

13/12/2024

Teo: TQBF e' Pspace hard

Dim: devo mostrare che ogni problema in Pspace si risolve in tempo ^{garantisce anche spazio polinomiale} $\text{poly}(n)$ e TQBF, lo dim merai idee già unite.

Dimostreremo che esiste una TM M che decide A (linguaggio generico in PSPACE) in spazio $O(n^2)$ per qualche a , mostreremo costruire $R: \{0,1\}^* \rightarrow \{0,1\}^*$ che restituisce una TQBF ϕ_M tale che $x \in A \Leftrightarrow M(x) = \text{accept} \Leftrightarrow R(x) = \phi_M$ e' vera.

Precisazione: Ci sono formulazioni equivalenti di una TQBF,

- Invece che \exists, \forall sono alternati. Se non lo fosse si poteva esprimere $\leq n$ DUMMY VARIABLES.
- Permettere nelle formule " \rightarrow " e " \leftrightarrow "; si possono ~~trasformare~~ trasformare in CNF

- permettere che i quantificatori non siano tutti all'inizio.

$$\exists x_1 \forall x_2 [x_1 \rightarrow x_2] \wedge \dots$$

Sono modi equivalenti di definire un TQBF.
Come poniamo oltre di $M(x)$? Sappiamo che
ha $2^{O(n^2)}$ configurazioni (SPAZIO LIMITA TEMPO)
Uno di queste serve C_{start} , la configura-
zione iniziale $q_0 \sqcup 1$.

Poniamo assumere senza perdita di generalità
che esiste UNICA configurazione accettata
 C_{acc} . Il fatto che $M(x)$ accetta corrisponde
al fatto che \exists cammino $C_{start} \rightarrow C_{acc}$
nel grafo $G_{M,x}$ definito come nel Teorema
di Savitch.

Oss: Il numero di nodi è esponenziale,
ricorda M è deterministica, una configurazione
porta ad un'altra. La relazione $R(x)$
dove ϕ_M tale che ϕ_M è soddisfacibile
 $\Leftrightarrow \exists$ cammino $C_{start} \rightarrow C_{acc}$. Basta che
 ϕ_M sia calcolabile in tempo polinomiale,
questo fatto serve immediatamente.

Idea 1) potremmo oltre che ϕ_M e del

Tipo: $\phi_H = \exists c_1 \exists c_2 \dots \exists c_l$ t.c.

$c_1 = c_{start}$, $c_l = c_{acc}$ ed i nodi intermedi sono quelli del cammino $c_{start} \rightarrow c_{acc}$.
Le c_i sono le config. codificabili con $O(n^2)$ bit, enumerando queste $2^{O(n^2)}$.

Fin in dettaglio procediamo come in Cook-Levin, definiamo una sottoformula

$$\phi_H = \phi_{fields}(c_{start}, c_2) \wedge \phi_{fields}(c_2, c_3) \dots \wedge \phi_{fields}(c_{l-1}, c_{acc})$$

ϕ_{fields} è lo stesso sinte nelle dimo. di

Cook-Levin. $\phi_{fields}(c_i, c_j) = \text{True}$ se c_j segue a c_i in accordo con le regole della 8.

Problema: $l \leq \text{numeri conf} = 2^{O(n^2)}$ e quindi in spazio esponenziale.

Idea 2) Allora l'algo. ricorsivo del Teo. di Savitch. Ovvero, costruiamo $\phi_K(c_0, c_1)$ che è vero \iff c'è un cammino da c_0 a c_1 di lunghezza 2^K .

fatto questo, $R(\infty)$ ritorna l'output di $\phi_{O(n^2)}(C_{start}, C_{accept})$

$\mathbb{L} \Rightarrow \text{Vera} \Leftrightarrow \exists$ cammino di lunghezza $2^{O(n^2)}$ $C_{start} \rightarrow C_{acc}$

- Caso base ($k=0$): $\phi_0(C_0, C_1) =$

$\phi_{\text{yields}}(C_0, C_1) \Rightarrow C_1$ può risultare da C_0
oppure $C_0 = C_1$

- Caso induttivo: $\phi_k(C_0, C_1) =$
 $\exists C_{mid}$ per cui

$$\phi_{k-1}(C_0, C_{mid}) \wedge \phi_{k-1}(C_{mid}, C_1)$$

Questo approccio **non va bene**: dato che la nostra ϕ_k ha dimensione

$$|\phi_k| = O(n^2) + 2|\phi_{k-1}|$$

$$= O(n^k \cdot n^2) \text{ raddoppio ad ogni passo}$$

Idea finale) Stena (idea 2) ma scritta in maniera compatta, definiremo

$\phi_k(C_0, C_1) = \exists C_{mid} \text{ t.c. } \forall D, D' \text{ la sequenza è vera}$

$$\left(\begin{array}{c} (D, D') = (C_0, C_{mid}) \\ \vee \\ (D, D') = (C_{mid}, C_1) \end{array} \right) \Rightarrow \phi_{k-1}(D, D')$$

Espressione equivalente che richiama ϕ_{k-1} solo una volta, quindi per le dimensioni:

$$|\phi_k| = O(n^2) + |\phi_{k-1}|$$

$$\Rightarrow |\phi_k| = O(k \cdot n^2)$$

$$\Rightarrow |\phi_{O(n^2)}| = O(n^{2^2}) \Rightarrow \text{polinomiale.} \quad \blacksquare$$

Concludiamo con la complessità con un caso più generale. Il Teorema di

Sapierow che $NSPACE = CONSPACE = PSPACE$, la versione "reale" è $NLOG = CONLOG$.

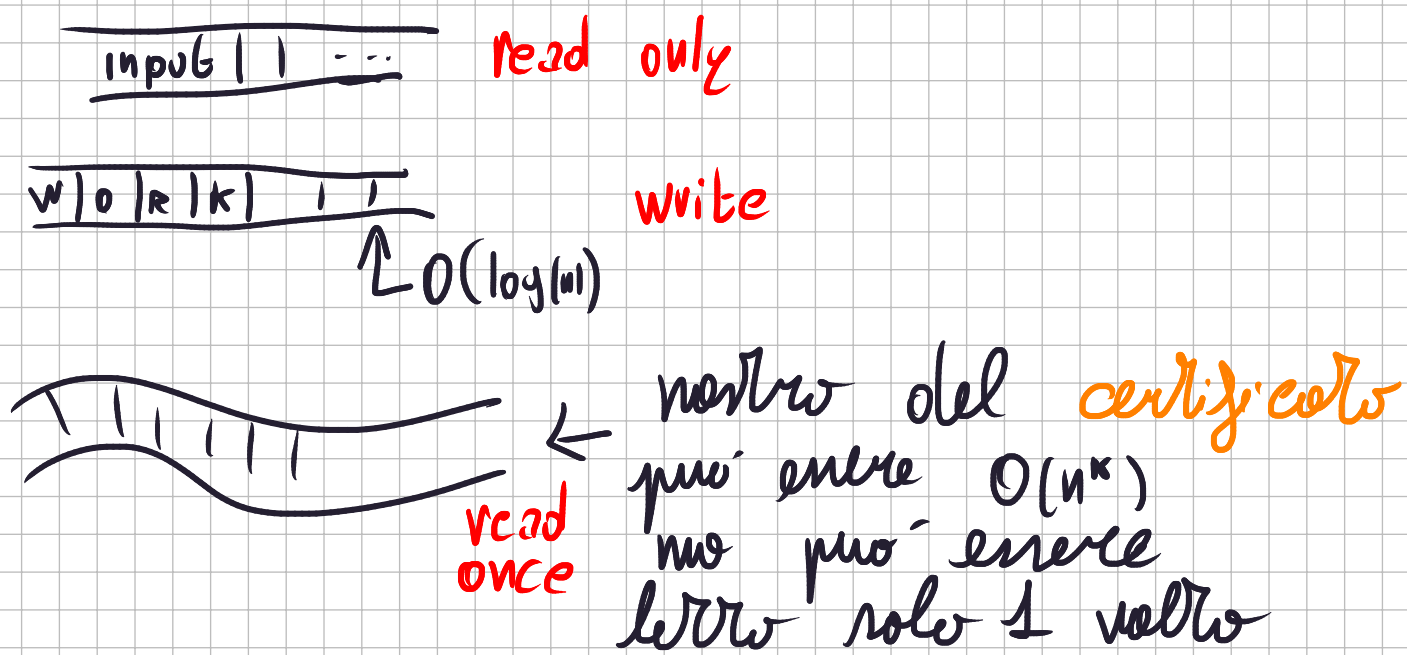
Teo (Immerman-Szelepcsenyi): $NLOG = CONLOG$

Dim: basta dimostrare che $PATH \in CONLOG$.

Questo è vero se $\overline{PATH} \in NLOG$, sappiamo che $\forall A \in NLOG, A \leq_m \overline{PATH}$, se $PATH \in CONLOG$ allora $NLOG \subseteq CONLOG$, d'altra parte

se $B \leq_n^L \text{PATH} \Rightarrow \overline{B} \leq_n^L \text{PATH}$ e se $\text{PATH} \in \text{NLOG}$
 $\exists \text{NTM}$ che decide \overline{B} in spazio $O(\log(n))$,
 quindi $\text{CONLOG} \subseteq \text{NLOG}$. Questo sembra
 sorprendente: esiste un certificato polinomiale
 del fatto che non esiste un cammino da
 $s \rightarrow t$ che può essere verificato in
 spazio $O(\log(n))$.

Verificatore per NLOG:



Questo def. è equivalente a quello che uno
 il non determinismo.

Verifichiamo il certificato

Input: grafo (G, s, t)

dato G , siano R_ℓ i vertici raggiungibili
da s in al più ℓ passi, e definiamo
 $r_\ell = |R_\ell|$

Ad esempio, $R_0 = \{s\}$, raggiunge al più
se stesso $\Rightarrow r_0 = 1$

Il certificato è diviso in più parti:
certificato per r_1 ; per $r_2 \dots$ per r_n ,
per $s \rightarrow t$ (non c'è il cammino).

Idea) Una volta che V (verificatore) ha
verificato il cert. per r_ℓ , deve leggere
solo gli ℓ, r_ℓ sul nastro gli ha
per verificare il cert. di $r_{\ell+1}$.

Ma come sono fatti gli certificati?

Supponiamo che V sia convinto di r_n , e
lo abbia verificato. $r_n =$ numero vertici

raggiungibili da s in n passi \Rightarrow dim.

massimo di passi per un cammino $= n$, il

certificato per (non esiste cammino $s \rightarrow t$)

non è una cosa del tipo

$S \rightsquigarrow v_2; S \rightsquigarrow v_{s_1} \dots;$

\hookrightarrow sono in numero r_n , ma verificato che t non sia fra questi.

Il verificatore controlla che ogni cammino è in G (numero $O(\log(n))$ spaziale) per controllare che ci sono r_n cammini; contenuto in $O(\log(n))$ spaziale, e controllo che t non è nodo finale dei cammini. Non devono essere cammini ripetuti, quindi.

Si assume che nel certificato i nodi ^{endpoint} siano in ordine lexicografico. Così è possibile controllare che non ci siano ripetizioni salvando 1 solo vertice.

Per il resto vediamo il certificato per r_{l+1} numero che è verificato r_l .

Il certificato sarà fatto così:

$v_1 \in R_{l+1}$ perché

$v_2 \notin R_{l+1}$ perché

$v_n \in R_{l+1}$ perché...

Saremo capaci di n piccoli certificati che certifichino se un vertice v o non v è raggiungibile.

In $O(\log(n))$ spazio posso costruire $V_{l+1} = |R_{l+1}|$
Resta da specificare come faccio a certifi-
care uno cone del tipo

$$v_i \in R_{l+1} \dots$$

Il certificato v è un cammino $S \rightarrow v_i$ in
al più $l+1$ passi. Per fare ciò V controlla
che il cammino sia valido e di lunghezza
corretta, usando $O(\log(n))$ spazio.

$$v_i \notin R_{l+1}$$

per verificare ciò, sappiamo già (per ipotesi)
che $v_i \notin R_l$, il certificato ricorrendo a V
tutti i vertici in R_l , può essere verificato
usando solo v_n usando $O(\log(n))$ spazio. ■