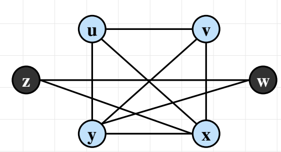
Tesi. VERTEX COVER $\leq_p$ CLIQUE.

- Dato un grafo non orientato G = (V, E), e il suo complemento G' = (V, E'), dove E' = { (v, w) : (v, w) ∉ E}.
- G ha una clique di dimensione k sse G' ha un vertex cover di dimensione |V|- k.

Prova. ⇐



- Ipotesi: G' ha una cover S' con |S'| = |V| - k.
- Considera S = V – S'.
- chiaramente |S| = k.
- Per mostrare che S è una clique, considera un arco (v, w) ∈ E'.
  - se (v, w) ∈ E', allora v ∈ S', e/o w ∈ S',
  - se v ∉ S' e w ∉ S', allora (v, w) ∈ E
  - quindi S è una clique in G



Vertex Cover $\leq_p$ Subset Sum

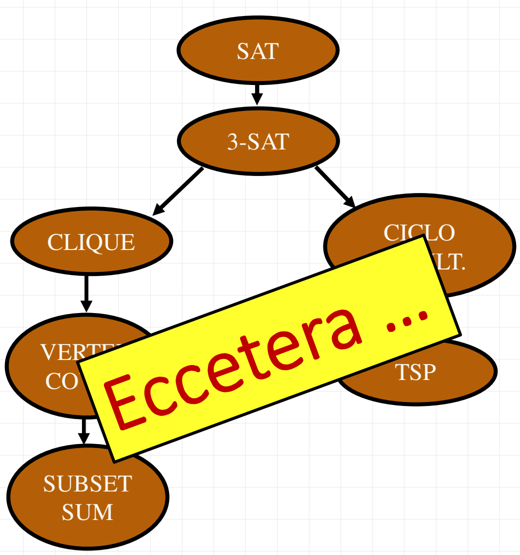
Subset Sum:

Dato un insieme S di numeri e un numero *t*, si vuole determinare se esiste un S'$\subseteq$ S tale che la somma dei numeri in S' sia uguale a *t*.

Dato un grafo G e una opportuna procedura di costruzione di S e *t*, si dimostra che G ha una copertura di ordine k sse $\exists$S'$\subseteq$S di somma *t*.

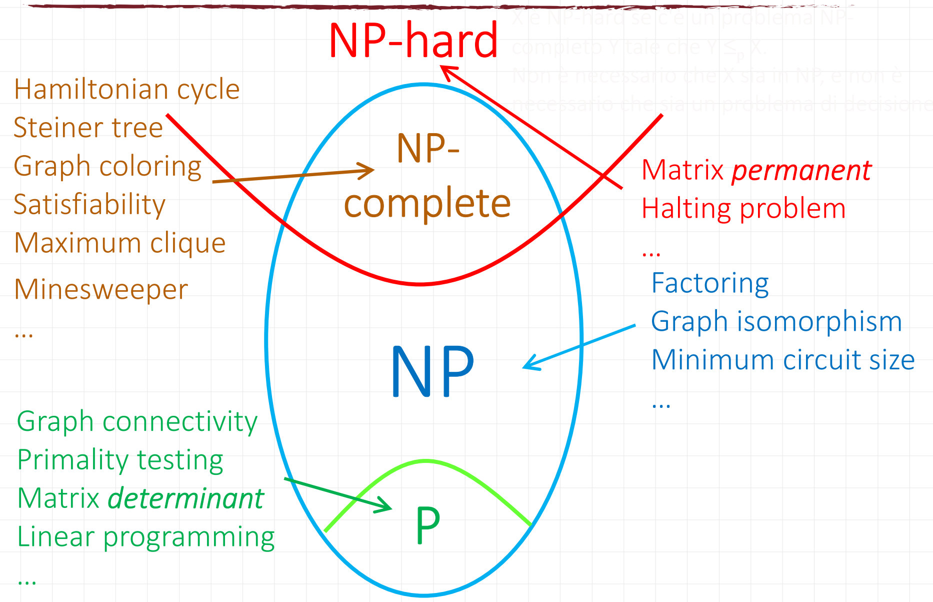
. . .

Albero di riduzioni



Problemi e complessitá

X è NP-hard se c'è un problema NP-completo Y tale che $Y \leq_p X$.
Non è necessario che X sia in NP, e non è necessario che sia un problema di decisione.



33