Teorema di Savitch - enunciato (semplificato)

Teorema. Per ogni NTM T con complessità di spazio $s(n)$ esiste una TM T' equivalente con complessità di spazio $O(s^2(n))$.

Prova. Sia $T=(Q,\Sigma,\Gamma,\delta,q_0,q_a,q_r)$ una NTM.

Modifichiamo T in modo che abbia un'unica configurazione di accettazione: $c_a = q'_a\perp$. Basta introdurre due delimitatori, sinistro e destro, del nastro e aggiungere un nuovo stato di accettazione, q'_a, che T raggiungerà solo da q_a dopo aver cancellato il contenuto del nastro e riposizionato la testina sulla prima cella.

Ricordiamo che la complessità di spazio, $s(n)$, e la complessità di tempo, $t(n)$, di T soddisfano le seguenti relazioni:

$$n \leq s(n) \text{ e } t(n) \leq 2^{O(s(n))}.$$

Teorema di Savitch - raggiungibilità

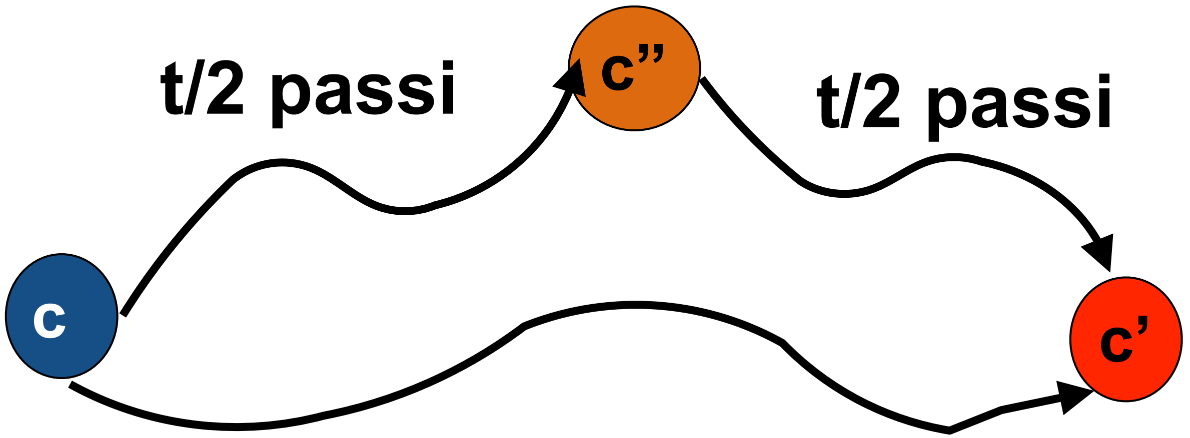
Definiamo la funzione

$$\text{canyield}(c,c',t) = \begin{cases} 1 & \text{se } c \Rightarrow_T^* c' \text{ in al più } t \text{ passi} \\ 0 & \text{altrimenti} \end{cases}$$

$x \in L(T)$ sse $\text{canyield}(c_0,c_a,2^{ks(n)})=1$, dove $c_0 = q_0x$ e k è un'opportuna costante.

D'ora in poi assumiamo che t sia sempre una potenza di 2.

Calcolo ricorso di canyield



t >1 passi

canyield(c,c',t) = 1 sse

canyield(c, c'',t/2) = 1 e canyield(c'',c',t/2) = 1

per una configurazione c''

la TM che calcola CANYIELD

CANYIELD :

sull'input (c,c1,t)

if t=1 then

    if c1 = c then restituisci 1

                else

                if se c porta a c1 in un passo then restituisci 1

                else restituisci 0

else

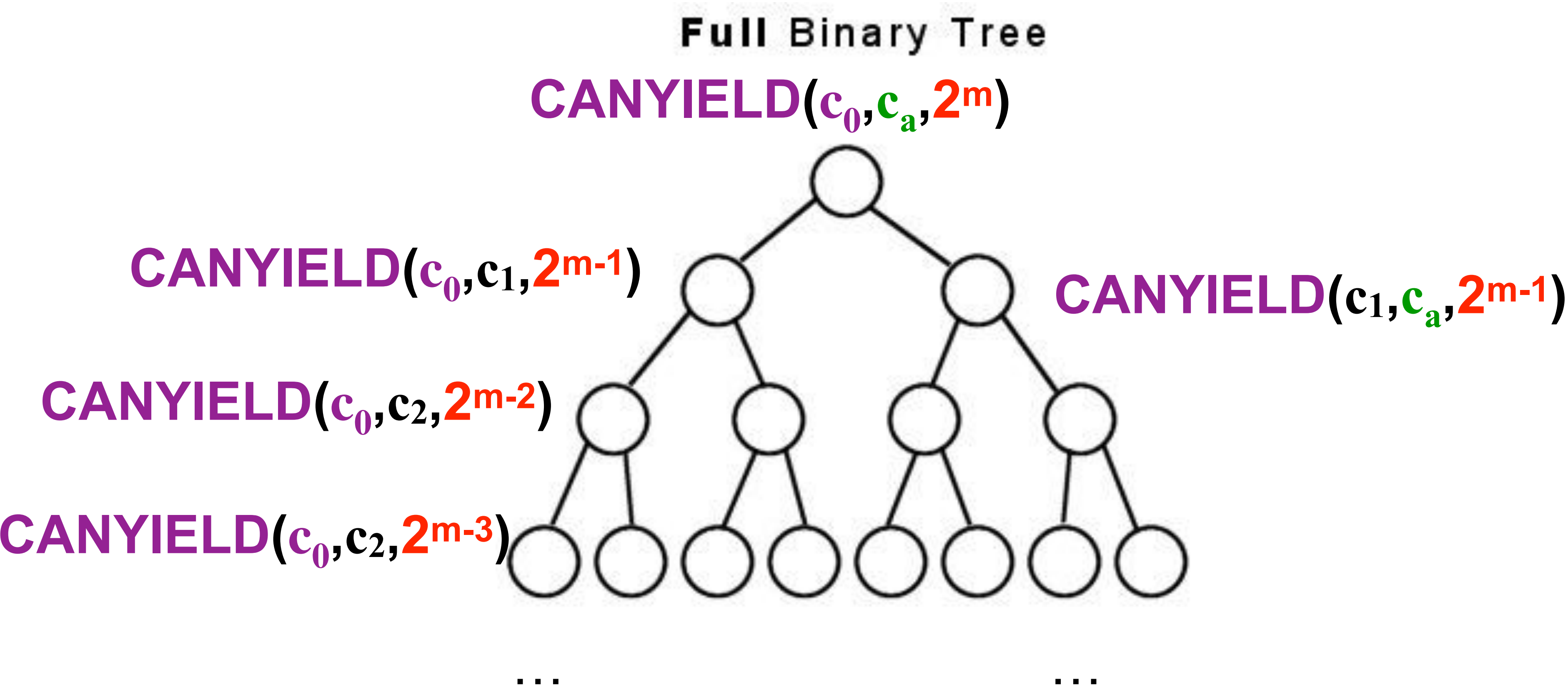
for ogni configurazione c" di lunghezza ≤ s(n) do

    if CANYIELD(c, c",t/2) = 1 and CANYIELD(c",c', t/2) = 1

    then return 1

return 0.

Calcolo ricorsivo di CANYIELD



la TM T’ equivalente alla NTM T

$\boxed{|x| = n}$

T' =

INPUT x

esegui CANYIELD(c_0,c_a,$2^{ks(n)}$) e accetta se restituisce 1, rifiuta altrimenti.

Teorema di Savitch - complessità 1

Il nastro di T' si comporta come uno stack durante il calcolo di CANYIELD(c_1,c_2,2^m) e contiene gli argomenti delle chiamate:

$c_1,c_2,2^m$	$c_1,c_3,2^{m-1}$	$c_1,c_4,2^{m-2}$	…	$c_1,c_{m+2},2^{m-m}$

T' ha complessità di spazio O(s^2(n)). Infatti CANYIELD(c_0,c_a,$2^{ks(n)}$) è calcolata in spazio O(s^2(n)), visto che la profondità della ricorsione è ks(n) e in ogni chiamata lo stack utilizza O(s(n)) spazio.

Teorema di Savitch - complessità 2

Comunque T' ha bisogno di conoscere anche s(n), che dipende da x. Modifichiamo T' per costruire una TM che non utilizza la conoscenza di s(n).
T" semplicemente tenta tutti i possibili valori di s(n)=1,2,3,...
Per ogni valore i,1≤ i ≤ s(n), T" usa CANYIELD per verificare se c_a è raggiungibile e accetta se la risposta è si. Così se T' accetta x in al più j passi allora anche T" accetta x quando tenta s(n) = i per qualche i ≤ j .
Nel caso x non sia accettata bisogna impedire che i sia incrementato indefinitamente, quindi prima di incrementare i a i+1 T" verifica se c'è una configurazione che utilizza i+1 celle di nastro raggiungibile da c_0 .
Se non ce ne sono T" rifiuta. Naturalmente il più grande i per il quale dovrà tentare è s(n). Quindi lo spazio necessario per ricordare l'i corrente è solo log(s(n)) e questo è l'unico spazio in più che T" usa rispetto a T', quindi anche T" lavora in spazio O(s(n)).

Classi di complessità

Ricordiamo che (N)SPACE(f(n)) è la classe dei linguaggi decisi da una (N)TM in spazio O(f(n))

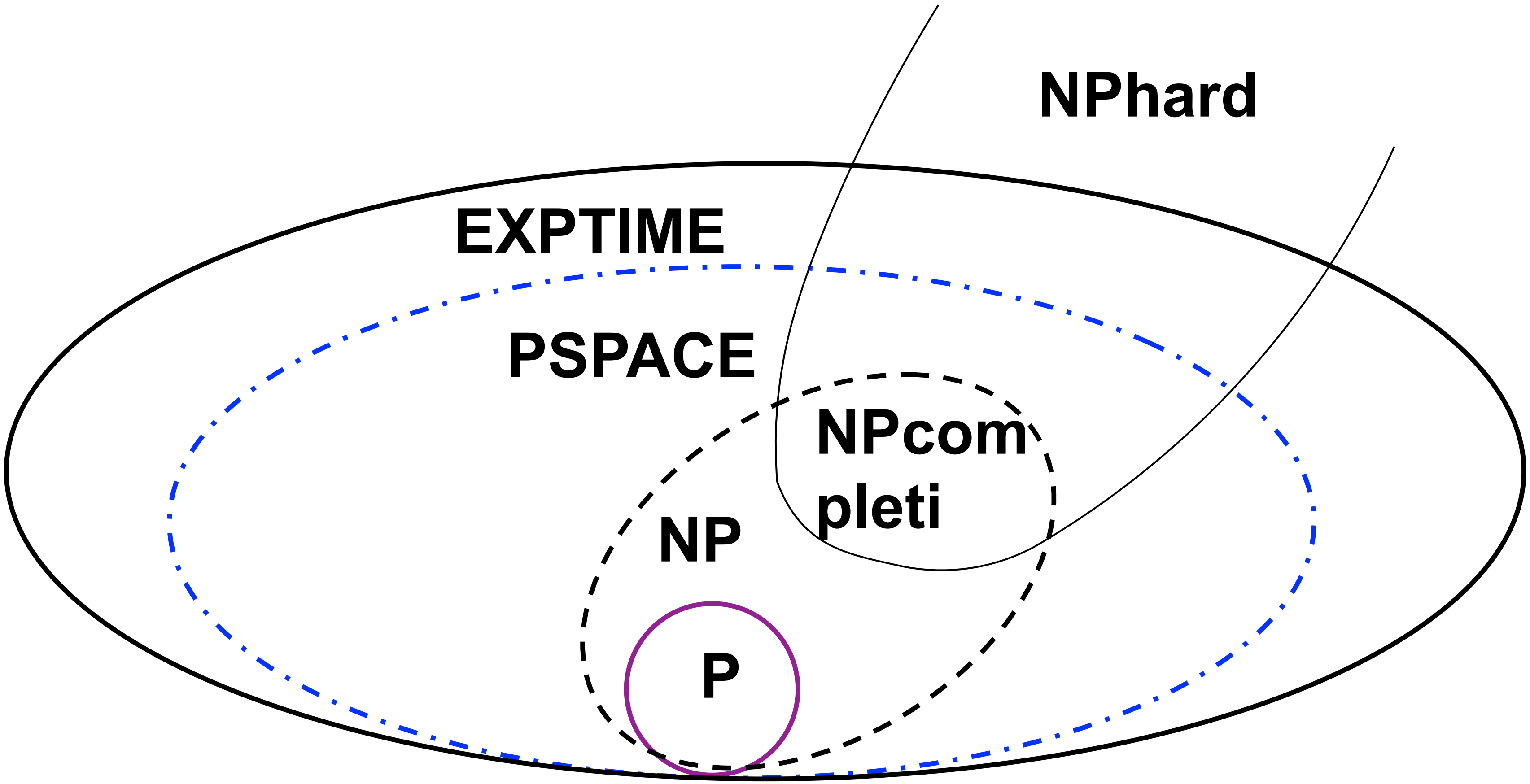
$$\mathbf{PSPACE} = \bigcup_{k\geq 0} \mathbf{SPACE}(n^k)$$

$$\mathbf{NPSPACE} = \bigcup_{k\geq 0} \mathbf{NSPACE}(n^k)$$

$$\mathbf{PSPACE} = \mathbf{NPSPACE}$$

$$\mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$$

Tra P e EXPTIME?



Si sa solo che P ≠ EXPTIME

- **E' vero che se B è in SPACE(n^2) e A $\leq_p$ B allora anche A è SPACE(n^2)?**

- **E' vero che NTIME(n^3) $\subseteq$ SPACE(n^3)**

- **E' vero che NTIME(n^2) $\subseteq$ TIME($2^{O(n^2)}$)**
    a. **Sì, perchè ……**
    b. **No, possiamo dire che NTIME(n^2) $\subseteq$ TIME(….) perchè ….**

- **Se L1 $\leq_p$ L2 e L2 $\leq_p$ L1 e L2 è NP-completo allora L1 è NP-completo.**

- **Se L1 ,L2 e L3 sono linguaggi su {0,1}*, diversi dal linguaggio vuoto e da {0,1}*, si dimostri che se L1 $\leq_p$ L2 $\cap$L3, L2 è in NP e L3 è in P, allora L1 è in NP.**

Esercizio NPcompleto 1

Si consideri il seguente problema
SUBGRAPH_ISO = {<G,H> | G e H sono grafi non diretti e G contiene un sottografo isomorfo ad H }

Ricordiamo che un grafo G=(V,E) è isomorfo a un grafo G'=(V',E') se esiste una biezione f : V -> V' tale che {a,b} è un arco di G sse {f(a),f(b)} è un arco di G'.
Si dimostri che SUBGRAPH_ISO è NP-completo.

Suggerimento: per la riduzione si consideri il problema del CLIQUE.

Soluzione 1

Dimostriamo che CLIQUE $\le_p$ SUBGRAPH_ISO.

Si deve associare a <G,m> una coppia <G_1,H> in modo tale che
G ha un m-clique sse G_1 ha un sotto grafo isomorfo ad H.

Prendiamo G_1=G e H =K_m, dove K_m è il grafo completo di m vertici, in altre parole consideriamo la funzione f : <G,m> $\mapsto$ <G,K_m> e dimostriamo che si tratta di una riduzione.

Se G ha un m-clique allora questo è isomorfo a K_m, d'altra parte se G non ha un k clique allora non ha un sottografo isomorfo a K_m. Quindi f associa istanze sì a istanze sì e istanze no a istanze no.

Esercizio NPcompleto 2

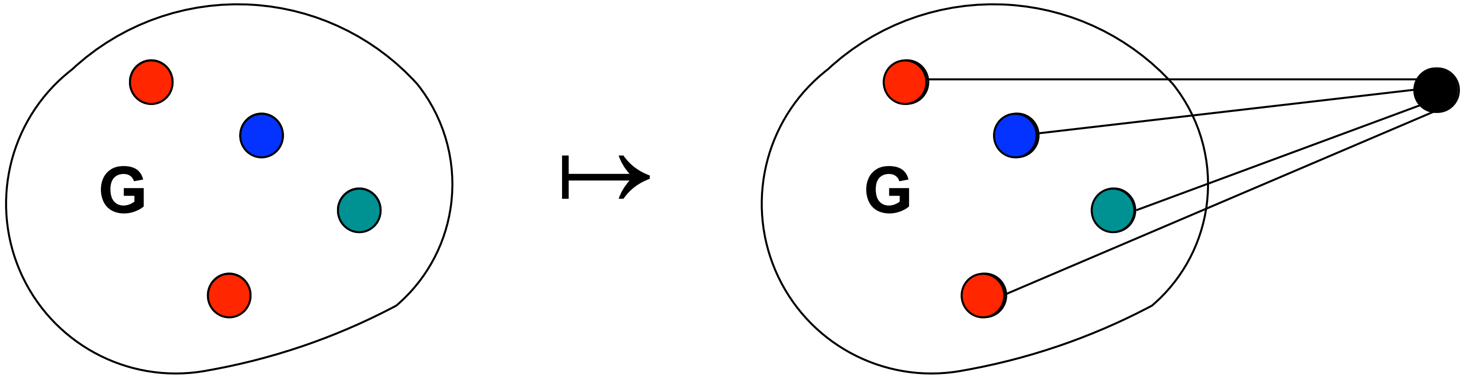
Sia k-COLOR = { G | G è un grafo non diretto ed è possibile colorare i vertici di G con al più k colori in modo tale che due vertici adiacenti non abbiano lo stesso colore}

Si definisca una riduzione polinomiale da 3-COLOR a 4-COLOR.

Soluzione 2

Dimostriamo che 3-COLOR $\leq_p$ 4-COLOR.

Si deve associare a <$G=(V_G,E_G)$> un grafo <$H=(V_H,E_H)$> in modo tale che G è colorabile con 3 colori sse H lo è con 4.



Costruiamo H: $V_H = V_G \cup \{w\}$, dove w è un nuovo vertice, e $E_H = E_G \cup \{\{v,w\} \mid v \text{ è in } V_G\}$

3-colorabile $\Leftrightarrow$ 4 colorabile

Esercizio NPcompleto 3

Sia MAXCLIQUE = {(G, k) | G è un grafo il cui clique massimale ha dimensione k}, per cui notMAXCLIQUE = {(G, k) | G è un grafo il cui clique massimale non ha dimensione k}
Si dimostri notMAXCLIQUE è NP-hard, con la riduzione: CLIQUE $\leq_p$ notMAXCLIQUE.

Esercizio NPcompleto 3

Sia MAXCLIQUE = {(G, k) | G è un grafo il cui clique massimale ha dimensione k}, per cui notMAXCLIQUE = {(G, k) | G è un grafo il cui clique massimale non ha dimensione k}

Dimostriamo che CLIQUE $\leq_p$ notMAXCLIQUE.

A <G,k>, istanza per CLIQUE, associamo <G',k-1>, istanza per notMAXCLIQUE, dove G' è G unito a un grafo completo di k-1 vertici.
Se G ha un k-clique, allora <G',k-1> è un'istanza sì di notMAXCLIQUE, perché in G' c'è un clique di k vertici.
Se invece G non ha un k-clique, allora <G',k-1> è un'istanza sì di MAXCLIQUE, e quindi un'istanza no di notMAXCLIQUE.