Teo TQBF è PSPACE-hard.

Dim. Devo mostrare che $\forall A \in PSPACE$ allora $A \leq_m^p TQBF$. La dimostrazione usa molte quelle viste nel teorema di Cook-Levin e Savitch.

Ha essenzialmente una TM M che decide A in spazio $O(n^a)$ per qualche a. Dobbiamo costruire una riduzione $R: \{0,1\}^* \to \{0,1\}^*$ che restituisce una TQBF φ_M t.c.

$x \in A$ sse $M(x) = ACC$ sse $R(x)$ è VERA

Una precisazione sulle TQBF; ci sono formule

Vorrei eliminare:

- Invertire che $\exists, \forall$ sono alternati. Se non le posso più portare all'inizio ≤ n DUMMY VARIABLES.

- Permettere nelle formule "$\rightarrow$" e "$\leftrightarrow$"; si possono trasformare in CNF.

- Permettere che i quantificatori non siano tutti all'inizio

$$\exists x_1 \forall x_2 \, [x_1 \rightarrow x_2) \wedge (\exists x_3 \ldots$$

Cosa possiamo dire di $M(x)$? Sappiamo che ha $2^{O(n^a)}$ configurazioni; una di queste sarà $C_{start} = \underline{q_0}\sqcup$; 1 e poi possiamo assumere WLOG che ci sia una unica configurazione accettante C_{Acc}. In questo modo, il fatto che $M(x)$ accetta corrisponde al fatto che $C_{start} \rightsquigarrow C_{Acc}$ nel grafo $G_{M,x}$.

Osservazioni: # nodi è esponenziale; siccome M è deterministica una configurazione porta solamente ad un'altra configurazione.

La riduzione $R(x)$ darà in output una

TQBF φ t.c. φ è soddisfacibile sse
$\exists$ $C_{start} \rightsquigarrow C_{acc}$ in $G_{\varphi, n}$.

Il punto cruciale è assicurarsi che $|\varphi| \leq$ poly (n); il fatto che φ sia calcolabile in tempo polinomiale sarà immediato.

Vediamo una prima idea di partenza:

$\varphi = \exists C_1 \exists C_2 \cdots \exists C_\ell$ t.c. $C_1 = C_{start}$,
$C_\ell = C_{acc}$ e le altre C_i sono i nodi del cammino $C_{start} \rightsquigarrow C_{acc}$. Qui le C_i sono le configurazioni, codificabili con $O(n^2)$ bit.

Più un dettaglio, per fare questo procederemo come un Cook - Levin:

$$\varphi: \varphi_{YIELDS}(C_{START}, C_2) \wedge \varphi_{YIELDS}(C_2, C_3)$$
$$\dots \wedge \varphi_{YIELDS}(C_{\ell-1}, C_{ACC})$$

La formula φ_{YIELD} è la stessa vista nella prova di Cook - Levin, problema: $\ell \leq \#$ configurazioni che è $2^{O(n^a)}$ spazio esponenziale.

Secondo noi: Usare l'algoritmo ricorsivo del Teorema di Savitch. Ovvero costruiamo $\psi_k(C_0, C_1)$ che è vera sse $C_0 \Rightarrow C_1$ di lunghezza 2^k. Fatto questo, R(x) restituisce

$$\psi_{O(n^e)}(C_{start}, C_{acc}).$$

- Caso base: $\psi_0(C_0, C_1) =$

$$\psi_{yields}(C_0, C_1) \vee (C_0 = C_1)$$

- Caso induttivo: $\psi_k(C_0, C_1) =$

$$\exists\, C_{MID} \quad \varphi_{k-1}(C_0, C_{MID}) \wedge$$

$$\varphi_{k-1}(C_{MID}, C_1)$$

Ancora non va bene. Questo perché:

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

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

Infine finale:

$$\varphi_k(C_0, C_1) = \exists\, C_{MID} \;\forall\, D, D'$$

$$\left(\begin{array}{c} (D, D') = (C_0, C_{M|D}) \\ \vee \\ (D, D') = (C_{M|D}, C_1) \end{array} \right) \longrightarrow \varphi_{k-1}(D, D')$$

$$|\varphi_k| = O(M^e) + |\varphi_{k-1}|$$

$$\Rightarrow |\varphi_k| = O(k \cdot M^e)$$

$$\Rightarrow |\varphi_{O(M^e)}| = O(M^{2e}) \qquad \blacksquare$$

Concludiamo sulla complessità di spazio con un risultato classico: Teorema di IMMERMAN - SZELEPCSENYI.

Supponiamo che: $NPSPACE = coNPSPACE = PSPACE$.

La nostra scelta sarebbe $NL = coNL$.

Se credevo falso, ma nel '88 è stato dimostrato vero.

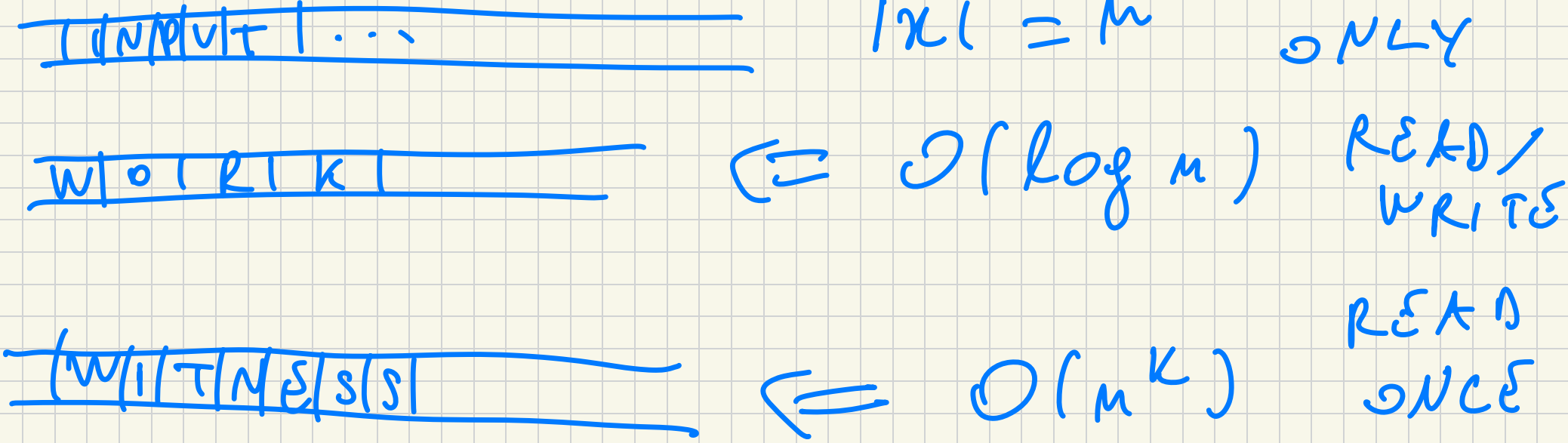
Teo $NL = coNL$.

DIM. Basta mostrare che $PATH \in coNL$.

Questo perché: $\overline{PATH} \in NL$ ovvero $\overline{PATH} \in coNL$; questo ci dice $NL \subseteq coNL$. D'altra parte $B \leq_m^L PATH \Rightarrow \overline{B} \leq_m^L \overline{PATH}$ e

Se $\overline{PATH} \in NL$, $\exists$ NTM che decide $\overline{B}$ in log spazio, ovvero $coNL \subseteq NL$.
Questo sembra sorprendente: esiste un certificato polinomiale dal fatto che $s \not\to t$ che può essere verificato in $\log n$ spazio.

Verificatore per NL:



L'input: $\langle G, s, t \rangle$

Notazione: Dato G, siamo R_ℓ i vertici raggiungibili da s in $\leq \ell$ passi. Sia anche $r_\ell = \# R_\ell$.

Ad es. $R_0 = \{s\}$ e $r_0 = 1$. Com'è fatto il certificato:

certificato per r_1 ; certificato per r_2 ; ...

certificato per r_n ; certificato per $s \rightsquigarrow t$

Idea: Una volta V ha verificato il cert. per r_ℓ, avrei bisogno solo di ℓ, r_ℓ sul nastro di lavoro per verificare cert. per $r_{\ell+1}$

Partiamo dalla fine: Supponiamo che V è convesso da π_M (# vertici raggiungibili da s). A questo punto il certificato per $s \not\rightsquigarrow t$:

$$s \rightsquigarrow v_2 \; ; \; s \rightsquigarrow v_5 \; ; \; \dots \; ; \; s \rightsquigarrow v_7$$

$$\hookrightarrow \# \pi_M$$

Come controlla il verificatore? Che ogni cammino è in G usando $O(\log n)$ spazio; controlla che ci sono π_M cammini controllando in $O(\log n)$ spazio; t non è nodo finale.

L'unica soluzione: Non dobbiamo ritrovare
comunque rappresentanti. Per evitare questo
assumiamo un ordine lessicografico degli
end points.
Per il resto, vediamo il conteggio per $r_{\ell+1}$
dato che abbiamo già visto r_ℓ. Sarà
fatto così:

$v_1 \in R_{\ell+1}$ perché ...; $v_2 \notin R_{\ell+1}$ perché;

.... $v_M \in R_{\ell+1}$ perché ...

$\Rightarrow$ In $O(\log M)$ sposto contare $r_{\ell+1} = \#\ell_{\ell+1}$

Resta da specificare, ad es. il certificato per $v_8 \in R_{l+1}$ e $v_8 \notin R_{l+1}$

$v_8 \in R_{l+1}$: mio comunico $s \rightsquigarrow v_8$ un $\leq l+1$ passo. V controlla che il comunico sia valido e in lunghezza corretta controllando in $O(\log n)$ spazio.

$v_8 \notin R_{l+1}$: il certificato records a V tutti i vertici in R_l:

$s \overset{\leq l}{\rightsquigarrow} v_1$; $s \overset{\leq l}{\rightsquigarrow} v_7$; ...

Anche questo può essere verificato mediante
lo stesso T_M e usando $O(\log n)$ spazio
come sopra. ∎