Computabilità, Complessità e Logica

Lezione 17

Spazio Polinomiale

- Ricordiamo che $\mathrm{SPACE}(f(n))$ indica i linguaggi decisi da macchine di Turing deterministiche che lavorano in **spazio** $O(f(n))$
- E che $\mathrm{NSPACE}(f(n))$ indica i linguaggi decisi da macchine di Turing non deterministiche che lavorano in **spazio** $O(f(n))$
- In particolare abbiamo:
  - PSPACE. Linguaggi riconosciuti da MdT deterministiche che lavorano in spazio polinomiale
  - $\mathrm{NPSPACE}$. Linguaggi riconosciuti da MdT non deterministiche che lavorano in spazio polinomiale

Spazio Polinomiale: Osservazioni

- Se abbiamo una macchina deterministica $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ che lavora in spazio $s(n)$ possiamo fare alcune considerazioni:
  - Il numero di configurazioni in cui la macchina può essere è finito e limitato dal prodotto di:
    - $|Q|$ numero degli stati
    - $s(n)$ posizioni della testina
    - $|\Gamma|^{s(n)}$ numero di parole che possono essere scritte sul nastro

Spazio Polinomiale: Osservazioni

- Quindi la MdT M su input w con $|w| = n$ può avere al più $|Q||\Gamma|^{s(n)}s(n)$ configurazioni distinte
- Se una configurazione si ripete due volte, dato che la macchine è deterministica, sappiamo che non si arresterà mai
- Quindi se una MdT che lavora in spazio $s(n)$ non si è arrestata dopo $|Q||\Gamma|^{s(n)}s(n)$ passi allora non si arresterà mai

Spazio Polinomiale: Osservazioni

E tempo esponenziale

Dipende dalla dimensione dell'input

$$|Q||\Gamma|^{s(n)}s(n)$$

Costante

Possiamo mostrare qualche relazione tra PSPACE e i linguaggi definibili in tempo esponenziale?

EXPTIME

- Ricordiamo le definizione di $\mathrm{EXPTIME}$ e $\mathrm{NEXPTIME}$
- La classe $\mathrm{EXPTIME}$ è la classe dei linguaggi decisi da MdT deterministiche in tempo esponenziale

$$\mathrm{EXPTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{TIME}(2^{n^k})$$

- La classe $\mathrm{NEXPTIME}$ è la classe dei linguaggi decisi da MdT non-deterministiche in tempo esponenziale

$$\mathrm{NEXPTIME} = \bigcup_{k \in \mathbb{N}} \mathrm{NTIME}(2^{n^k})$$

Spazio Polinomiale: Osservazioni

- Iniziamo a riscrivere $|Q||\Gamma|^{s(n)}s(n)$
- Dato che $s(n) = |\Gamma|^{\log_{|\Gamma|} s(n)}$ abbiamo:
- $|Q||\Gamma|^{s(n)}s(n) = |Q||\Gamma|^{s(n)}|\Gamma|^{\log_{|\Gamma|} s(n)}$
- Per le proprietà delle potenze: $|Q||\Gamma|^{s(n)+\log_{|\Gamma|} s(n)}$
- In modo simile: $|\Gamma|^{s(n)+\log_{|\Gamma|} s(n)+\log_{|\Gamma|}|Q|}$
- Cambiando la base dell'esponente:
  $|\Gamma|^{s(n)+\log_{|\Gamma|} s(n)+\log_{|\Gamma|}|Q|} = 2^{(s(n)+\log_{|\Gamma|} s(n)+\log_{|\Gamma|}|Q|)\log_2(|\Gamma|)}$

Spazio Polinomiale: Osservazioni

- Usando la notazione O-grande all'esponente: $2^{(s(n)+\log_{|\Gamma|} s(n)+\log_{|\Gamma|}|Q|)\log_2(|\Gamma|)}$ diventa $2^{O(s(n))}$

- Se $s(n)$ è un polinomio allora abbiamo che una MdT deterministica che riconosce il linguaggio L in spazio $s(n)$ richiede tempo al più $2^{O(s(n))}$

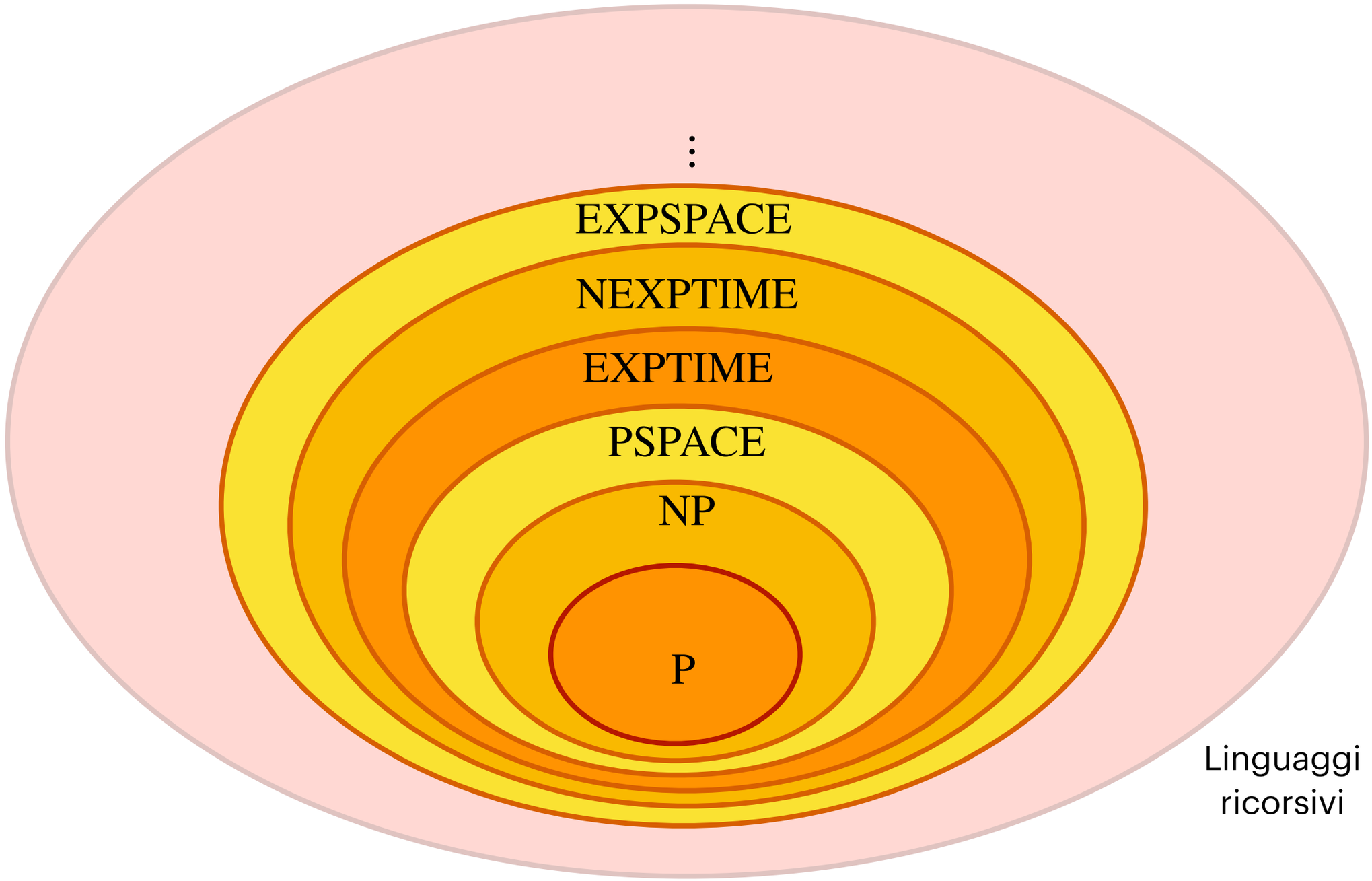
- Quindi $L \in \mathrm{PSPACE}$ implica $L \in \mathrm{EXPTIME}$

- Abbiamo quindi una relazione tra classi temporali e spaziali:

- $\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$

Classi di complessità

Questa volta con le inclusioni motivate



Spazio Polinomiale

- Vogliamo mostrare che il non determinismo non cambia i linguaggi riconoscibili in spazio polinomiale
- Vogliamo quindi mostrare che $\mathrm{PSPACE} = \mathrm{NPSPACE}$
- In realtà possiamo mostrare che questo vale per ogni classe in cui lo spazio è più che lineare
- Questo è dato dal **Teorema di Savitch** dimostrato da Walter Savitch (1943—2021) nel 1970

Teorema di Savitch

Teorema

Per ogni funzione $f : \mathbb{N} \to \mathbb{N}$ tale per cui $f(n) \geq n$ per ogni n vale la seguente relazione:

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2)$$

Corollario

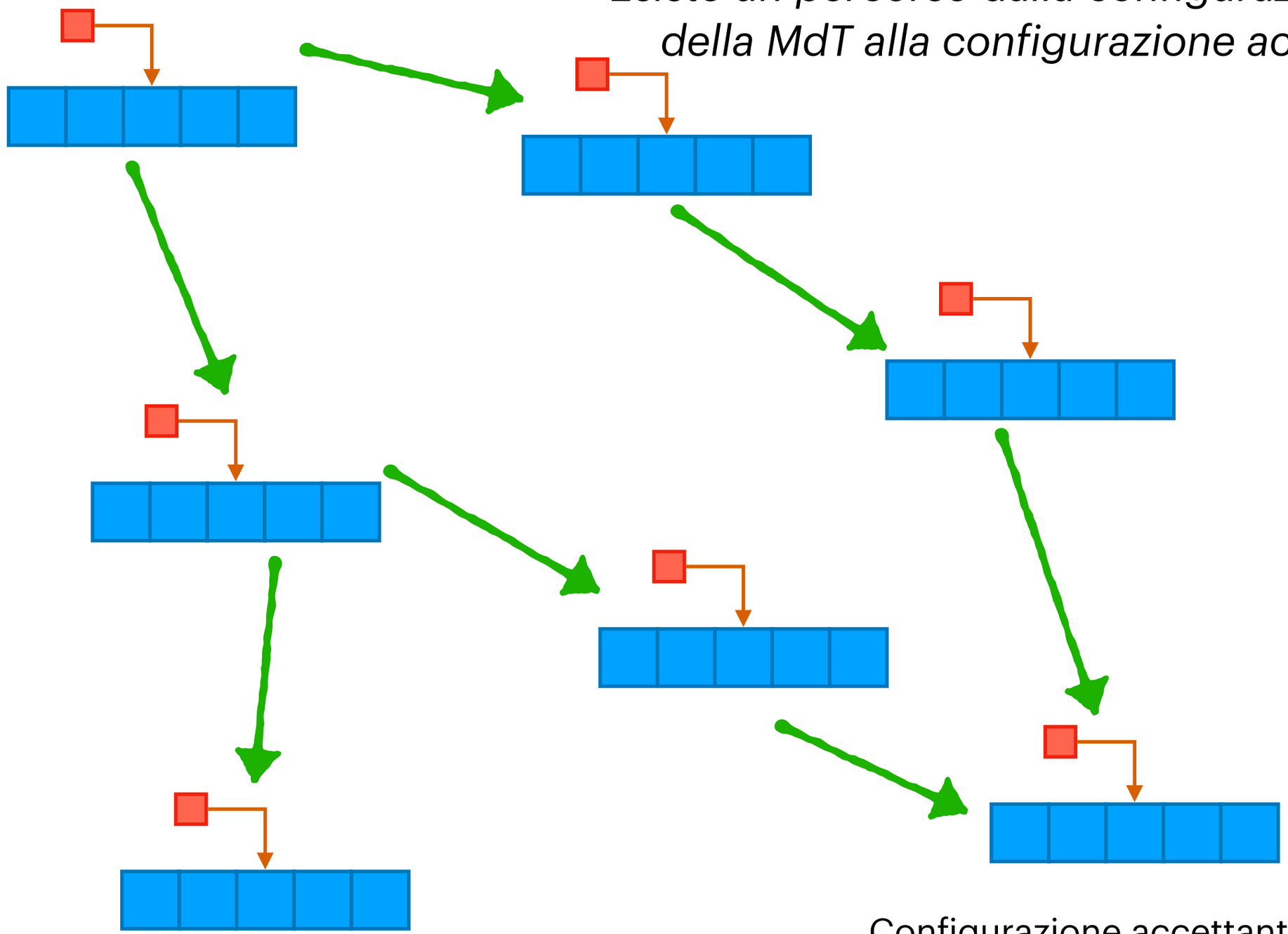
$$\text{PSPACE} = \text{NPSPACE}$$

In quanto se $f(n)$ è limitata da un polinomio allora $f(n)^2$ è anch'essa limitata da un polinomio

Teorema di Savitch: idea

Configurazione iniziale della MdT

Esiste un percorso dalla configurazione iniziale della MdT alla configurazione accettante?



Configurazione accettante della MdT

Teorema di Savitch: idea

- Per semplicità consideriamo che una MdT non deterministica che lavora in spazio polinomiale abbia una sola configurazione accettante C_{accept} e la sua configurazione iniziale sia C_0
- Pensiamo a configurazioni come nodi di un grafo...
- ... e alla relazione "La configurazione C_i permette di arrivare in un passo alle configurazione C_{i+1}" come una relazione che definisce gli archi di un grafo

Teorema di Savitch: idea

- Se possiamo stabilire se esiste un percorso tra C_0 e C_{accept} in spazio $O(f(n)^2)$ con una MdT deterministica allora mostreremmo che $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2)$
- Notate che il grafo è implicito! Non viene dato direttamente (sarebbe di dimensione esponenziale) ma:
  - Sapendo $f(n)$ sappiamo che dimensione hanno le configurazioni
  - Possiamo stabilire se due configurazioni sono connesse da un arco osservando la tabella di transizione della MdT

Teorema di Savitch: algoritmo

```
def connected(G, u, v, k):
    if k == 1:
        return (u,v) in G.V
    if k == 0:
        return u==v
    k1 = ⌊k/2⌋
    k2 = ⌈k/2⌉
    for each possible configuration c:
        first = connected(G, u, c, k1)
        second = connected(G, c, v, k2)
        if first and second:
            return True
    return False
```

Se due nodi sono a distanza uno allora basta controllare che abbiano un arco tra di loro

A distanza zero devono essere lo stesso nodo

Spezziamo in due il percorso

Se esiste una configurazione intermedia

Tale per cui esiste un percorso da u a c e da c a v

Allora u e v sono connessi

Teorema di Savitch

Chiamiamo questa funzione senza dare il grafo esplicito (se notate ci serve solo sapere se due configurazioni sono consecutive, quindi basta avere la descrizione della MdT)

Diamo come due configurazioni iniziali u e v rispettivamente C_0 e C_{accept}

Il numero k ci fornisce un limite alla lunghezza del percorso

Questa lunghezza è al massimo il numero di nodi del grafo...

...che sappiamo essere pari al numero di possibili configurazioni, ovvero $2^{O(f(n))}$

Teorema di Savitch

Possiamo verificare il caso base in spazio polinomiale date due configurazioni C_i e C_{i+1}

Le variabili locali sono:

- Interi k, k_1, k_2
- Tre configurazioni u, v, c

Quindi ogni chiamata alla funzione richiede spazio $O(f(n))$ per le sue variabili locali

Ma la funzione è ricorsiva...

Teorema di Savitch

...ci serve verificare quanto sia profondo lo stack di chiamate in ogni momento

Spazio totale usato dalla funzione è dato da

Spazio delle variabili locali $\times$ *profondità dello stack*

A ogni chiamata ricorsiva dimezziamo la lunghezza del percorso (il valore k), quindi il numero massimo di stack frame attivi in un dato momento è

$$\log_2 k = \log_2 2^{O(f(n))} = O(f(n))$$

Quindi lo spazio totale occupato dalla funzione è

$$O(f(n)) \times O(f(n)) = O(f(n)^2)$$

Teorema di Savitch

La funzione è deterministica e può tranquillamente verificare se esiste una computazione accettante di una macchina non deterministica che lavora in spazio $f(n)$

Quindi questo mostra che

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2)$$

Lo abbiamo visto non per una macchina di Turing deterministica, ma il procedimento rimane lo stesso.

Notate come la macchina deterministica costruita non sia efficiente in termini di tempo (esplora tutte le configurazioni), ma solo in termini di spazio

PSPACE

- Determinismo e non determinismo non cambiano PSPACE
- In particolare PSPACE è chiuso rispetto alla complementazione, così come P
- Esistono problemi PSPACE-completi?