Sommario

Le classi L e NL

NL completezza

PATH è NL completo

NL=coNL

TM modificate

Per poter considerare limiti di spazio sublineari dobbiamo dividere lo spazio necessario per l'input dallo spazio di memoria aggiuntivo.

Ogni TM d'ora in poi ha due nastri:
il nastro di input di sola lettura
il nastro di lavoro di lettura e scrittura

Prendiamo in considerazione solo il nastro di lavoro quando consideriamo la complessità di spazio della TM

Classi di complessità di spazio logaritmico

L è la classe dei linguaggi che sono decidibili in spazio logaritmico da una TM :

L = SPACE(log n)

NL è la classe dei linguaggi che sono decidibili in spazio logaritmico da una NTM :

NL = NSPACE(log n)

Esempio di linguaggio in L

Il linguaggio A = $\{0^k 1^k \mid k \geq 0\}$ è in L

Basta contare il numero degli 0 e poi quello degli 1 e confrontare i due numeri.
Questi numeri in binario occupano spazio logaritmico in k.

Esempio di linguaggio in NL

Il linguaggio PATH = {<G,s,t> | G è un grafo diretto con un cammino da s a t} è in NL.

Non è noto se c’è un algoritmo deterministico che lavora in spazio logaritmico per PATH.
L’algoritmo polinomiale in tempo è lineare in spazio.

Un algoritmo non deterministico per PATH, che lavora in spazio logaritmico, procede scegliendo il prossimo nodo tra tutti quelli adiacenti al nodo corrente, che è l’unico memorizzato sul nastro di lavoro.
Quindi lo spazio necessario è O(log |V|).

La NTM PATH

PATH(G=(V,E),s,t)
% accetta se c'è un cammino da s a t,
altrimenti rifiuta%
if t = s then accetta return
c = 1; p← s
ripeti finchè c ≤ |V|
    if ci sono vertici q tali   che (p,q) è in E
      then scegli non deterministicamente uno di
              questi vertici q e incrementa c,
      else rifiuta % il cammino non porta a t %
    if q = t then accetta
    p←q

Lo spazio necessario è quello per i valori delle variabili p,q i cui valori sono al più |V| e quindi rappresentabili in log (|V|)

Relazione tra spazio e tempo

Per una TM che lavora in spazio sublineare non è detto che $t_T(n) \leq 2^{O(s_T(n))}$

Per esempio una TM che lavora in spazio costante potrebbe prendere tempo lineare.

Dobbiamo modificare la definizione di configurazione.

Se T è una TM con un nastro di input di sola lettura una configurazione di T deve dar conto del contenuto del nastro di lavoro, dello stato e delle posizioni delle due testine, ma l'input non è parte della configurazione.

Relazione tra spazio e tempo - 2

Contiamo il numero delle configurazioni di una TM di cui è nota $s_T(n)$:
se d è il numero di simboli dell'alfabeto di nastro, il numero delle parole di lunghezza $s_T(n)$, cioè i diversi contenuti del nastro di lavoro, è $d^{s_T(n)}$

Tenendo conto che in ogni stato si può avere un qualsiasi contenuto di nastro, detto q il numero degli stati, abbiamo

$$q * d^{s_T(n)}$$

Relazione tra spazio e tempo - 2

Tenendo conto che la testina di lettura del nastro di input può trovarsi su una cella qualunque, tra gli n simboli in input, mentre la testina del nastro di lavoro si può trovare tra le $s_T(n)$ posizioni otteniamo

$$n * s_T(n) * q * d^{s_T(n)}$$

In conclusione $t_T(n) \leq n2^{O(s_T(n))}$

Se $s_T(n) \geq \log n$ allora

$$n2^{O(s_T(n))} \leq 2^{\log(n)} 2^{O(s_T(n))} \leq 2^{O(s_T(n))}$$

Teorema di Savitch per spazi sublineri

Abbiamo dimostrato il teorema per una NTM T con $s_T(n) \geq n$

Se $s_T(n) \geq \log n$ possiamo seguire gli stessi passi della prova, infatti abbiamo visto che $t_T(n) \leq 2^{O(s_T(n))}$, inoltre ogni configurazione (contenuto del nastro, stato e posizione delle due testine) occupa uno spazio

$$O(s_T(n) + \log n) = O(s_T(n))$$

se $s_T(n) \geq \log n$

NL-completezza

Quale riduzione?
Riduzione in spazio logaritmico:

Dati A $\subseteq\Sigma^*$ e B $\subseteq \Sigma^*$ due linguaggi diciamo che A <u>si riduce</u> in spazio logaritmico a B, in breve A $\leq_L$ B,

se esiste una funzione f : $\Sigma^* \rightarrow \Sigma^*$ <u>Turing calcolabile</u> in spazio logaritmico tale che

$$\boxed{x \in A \Leftrightarrow f(x) \in B}$$

Funzioni calcolabili in spazio logaritmico

Una TM T, con un nastro di input di sola lettura, un nastro di output di sola scrittura e un nastro di lavoro, calcola in spazio logaritmico una funzione

$$f : \Sigma^* \rightarrow \Sigma^*$$

se partita nello stato iniziale con x in Σ^* sul nastro di input, scrive f(x) sul nastro di output, utilizzando al più O(log n) celle del nastro di lavoro.

La TM T è detta trasduttore.

N.B. Poichè se $s_T(n) = O(\log n)$ allora $t_T(n) \leq n2^{k\log n} = n(2^{\log n})^k = n^{k+1}$. Quindi f è calcolata in tempo polinomiale.

NL-completezza

Un problema A è in NL- completo se

1. A è in NL,
2. B $\leq_L$ A, per ogni problema B in NL.

Teorema. Se B $\in$ L e A $\leq_L$ B, allora A $\in$ L.

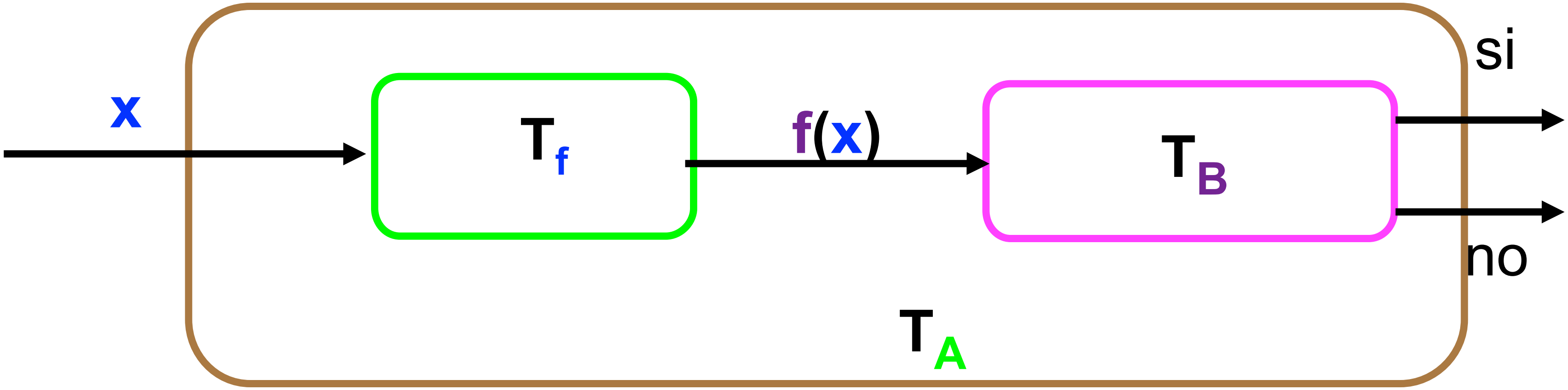
Potremmo utilizzare una prova analoga a quella per dimostrare che

se B $\in$ P e A $\leq_p$ B allora A $\in$ P

ma dobbiamo fare attenzione!

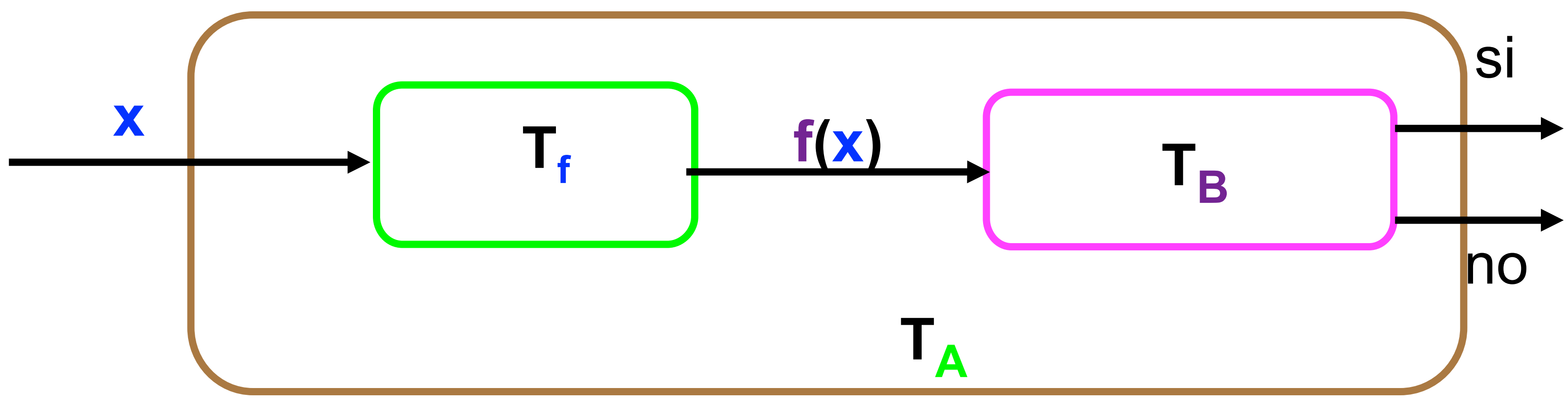
La TM per A: 1° tentativo

Se A $\leq_L$ B e T_B è la TM che decide B in spazio logaritmico, allora la TM T_A che decide A in spazio logaritmico, è ottenuta componendo <u>in sequenza</u> T_f , che è la TM che calcola f in spazio logaritmico, con T_B:



Ma f(x) potrebbe non essere rappresentabile in O(log n)!

La TM per A



T_B va in esecuzione su f(x), ma T_A non calcola prima f(x), bensì tiene conto della posizione, i, della testina di scrittura sul nastro di output di T_f, che produce l'input per T_B.

Ogni volta che T_B deve fare una mossa e ha bisogno del simbolo nella posizione i sul suo nastro in input chiama T_f, ne ignora i primi i -1 simboli di output scrivendo solo l' i-simo sul suo nastro di output. Una volta ottenuto il simbolo la TM T_B può eseguire la sua mossa.

Sappiamo che i può essere rappresentato in O(log n) perchè l'output non può essere più lungo del tempo e se $s_T(n) = O(\log n)$ allora $t_T(n) \leq n2^{k\log n} = n(2^{\log n})^k = n^{k+1}$.

PATH è NL-completo

Sappiamo già che PATH = {<G,s,t> | G è un grafo diretto con un cammino da s a t} è in NL.
Si tratta di far vedere che per ogni problema B in NL

$$B \leq_L \text{PATH}$$

Possiamo supporre che B sia sull'alfabeto binario.

Sia T_B una NTM che in spazio logaritmico decide B, modificata in modo da avere un'unica configurazione finale, c_F.

PATH è NL-completo

Costruiamo una riduzione in spazio logaritmico f tale che

$$f(w) = <G,s,t>$$

per ogni w in {0,1}*, in modo tale che

w è in B sse G ha un cammino da s a t.

Sia G = G(T_B) il grafo di transizione di T_B su w, cioè un grafo in cui i nodi rappresentano configurazioni su w e gli archi la possibilità di passare da una configurazione ad un altra.

Sia s = c_0 = (q_0,⊥,1,1), la configurazione iniziale su w dove ⊥ è il simbolo di cella vuota,

e t = c_F =(q_a,⊥,1,1) la finale di T_B.

Evidentemente w è in B sse c'è una computazione di T_B su w che porta alla configurazione finale e quindi un cammino da s a t.

Ma si può calcolare G(T_B) in spazio logaritmico?

PATH è NL-completo

Per rappresentare ciascun nodo serve solo O(log n) spazio visto che ogni nodo è una configurazione di T_B.
Il trasduttore elenca tutte le stringhe di lunghezza c * log n, dove c è un' opportuna costante, e salva sul nastro di output solo quelle che sono configurazioni legittime su w, cioè stringhe binarie che codificano una quadrupla in cui un elemento è uno degli stati della TM T_B un altro è un contenuto di nastro, e gli altri due rappresentano le due posizioni delle testine sul nastro di input e sul nastro di lavoro. In ogni passo solo una configurazione è sul nastro di lavoro.

Per gli archi il lavoro è analogo, il trasduttore elenca tutte le coppie (c_1,c_2) e salva solo quelle per cui c_2 è raggiungibile in un passo da c_1. In ogni passo solo una coppia di configurazioni è sul nastro di lavoro. Questo controllo può essere fatto in spazio logaritmico perché è solo necessario esaminare c_1 per sapere la posizione della testina di lettura sul nastro di lavoro, quindi il simbolo in lettura che insieme allo stato determina le possibili mosse e quindi le configurazioni raggiungibili.

$NL \subseteq P$

Per dimostrarlo osserviamo che:

1. PATH è in P

2. B $\leq_L$ PATH, per ogni problema B in NL.

3. Una riduzione in spazio logaritmico è polinomiale in tempo.