Elementi di
Teoria della Computazione

Classe: Resto_2 - Prof.ssa Marcella Anselmo

Tutorato
25/05/2022 ore 14:00-16:00

Terza Esercitazione

a cura della dott.ssa Manuela Flores

Esercizio: Complemento di HALT_TM non è Turing-riconoscibile
su elearning.informatica.unisa.it

Definire il linguaggio HALT e provare che il suo complemento ‘HALT non è Turing-riconoscibile, enunciando i risultati che vengono utilizzati, senza dimostrarli (si suggerisce l'utilizzo di riduzioni mediante funzioni studiate e di note proprietà delle riduzioni mediante funzione)

Lezione 27 pag. 25

Indecidibilità del problema della fermata

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ accetta } w\}$$

$$\boxed{HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}}$$

Teorema
$A_{TM} \leq_m HALT_{TM}$.

Dimostrazione
Occorre definire una funzione calcolabile $f \; : \Sigma^* \rightarrow \Sigma^*$ tale che $f(\langle M, w\rangle) = \langle M', w\rangle$ e

$$\langle M, w\rangle \in A_{TM} \quad \text{sse} \quad \langle M', w\rangle \in HALT_{TM}$$

Lezione 27 pag. 25

Indecidibilità del problema della fermata

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ accetta } w\}$$

$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ è una MdT e } M \text{ si arresta su } w\}$$

Teorema
$A_{TM} \leq_m HALT_{TM}$.

Dimostrazione
Occorre definire una funzione calcolabile $f : \Sigma^* \to \Sigma^*$ tale che $f(\langle M, w\rangle) = \langle M', w\rangle$ e

$$\langle M, w\rangle \in A_{TM} \quad \text{sse} \quad \langle M', w\rangle \in HALT_{TM}$$

Lezione 27 pag. 19

Teoremi

Teorema

$A \leq_m B$ *se e solo se* $\overline{A} \leq_m \overline{B}$.

Dimostrazione

Per ipotesi $A \leq_m B$, quindi esiste una riduzione di A a B.

Poiché f è una riduzione, f è calcolabile e inoltre

$$\forall w \in \Sigma^* \quad w \in A \Leftrightarrow f(w) \in B$$

Proviamo che f è anche una riduzione da $\overline{A}$ a $\overline{B}$.

Lezione 26 pag. 47

Un linguaggio che non è Turing riconoscibile

Teorema

$\overline{A_{TM}}$ *non è Turing riconoscibile.*

Dimostrazione.

Supponiamo per assurdo che $\overline{A_{TM}}$ sia Turing riconoscibile.
Sappiamo che A_{TM} è Turing riconoscibile.
Quindi A_{TM} è Turing riconoscibile e co-Turing riconoscibile.
Per il precedente teorema, A_{TM} è decidibile.
Assurdo, poichè abbiamo dimostrato che A_{TM} è indecidibile.

$\square$

Lezione 28 pag. 6

Risultati

Teorema
$A \leq_m B$ se e solo se $\overline{A} \leq_m \overline{B}$.

Teorema
Se $A \leq_m B$ e B è *decidibile*, allora A è *decidibile*.

Teorema
Se $A \leq_m B$ e B è Turing *riconoscibile*, allora A è Turing *riconoscibile*.

Corollario
Se $A \leq_m B$ e A è *indecidibile*, allora B è *indecidibile*.

Corollario
Se $A \leq_m B$ e A *non* è Turing *riconoscibile*, allora B *non* è Turing *riconoscibile*.

Esercizio: Riduzione da ALL_DFA ad E_DFA

su elearning.informatica.unisa.it

(a) Dare la definizione di riducibilità mediante funzione di un linguaggio A a un linguaggio B.

(b) Siano
$ALL_{DFA} = \{\langle \mathcal{A} \rangle \mid \mathcal{A}$ è un DFA e $L(\mathcal{A}) = \Sigma^*\}$ e $E_{DFA} = \{\langle \mathcal{A} \rangle \mid \mathcal{A}$ è un DFA e $L(\mathcal{A}) = \emptyset\}$.
Provare che $ALL_{DFA} \leq_m E_{DFA}$.

Lezione 27 pag. 17

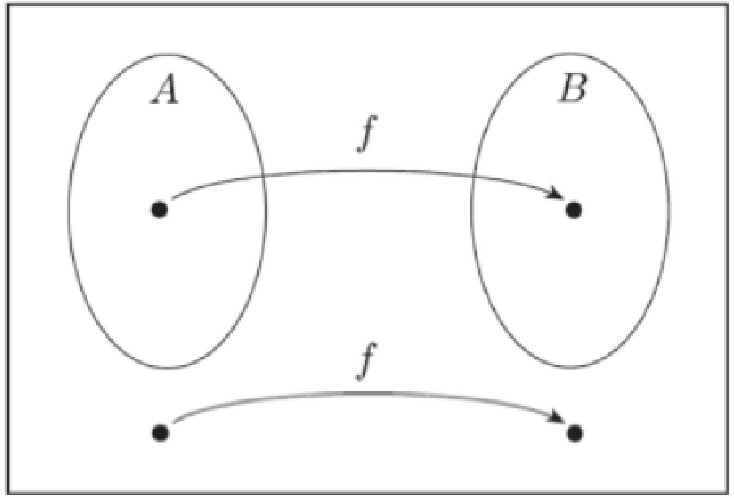
Riducibilità mediante funzione

Definizione

Un linguaggio $A \subseteq \Sigma^*$ è riducibile mediante funzione a un linguaggio $B \subseteq \Sigma^*$, e scriveremo $A \leq_m B$, se esiste una funzione calcolabile $f : \Sigma^* \to \Sigma^*$ tale che $\forall w \in \Sigma^*$

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

La funzione f è chiamata una riduzione da A a B.



Esercizio: Teorema di Rice su ALL_TM

Sipser 5.18(c) e su elearning.informatica.unisa.it

Usare il teorema di Rice per provare l'indecidibilità di:

$$ALL_{TM} = \{\langle M \rangle \mid M \text{ è una MdT ed } L(M) = \Sigma^* \}$$

Lezione 28 pag. 24

Teorema di Rice

Teorema di Rice. Sia

$$L = \{\langle M \rangle \mid M \text{ è una MdT che verifica la proprietà } \mathcal{P}\}$$

un linguaggio che soddisfa le seguenti due condizioni:

1. $\mathcal{P}$ è una proprietà del linguaggio $L(M)$, cioè: prese comunque due MdT M_1, M_2 tali che $L(M_1) = L(M_2)$ risulta
$$\langle M_1 \rangle \in L \Leftrightarrow \langle M_2 \rangle \in L$$
2. $\mathcal{P}$ è una proprietà non banale, cioè: esistono due MdT M_1, M_2 tali che
$$\langle M_1 \rangle \in L, \langle M_2 \rangle \notin L.$$

Allora L è indecidibile.

Esercizio: Riduzione da HALT_TM ad HO_TM
su elearning.informatica.unisa.it

Poniamo

$$HO_{TM} = \{\langle M \rangle \mid M \text{ è una macchina di Turing che si arresta su ogni input di lunghezza dispari } \}$$

Definire il linguaggio $HALT_{TM}$ e provare che $HALT_{TM} \leq_m HO_{TM}$.

Esercizio: Riduzione da 3SAT a SUBSET-SUM

(a) Fornire la definizione di linguaggio NP-completo. Fornire la definizione di riduzione polinomiale.

(b) Data la seguente espressione booleana in 3-CNF

$$\phi = (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee x_3 \vee x_4)$$

si descriva l'immagine di $\langle\phi\rangle$ nella riduzione polinomiale di 3-SAT a $SUBSET$-SUM.

Lezione 32 pag. 17

NP - completezza

Vogliamo definire quando un linguaggio B è uno dei linguaggi «più difficili» della classe NP.

Abbiamo visto un modo per definire quando B è «più difficile» di A, ovvero quando A è di difficoltà «minore o uguale» a B:

$$A \leq_p B$$

Quindi B è uno dei linguaggi «più difficili» della classe NP.....

Definizione

Un linguaggio B è *NP-completo* se soddisfa le seguenti due condizioni:

1. B appartiene a NP
2. Per ogni linguaggio A in NP, $A \leq_p B$ (ovvero B è NP-hard)

Lezione 32 pag. 3

Riduzioni in tempo polinomiale

Definizione

Siano A, B linguaggi sull'alfabeto Σ.

Una riduzione in tempo polinomiale f di A in B è

- una funzione $f : \Sigma^* \to \Sigma^*$
- 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.

Lezione 31 pag. 4

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⟩.

$$(\overline{x}_1 \lor x_2 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor \overline{x}_3) \land (x_1 \lor \overline{x}_3 \lor \overline{x}_4) \land (\overline{x}_2 \lor x_3 \lor x_4)$$

Lezione 33-34(a) pagg. 57-58

Esercizio 3

Data la seguente formula booleana

$$\phi = (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

definire l'insieme S e l'intero t tali che $\langle S, t \rangle$ sia l'immagine di $\langle \phi \rangle$ nella riduzione polinomiale di 3-*SAT* a *SUBSET*-*SUM*.

Soluzione :

Numero	1	2	3	1	2	3	4
y_1	1	0	0	0	1	1	0
z_1	1	0	0	1	0	0	1
y_2	0	1	0	1	1	0	0
z_2	0	1	0	0	0	1	1
y_3	0	0	1	0	1	1	0
z_3	0	0	1	1	0	0	1
g_1	0	0	0	1	0	0	0
h_1	0	0	0	1	0	0	0
g_2	0	0	0	0	1	0	0
h_2	0	0	0	0	1	0	0
g_3	0	0	0	0	0	1	0
h_3	0	0	0	0	0	1	0
g_4	0	0	0	0	0	0	1
h_4	0	0	0	0	0	0	1
t	1	1	1	3	3	3	3

$S = \{y_1, z_1, y_2, z_2, y_3, z_3, g_1, h_1, g_2, h_2, g_3, h_3, g_4, h_4\}$,
$t = 1113333$

Prossimo tutorato

Ci vediamo mercoledì prossimo ore 14-16

sempre su questo canale del Team…

…buono studio ☺