16/04/2024 LEZ 12

DTIME VELOCIZZATO

SUPPONIAMO T, t.c.:

$$\forall x \in \Sigma^* \left[dTIME(T, x) \leq f(|x|) \right]$$

SIA INVECE T', t.c.

$$// \left[dTIME(T, x) \leq \frac{f(|x|)}{2} \right]$$

e così via, ora la domanda è: arriverà ad una fine questa sequenza? NO.

MOSTRATO NEL SEGUENTE TEOREMA

TEOREMA 6.7 (DISP. 6) → teorema della accelerazione

SIA L DECISO DA CON $dTIME(T, x) \leq f(|x|)$. ALLORA:

- $\forall k \in \mathbb{N} \, [\exists T'_k$, 1 NASTRO, CHE DECIDE L CON $dTIME(T'_k, x) \leq \frac{f(|x|)}{k} + |x^2| \; \forall x \in \Sigma^*]$
- // // T'_k A 2 NASTRI // //
  // // // // // // + $|x|$ //

X CAPIRE IDEA OF DIM., NUOVO TEOREMA

TEOREMA 6.6. teorema della compressione

SIA L DECISO DA $\mathcal{T}$, CON $dSPACE(\mathcal{T}, x) \leq f(|x|)$ THEN

$\forall k \in \mathbb{N} \, [\exists \mathcal{T}_k$ CHE DECIDE L CON

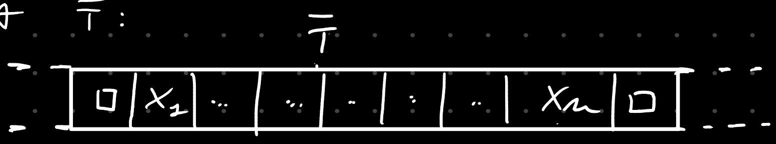
$$dSPACE(\mathcal{T}_k, x) \leq \frac{f(|x|)}{k} + |x|$$

la dimostrazione è molto tecnica, nella pagina dopo è scritta una 'idea' della dimostrazione

DIMOSTRAZIONE

START TO 6.6

SIA $\bar{T}$:



L'IDEA È "COMPRIMERE" X IN Y, COSÌ CHE:

∀ SINGOLA CELLA DI Y
=
K SIMBOLI DI X

SIA T_k:

ALFAB. $= \Sigma^k \cup \Sigma$

INPUT $= X$

PROC. → LEGGE K CARATTERI, SCRIVE 1 CARATTERE $Y_{x_1 \dots x_k}$ ↗ 1 SIMBOLO X RAPP.

CANCELLO I K CARATTERI

↓

RICOMINCIO FINO ALLA FINE.

QUESTO VIENE USATO IN 6.7., DATO T, T' FA:

1. COMPRESSIONE.
2. FASE 2 -> = A 6.6. (SEE DISPENSA)

LA ZIP COSTA IN BASE AI NASTRI:

- 1 NASTRO $= O(|x|^2)$ → x ZIP COMPRIMO C CARATTERI POI RETURN INDIETRO E RICOMINCIO.
- 2 NASTRI $= O(|x|)$ ↳ USO 2 TESTINE SENZA FARE GIRI LUNGHI.

DATO L, SE È DECIDIBILE IN $\mathcal{L}$ $TIME(T, x) = f(|x|)$, POSSIAMO QUINDI SEMPRE FARE MEGLIO, NON SERVE CONCENTRARSI SUL LOWER BOUND, QUANTO PIÙ SULL'UPPER BOUND.

ED È QUI CHE ENTRANO IN GIOCO LE CLASSI DI COMPLESSITÀ

CLASSE DI COMPLESSITÀ

DEF SET DI LINGUAGGI DECIDIBILI DA M.d.t, UTILIZZANDO RISORSE NON OLTRE UNA SOGLIA

DEF FORMALE

SET LINGUAGGI

$$DTIME = \{ L \in \{0,1\}^* : \exists \text{ CHE DECIDE } L \text{ e } \forall x \in \{0,1\}^* \; o(TIME(T,x)) \in \Theta(f(x)).$$

TUTTO MAIUSCOLO

F TOT. CALCOLABILE

ALTO = MODO

$$DSPACE[f(n)] = \{ \text{ // // // // // // } \forall x \in \{0,1\}^* \; dSPACE(T,x) \in \Theta(f(x)).$$

N.B.:

WHY F DEVE ESSERE TOTALE E CALCOLABILE?

XCHÈ CI SERVE COME MISURA LIMITE

↳ SE NON È DEFINITO, ALLE VOLTE NON SAPREMO MAI IL LIMITE.

CLASSI COMPLESS. NON. DETERMINIST.

$$NTIME = \{ L \subseteq \{0,1\}^* : \exists NT \text{ CHE ACCETTA } L \text{ e } \forall x \in L \; nTIME(NT,x) \leq \Theta(f(x)) \}$$

NSPACE = STESSA COSA CON nSPACE AL POSTO DI nTIME.

CLASSI COMPLEMENTO

$$\text{CODTIME}[f(n)] = \{L \subseteq \{0,1\}^* : L^c \in \text{DTIME}[f(n)]\}$$

$$\text{CONTTIME}[f(n)] = \{L \subseteq \{0,1\}^* : L^c \in \text{NTIME}[f(n)]\}$$

(= COSA APPLICABILE CON CODSPACE e CONSPACE

RELAZIONI TRA CLASSI

qua verrà scritto solo la parte di TIME, anche con SPACE sono validi questi teoremi. E con le macchine non deterministiche

6.8

① $\forall f$ TOT. $\wedge$ CALCO. $\Rightarrow$ $\text{DTIME}[f(n)] \subseteq \text{NTIME}[f(n)]$

= x DSPACE e NSPACE.

XCHÈ? $\{T_{\text{DETE}}\} \subseteq \{NT\}$

6.9

② $\forall f$ TOT e CALCOL. $\Rightarrow$ $\text{DTIME}[f(n)] \subseteq \text{DSPACE}[f(n)]$

WHY?

$\forall T \rightarrow d\text{SPACE}(T,x) \leq d\text{TIME}(T,x)$ x 6.1 (*)

TRADOTTO = $\forall$

$\Downarrow$

SIA $L \in \text{DTIME}[f(n)] \Rightarrow \exists T$ DECIDE L e $\forall x \; d\text{TIME}(T,x) \in O(f(n))$.

IMPLICA CHE $d\text{SPACE} \in O(f(n))$ (*) $\Rightarrow$ $L \in \text{DSPACE}[f(n)]$

③ 6.10

$\forall f$ TOT., CALCOLABILE $\longrightarrow$ $\text{DSPACE}[f(n)] \subseteq \text{DTIME}\left[2^{O(f(n))}\right]$

WHY? DALLA 6.1.

$\forall L \in \text{DSPACE}[f(n)] \Rightarrow \exists T$ CHE DEC. E $\forall x \; d\text{SPACE}(T,x) \in O(f(n))$

DA 6.1:

$$d\text{TIME}(T,x) \leq d\text{SPACE}(T,x)\,|Q|\,(|\Sigma|+1)^{d\text{SPACE}(T,x)}$$

$\Downarrow$ TRASFORMO CON $2^{\log_2}$

$$d\text{TIME}(T,x) \leq 2^{\log d\text{SPACE}(T,x)} \; |Q| \cdot 2^{d\text{SPACE}(T,x) \cdot \log(|\Sigma|+1)}$$

$$= \underbrace{|Q| \cdot 2^{[1+\log(|\Sigma|+1)]\, d\text{SPACE}(T,x)}}_{\text{COSTANTE}} \in 2^{O(f(n))}$$

4) $\forall f$. TOT. CALCOL.. $DTIME[f(n)] = coDTIME[f(n)]$

WHY? COSTRUZIONE

- $\forall L \in DTIME(f(n)) \Rightarrow \exists\ T$ DECIDE. L e $dTIME(T,x) \in \Theta(f(n))$.
- COSTR. L_c CHE DECIDE L_c, $\forall x \in \Sigma^*$:
  - SIMULO $T(x)$
  - SE $O_T(x) = q_A \rightarrow O_{T'}(x) = q_R$
  - " " $q_R \rightarrow$ " q_A

PRATICAMENTE

$$dTIME(T',x) = dcoTIME(T,x) = dTIME(T,x) + \underset{\text{ISTRUZIONE}}{1}$$

queste propietà, di per se, non bastano ad evitare strane relazioni che intercorrono tra classi di complessità. Come detto all'inizio, per ogni macchina T esiste una macchina T' che decide lo stesso linguaggio, ma in tempo minore. Quindi possiamo formulare il seguente teorema.

TEOREMA 6.12

SIANO f, g, TOT E CALCOLABILI, e $F(n) \in O(g(n))$.

ALLORA

- $DTIME[f(n)] \subseteq DTIME[g(n)]$
- $NTIME[f(n)] \subseteq NTIME[g(n)]$

STESSA COSA CON DSPACE e NSPACE

quindi, se io decidessi che L appa. a DSPACE[g(n)], ci sarà sempre la POSSIBILITÀ che una altro tizio dimostri che L può stare anche in f(n) (cioè di migliorarlo.

NON CI DANNO QUINDI UNA PRECISA MISURA DI DOVE L PUÒ STARE.

AH, MA ALMENO....

CON $F(n)$ MOLTO + PICCOLO DI $g(n)$ $\left[f(n) \in 2^{g(n)}\right]$ VALE SEMPRE LA 6.10. NO? NOP.

GAP THEORY 6.13.

$\exists$ UNA $F: \mathbb{N} \to \mathbb{N}$, TOT. CALCOLABILE, T.C.

$$\boxed{DTIME[2^{f(n)}] \subseteq DTIME[f(n)]}$$

QUINDI, SONO INSIEMI NON MIGLIORI, ANZI PEGGIORI.

E MO? CI AIUTERANNO NEW TOOL, DOVE 6.13. NON VALE.

DEF

$f: \mathbb{N} \to \mathbb{N}$ È TIME-CONSTRUCTBLE SE:

- F TOT e CALC.
- $\exists\, T_F$ CHE CON INPUT n IN UNARIO WRITE IN N_0 $F(n)$ IN UNARIO.

  e $\forall n \in \mathbb{N}$ $dTIME(T_F, n) \in O(f(n))$.

SPACE-CONSTRUCTBLE = TIME-CONST. MA SOST. COL SPACE.

TUTTE LE F. REGOLARI, ANCHE LE ESPONENZIALI $(2^{F(n)})$ SONO TIME e SPACE-CONSTRUCTBLE.

DA QUI SI SUSSEGUONO I TEOREMI DI GERARCHIA

TEOREMA DI GERARCHIA SPAZIALE

SIANO f, g : f È SPACE-CONSTRUCTBLE ∧ $\lim_{n \to x_0} \frac{f(n)}{g(n)} = 0$

g(n) CRESCE + FAST DI f(n)

ALLORA

$$\boxed{DSPACE[f(n)] \subset DPSACE[g(n)]}$$

TEOREMA DI GERARCHIA TEMPORALE

SIANO F, g, F TIME - CONTR.

$$\lim_{n \to n_0} \frac{g(n) * log(g(n))}{f(n)} = 0$$

$$\Downarrow$$

$$DTIME(g(n)) \subset DTIME(f(n))$$