PART THREE

COMPLEXITY THEORY

TEORIA DELLA COMPLESSITA'

Riduzioni polinomiali (e non)

17 maggio 2022

Teoria della complessità: argomenti trattati

Scorse lezioni:

- Definizione di complessità di tempo
- La complessità di tempo dipende dal modello di calcolo; useremo decisori e modelli polinomialmente equivalenti
- La complessità di tempo dipende dalla codifica utilizzata: useremo codifica in binario o polinomialmente correlata
- TIME (f(n)) = insieme dei linguaggi decisi in tempo O(f(n))
- La classe $P = \bigcup_{k \geq 0} TIME(n^k)$ e sua robustezza
- La classe EXPTIME
- algoritmi di verifica e la classe NP

Oggi:

- Il concetto di riduzione polinomiale
- Esercizi sulle riduzioni

Funzioni calcolabili

Le MdT possono essere utilizzate per il calcolo di funzioni.

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ *è calcolabile se esiste una macchina di Turing* $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ *tale che*

$$\forall w \in \Sigma^* \quad q_0 w \to^* q_{accept} f(w)$$

Funzioni calcolabili in tempo polinomiale

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ è **calcolabile in tempo polinomiale** *se esiste una macchina di Turing deterministica M di complessità di tempo* **polinomiale** *tale che su ogni input w, M si arresta con $f(w)$, e solo con $f(w)$, sul suo nastro.*

- Esempio 1. Consideriamo la funzione $f : \{0,1\}^* \to \{0,1\}^*$ tale che $f(\langle m \rangle) = \langle m+1 \rangle$, dove $m \in \mathbb{N}$ e $\langle m \rangle$ è la rappresentazione binaria di m.

  La funzione f è calcolabile in tempo polinomiale nella lunghezza dell'input ⟨m⟩.

- Esempio 2. Consideriamo la funzione $g : \{0,1\}^* \to \{0,1\}^*$ tale che $g(\langle m \rangle) = \langle m \rangle \# 1^m$, dove $m \in \mathbb{N}$ e $\langle m \rangle$ è la rappresentazione binaria di m.

  La funzione g è calcolabile, ma non in tempo polinomiale nella lunghezza dell'input ⟨m⟩.

Definizione

Siano A, B linguaggi sull'alfabeto Σ.

Una riduzione in tempo polinomiale f di A in B è

- una funzione $f : \Sigma^* \rightarrow \Sigma^*$
- calcolabile in tempo polinomiale
- tale che per ogni $w \in \Sigma^*$

$$w \in A \Leftrightarrow f(w) \in B$$

Definizione

Un linguaggio $A \subseteq \Sigma^*$ è riducibile in tempo polinomiale a un linguaggio $B \subseteq \Sigma^*$, e scriveremo $A \leq_p B$, se esiste una riduzione di tempo polinomiale di A in B.

Teorema

Se $A \leq p\ B$ e $B \in P$, allora $A \in P$.

Dimostrazione

… e altri teoremi … la prossima volta ;))

Una possibile codifica di MdT

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ con transizioni $m_1, m_2, \ldots m_t$ mediante la stringa

$$\langle M \rangle =$$

$$11111e(q_0)1e(q_{accept})1e(q_{reject})1e(\Sigma)1e(\Gamma)1e(m_1)1 \cdots 1e(m_t)11111$$

Codifichiamo una transizione m di una TM, ad esempio $\delta(q, a) = (p, b, D)$, con la stringa

$$e(m) = 11e(q)1e(a)1e(p)1e(b)1e(D)11$$

Codifichiamo un elemento $q_i \in Q_U$ con la stringa $e(q_i) = 0^{i+1}$

SAT

Una formula booleana ϕ è soddisfacibile se esiste un insieme di valori 0 o 1 per le variabili di ϕ (o assegnamento) che renda la formula uguale a 1 (assegnamento di soddisfacibilità). Diremo che tale assegnamento soddisfa ϕ o anche che rende vera ϕ.

Il problema della **soddisfacibilità di una formula booleana**:
Data una formula booleana ϕ, ϕ è soddisfacibile?

Il linguaggio associato è:

$$\boxed{SAT = \{\langle \phi \rangle \mid \phi \text{ è una formula booleana soddisfacibile}\}}$$

3 SAT

Definizione

Una clausola è un OR di letterali.

Esempio: $(\overline{x} \vee x \vee y \vee z)$

CNF

Definizione

Una formula booleana ϕ è in forma normale congiuntiva (o forma normale POS) se è un AND di clausole, cioè è un AND di OR di letterali.

Definizione

Una formula booleana è in forma normale 3-congiuntiva se è un AND di clausole e tutte le clausole hanno tre letterali.

Esempio:

$$(\overline{x_1} \vee x_2 \vee x_3) \wedge (x_3 \vee \overline{x_6} \vee x_6) \wedge (x_3 \vee \overline{x_5} \vee x_5)$$

$$\boxed{3SAT = \{\langle\phi\rangle \mid \phi \text{ è una formula 3CNF soddisfacibile}\}}$$

Teorema

$$\boxed{3SAT \leq_P CLIQUE}$$

$3SAT = \{\langle \phi \rangle \mid \phi$ è una formula 3CNF soddisfacibile$\}$

Una formula 3CNF è un AND di clausole e tutte le clausole hanno tre letterali.

$CLIQUE =$
$\{\langle G, k \rangle \mid G$ è un grafo non orientato in cui esiste una k-clique$\}$

Ricorda:
Una clique (o cricca) in un grafo non orientato G è un sottografo G' di G in cui ogni coppia di vertici è connessa da un arco.
Una k-clique è una clique che contiene k vertici.

$3SAT \leq_p CLIQUE$

Dobbiamo dimostrare che esiste una funzione $f : \Sigma^* \rightarrow \Sigma^*$

- calcolabile in tempo polinomiale
- tale che per ogni $w \in \Sigma^*$ $w \in 3SAT \Leftrightarrow f(w) \in CLIQUE$

Convenzione: non specificheremo il valore di f sulle stringhe che non rappresentano un'istanza del problema.

Quindi definiremo la f solo su stringhe che codificano formule booleane in 3CNF ϕ e ad esse assoceremo stringhe che codificano (G, k).

$$f : \langle\phi\rangle \rightarrow \langle G, k\rangle$$

$$\langle\phi\rangle \in 3SAT \Leftrightarrow \langle G, k\rangle \in CLIQUE$$

$3SAT \leq_p CLIQUE$

$$f : \langle \phi \rangle \rightarrow \langle G, k \rangle$$

$$\langle \phi \rangle \in 3SAT \Leftrightarrow \langle G, k \rangle \in CLIQUE$$

Poi dimostreremo che:

- f è calcolabile
- Se ϕ è soddisfacibile allora G ha una k-clique
- Se il grafo associato G ha una k-clique allora ϕ è soddisfacibile.

Vediamo come associare ad ogni formula in 3CNF un grafo e un intero.

$3SAT \leq_p CLIQUE$

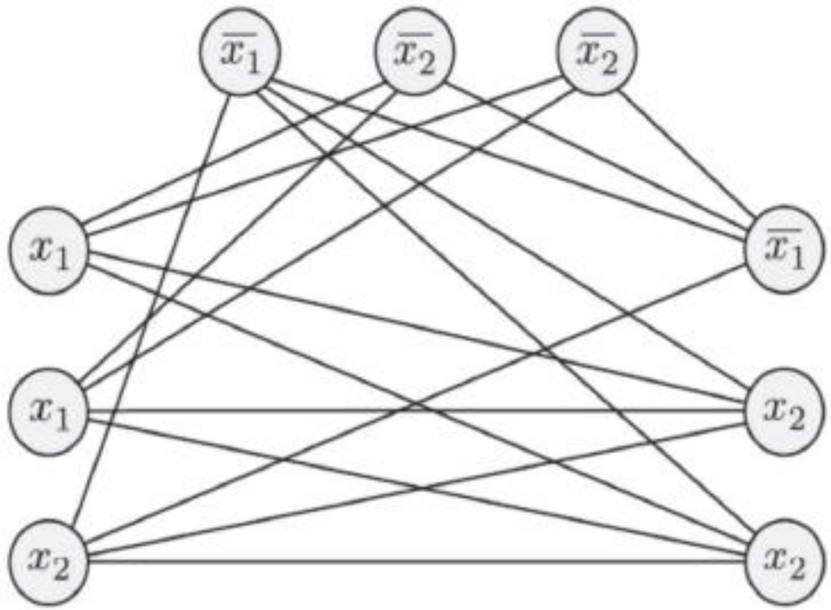

FIGURA 7.33
Il grafo che la riduzione produce per $\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$

$3SAT \leq_p CLIQUE$

Teorema

3SAT è riducibile in tempo polinomiale a CLIQUE.

Dimostrazione

- Sia ϕ una formula $3CNF$ con k clausole:

$$(a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$$

- Consideriamo la funzione f che associa a $\langle \phi \rangle$ la stringa $\langle G, k \rangle$ dove $G = (V, E)$ è il grafo non orientato definito come segue:
- V ha $3 \times k$ vertici. I vertici di G sono divisi in k gruppi di tre nodi (o **triple**) $t_1, \ldots, t_k$: t_j corrisponde alla clausola $(a_j \vee b_j \vee c_j)$ e ogni vertice in t_j corrisponde a un letterale in $(a_j \vee b_j \vee c_j)$. Quindi $V = \{a_1, b_1, c_1, \ldots, a_k, b_k, c_k\}$.
- Non ci sono archi tra i vertici in una tripla t_j, non ci sono archi tra un vertice associato a un letterale x e i vertici associati al letterale $\overline{x}$.
- Ogni altra coppia di vertici è connessa da un arco.

Nota: k è il numero di clausole in ϕ.

La funzione f è calcolabile e può essere calcolata in tempo polinomiale.

Per provare che f è una riduzione di tempo polinomiale di 3SAT in *CLIQUE* resta da dimostrare che

$$\langle \phi \rangle \in \mathit{3SAT} \Leftrightarrow \langle G, k \rangle \in \mathit{CLIQUE}$$

Cioè ϕ è soddisfacibile se e solo se G ha una k-clique.

$3SAT \leq_p CLIQUE$

- Supponiamo che ϕ abbia un assegnamento di soddisfacibilità. Questo assegnamento di valori alle variabili rende vera ogni clausola $(a_j \vee b_j \vee c_j)$ e quindi esiste almeno un letterale vero in ogni clausola $(a_j \vee b_j \vee c_j)$.
- Scegliamo un letterale vero in ogni clausola $(a_j \vee b_j \vee c_j)$ e consideriamo il sottografo G' di G indotto dai nodi corrispondenti ai letterali scelti.
- G' è una k-clique.
- Infatti G' ha k vertici poiché abbiamo scelto un letterale in ognuna delle k clausole e poi i k vertici di G corrispondenti a tali letterali.
- Due qualsiasi vertici in G' non si trovano nella stessa tripla (corrispondono a letterali in clausole diverse) e non corrispondono a una coppia $x, \overline{x}$ perché corrispondono a letterali veri nell'assegnamento di soddisfacibilità. Quindi due qualsiasi vertici in G' sono connessi da un arco in G.

- Viceversa, supponiamo che G abbia una k-clique G'.
- Poiché due nodi in una tripla non sono connessi da un arco, ognuna delle k triple contiene esattamente uno dei nodi della k-clique.
- Consideriamo l'assegnamento di valori alle variabili di ϕ che renda veri i letterali corrispondenti ai nodi di G'. Ciò è possibile perché in G' non ci sono archi che collegano una coppia $x, \overline{x}$.
- Ogni tripla contiene un nodo di G' e quindi ogni clausola contiene un letterale vero.
- Questo è un assegnamento di soddisfacibilità per ϕ cioè $\langle \phi \rangle \in 3SAT$.

$\square$

I risultati precedenti ci dicono che:

se *CLIQUE* fosse decidibile in tempo polinomiale anche *3SAT* lo sarebbe.

Questa connessione tra i due linguaggi sembra veramente notevole perché i linguaggi sembrano piuttosto differenti.

Esercizi svolti

Riduzione da A_{TM} a EQ_{TM}

$$A_{TM} \leq_m EQ_{TM}$$

Idea: Data $\langle M, w \rangle$, considerare le MdT M_1 e M_2 tali che

Per ogni input x:

M_1 accetta x,

M_2 simula M su w. Se M accetta w, M_2 accetta x.

$f : \langle M, w \rangle \rightarrow \langle M_1, M_2 \rangle$ è riduzione da A_{TM} a EQ_{TM}.

Perchè?

$$L(M_1) = \Sigma^*;\ L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w \rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w \rangle \notin A_{TM} \end{cases}$$

Riduzione da A_{TM} al complemento di EQ_{TM}

$f : \langle M, w\rangle \to \langle M_1, M_2\rangle$ è riduzione che prova $A_{TM} \leq_m EQ_{TM}$.

$$L(M_1) = \Sigma^*;\ L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w\rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w\rangle \notin A_{TM} \end{cases}$$

Possiamo modificare f per dimostrare che $A_{TM} \leq_m \overline{EQ_{TM}}$?

Lasciamo la stessa M_2 e cambiamo M_1 in $\mathbf{M_3}$.
$g : \langle M, w\rangle \to \langle M_3, M_2\rangle$

$$\mathbf{L(M_3)} = \emptyset;\ L(M_2) = \begin{cases} \Sigma^* & \text{se } \langle M, w\rangle \in A_{TM} \\ \emptyset & \text{se } \langle M, w\rangle \notin A_{TM} \end{cases}$$

Riduzione da linguaggio diagonale

Sia $L_d = \{\langle M \rangle \mid M \notin L(M)\}$ il linguaggio diagonale e $L_{ne} = \{\langle M \rangle \mid L(M) \neq \emptyset\}$. Si dimostri che

$$\bar{L}_d \leq L_{ne}$$

Riduzione da A_{TM}

Si consideri il linguaggio

$$L = \{\langle M_1, M_2, w\rangle \mid M_1 \text{ ed } M_2 \text{ sono } TM,\ M_1 \text{ accetta } w \text{ ed } M_2 \text{ accetta } w\}.$$

Provare che $A_{TM} \leq L$.

Esercizio 5.11 da [Sipser]

Mostrare che A è decidibile se e soltanto se A si riduce mediante funzione al linguaggio 0*1*.

Esercizi da svolgere

Riduzione da $HALT_{TM}$

Si consideri il linguaggio

$$L = \{\langle M \rangle \mid M \text{ è una MdT che si arresta su 11 e non si arresta su 00}\}.$$

Definire il linguaggio $HALT_{TM}$ e dimostrare che $HALT_{TM} \leq_m L$.

…. e altri dalla piattaforma.