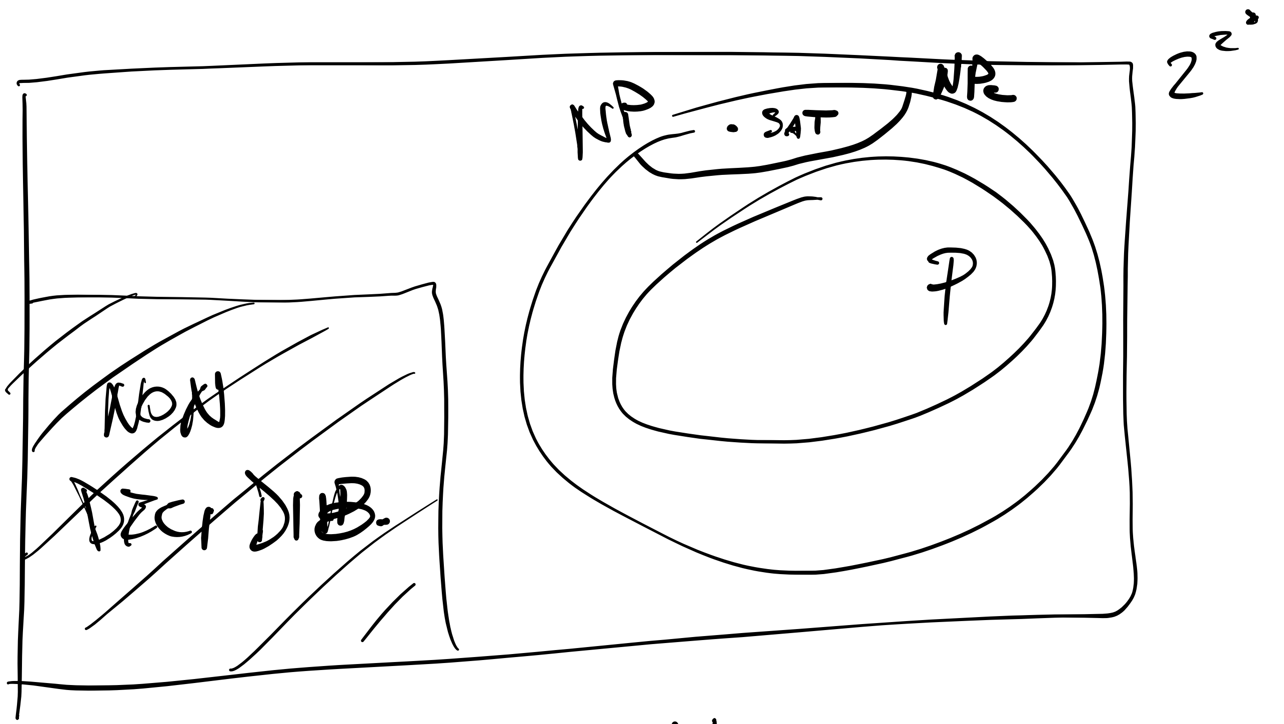
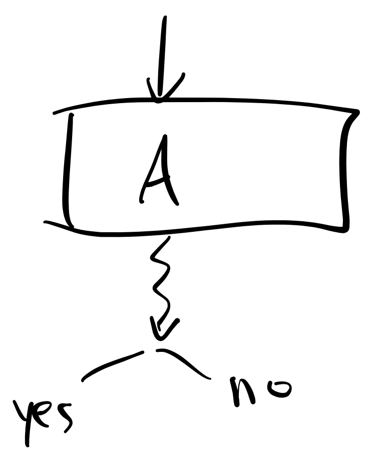


P = insieme dei problemi di decisione che ammettono un algoritmo polinomiale

NP = insieme dei problemi di decisione che ammettono un algoritmo polinomiale NON DET.

Python standard

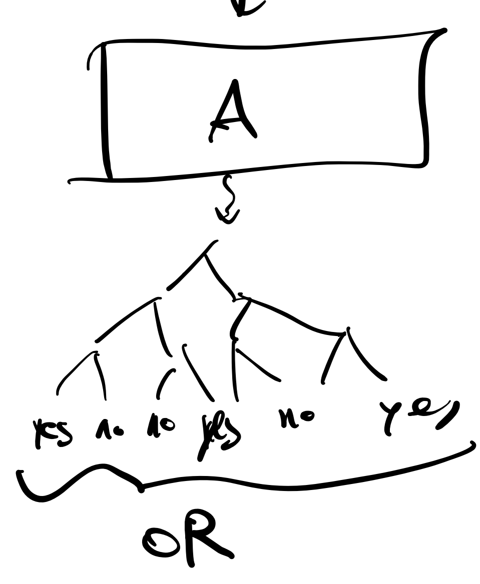
$X \in 2^*$



$X = ?$

Python "magico"

$X \in 2^*$



RIDUZIONE IN TEMPO POLINOMIALE

$\Pi_1, \Pi_2 \subseteq 2^*$ problemi di decisione

Una riduzione polinomiale di Π_1 a Π_2

$$f : 2^* \longrightarrow 2^*$$ calcolabile in tempo polinomiale

$\forall x \in 2^*$

$$x \in \Pi_1 \text{ sse } f(x) \in \Pi_2$$

$$\Pi_1 \leq_p \Pi_2$$

Proprietà: $\Pi_1 \leq_p \Pi_2$ e $\Pi_2 \in P$

$\Rightarrow \Pi_1 \in P$

Def: Π è NP-completo se

1) $\Pi \in NP$
2) $\forall \Pi' \in NP. \quad \Pi' \leq_p \Pi$

Teorema di Cook:

SAT è NP-completo

Teorema: SAT $\in P \iff P = NP$

PROBLEMI DI OTTIMIZZAZIONE

Un p.o. Π

1) insieme di input $I_\Pi \subseteq 2^*$
2) funzione $Amm_\Pi : I_\Pi \to 2^{2^*} \setminus \{\emptyset\}$
   che associa a ogni input x un insieme non-vuoto di soluzioni ammissibili $Amm_\Pi(x)$
3) funzione obiettivo
   $$c_\Pi : 2^* \times 2^* \to \mathbb{R}$$
   che assegna un valore $c_\Pi(x,y)$ $\forall x \in I_\Pi$, $\forall y \in Amm_\Pi(x)$
4) tipo $t_\Pi \in \{MIN, MAX\}$

DATO x volete un $y^* \in Amm_\Pi(x)$ t.c.

$$c_\Pi(x,y^*) \overset{MAX}{\geq} c_\Pi(x,y')$$

$$c^*(x) \underset{MIN}{\leq} \qquad \forall y \in A_{mm_\pi}(x)$$

ESEMPIO

MAX SAT

I_π = formule booleane in CNF

$$X = \underbrace{(x_7 \vee \neg x_2)}_{\text{clausola}} \wedge (x_8 \vee x_3 \vee x_4) \wedge (x_3 \vee \neg x_1)$$

letterali

$A_{mm_\pi}(x)$ = assegnamenti di valori di verità per le variabili che compaiono in X

$c_\pi(x, y)$ = # di clausole di X rese <u>true</u> da Y

t_π = MAX

CLASSE PO

- Un problema di ottimizzazione Π è $\in PO$ sse esiste un algoritmo che lo risolve in tempo polinomiale.

CLASSE NPO

$$\Pi \in NPO$$

a) $I_\Pi \in P$

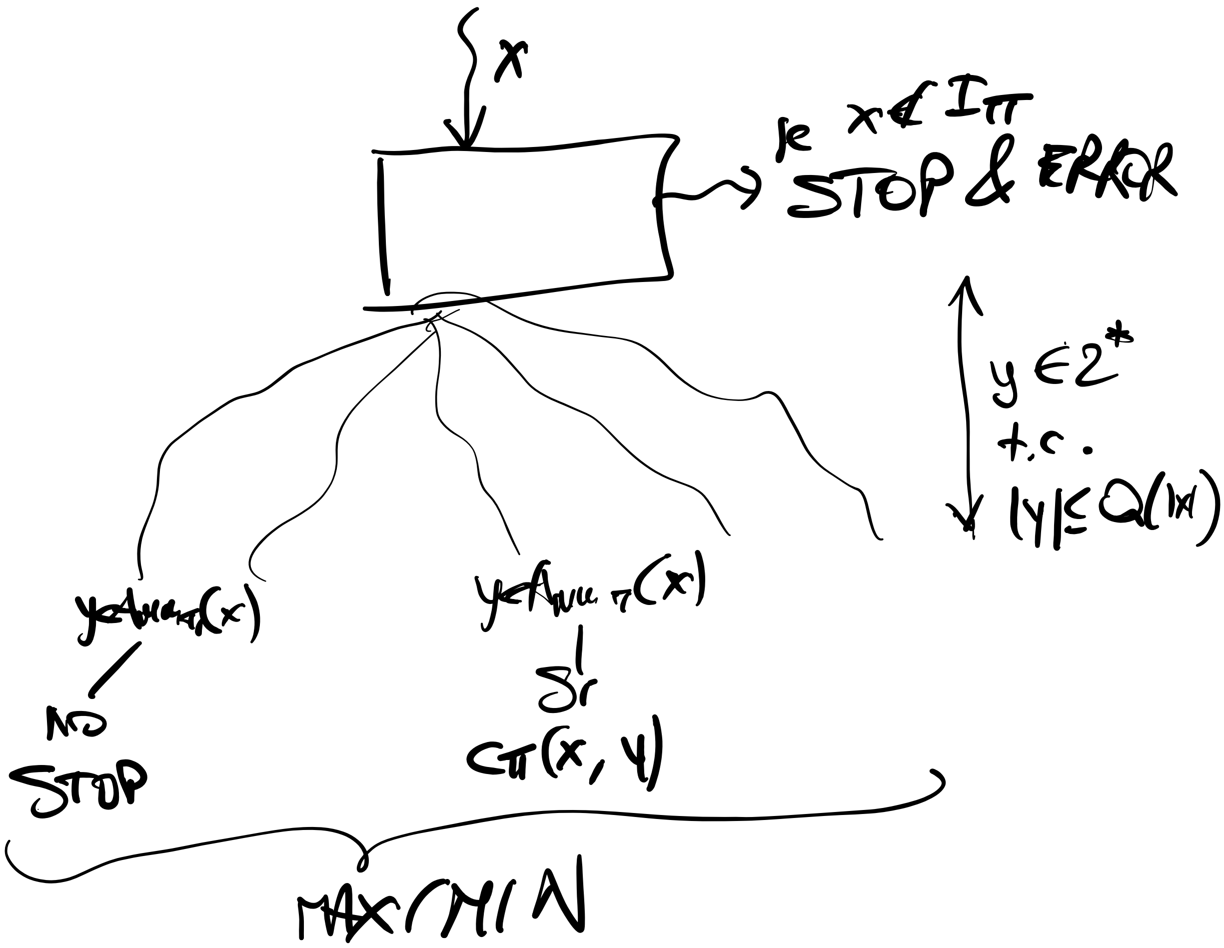
b) esiste un polinomio Q t.c.

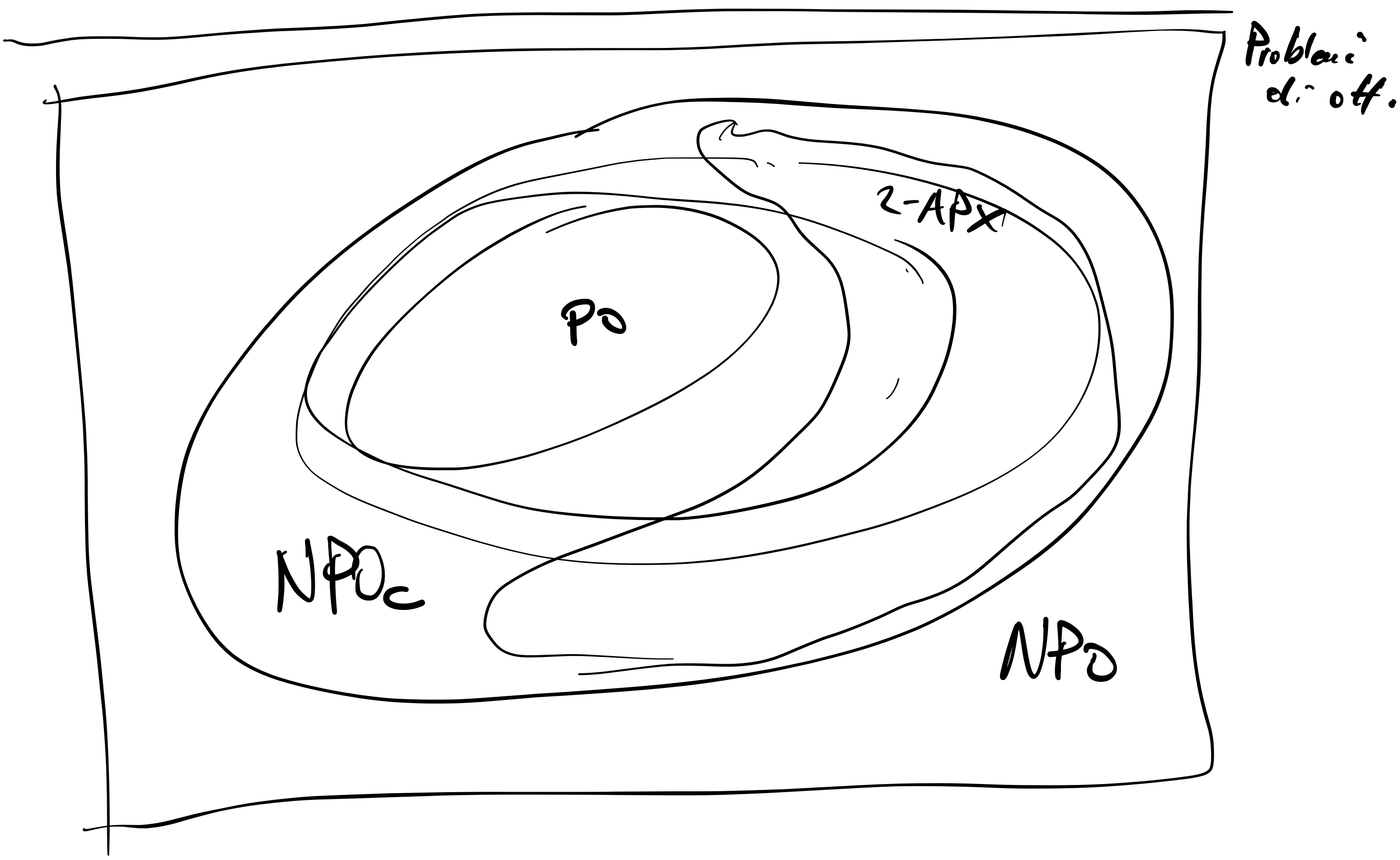
- $\forall x \in I_\Pi \quad \forall y \in Amm_\Pi(x)$

$$|y| \leq Q(|x|)$$

- $\forall x \in I_\Pi \quad \forall y \in \Sigma^*$ se $|y| \leq Q(|x|)$, è decidibile in tempo polinomiale se $y \in Amm_\Pi(x)$

c) c_Π è calcolabile in tempo polinomiale


x
se x ∉ Iπ
STOP & ERROR
y ∈ 2*
t.c.
|y| ≤ Q(|x|)
y ∉ Ammπ(x)
no
STOP
y ∈ Ammπ(x)
sì
cπ(x, y)
MAX/MIN


Problemi di ott.
2-APX
PO
NPOc
NPO

PROBLEMA DI DECISIONE ASSOCIATO A UN PROBLEMA DI OTTIMIZZAZIONE

Π problema ottimizzazione $\longrightarrow$ $\hat{\Pi}$ probl. decisione

$I_{\hat{\Pi}} = I_{\Pi} \times \mathbb{N}$

$(x, k) \in I_{\hat{\Pi}} \longrightarrow$

output yes

se $c^*_{\Pi}(x) \begin{cases} \geq k & t_{\Pi} = MAX \\ \leq k & t_{\Pi} = MIN \end{cases}$

MaxSat

(formula CNF φ, k)

esiste un assegnamento che rende vere $\geq k$ clausole di φ?

Teorema: Se $\pi \in PO$, $\hat{\pi} \in P$.
Se $\pi \in NPO$, $\hat{\pi} \in NP$.

Def. NPO_c è l'insieme dei problemi di ottimizzazione π t.c. $\pi \in NPO$ e $\hat{\pi} \in NP_c$

Teorema: Se $\pi \in NPO_c$, allora $\pi \notin PO$ a meno che $P = NP$

Dim: Spg assumiamo $t_\pi = MAX$.
Per assurdo supponiamo che esista A

$$x \in I_\pi \longrightarrow \boxed{A} \longrightarrow y^* \text{ t.c. } c_\pi(x, y^*) \text{ MAX}$$

che risolve π in tempo polinomiale.

Considerate

decide $\hat{\Pi}$ in tempo polinomiale

INPUT $(x, k) \in I_\pi \times \mathbb{N}$

- Uso A per calcolare y^* e calcolo $c_\pi(x, y^*)$
- Se $c_\pi(x, y^*) \geq k$
  output yes
  altrimenti
  output no

$\hat{\Pi} \in NP$-completo $\Rightarrow$ non si può risolvere $\hat{\Pi}$ in tempo polinomiale a meno che $P = NP$. $\square$

RAPPORTO DI APPROSSIMAZIONE

Π problema ottimizzazione

$$x \in I_\Pi \longrightarrow \boxed{A} \longrightarrow \bar{y} \in Amm_\pi(x)$$

$$R_A(x) = \underline{\max} \left\{ \frac{c_\pi(x, \bar{y})}{c_\pi(x, y^*)}, \frac{c_\pi(x, y^*)}{c_\pi(x, \bar{y})} \right\} \geq 1$$

Si dice che A è una <u>α-approssimazione</u> sse $\forall x \in I_\pi$

$$R_A(x) \leq \alpha$$