14/12

TEO Se $A \le_m B$ e B decidibile, allora A è decidibile.

COR. Se $A \le_m B$ e A non è decidibile, allora B non è decidibile.

ESEMPI. *) $A_{TM} \le_m HALT_{TM}$.

Voglio costruire $f : \Sigma^* \to \Sigma^*$ t. c.

$f(\langle M, w \rangle) = \langle M', w' \rangle$ e

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

Definisco la mapping reduction.

$\underline{M'}$:

- Su x, esegue M su x. Se M accetta, M' accetta; se M rifiuta M' muove a DX per sempre (loop infinito).

$$\Rightarrow f(\langle M, w \rangle) = \langle M', w \rangle$$

Correttezza: f è CALCOLABILE.

($\Rightarrow$) $\langle M, w \rangle \in A_{TM}$; ovvero $M(w)$ accetta.
Siccome $M'(w)$ esegue $M(w)$ abbiamo che M' termina su w.

($\Leftarrow$) $\langle M, w \rangle \notin A_{TM}$; $M(w)$ rifiuta o loop.
Pertanto M' va sempre in loop. $\langle M', w \rangle \notin HALT$.

*) $E_{TM} \leq_m EQ_{TM}$.

Costruisco $f : \Sigma^* \to \Sigma^*$ t.c.

$f(\langle M \rangle) = \langle M_1, M_2 \rangle$ e

$\langle M \rangle \in E_{TM}$ sse $f(\langle M \rangle) \in EQ_{TM}$

$M_1 = M$ e M_2 è la TM che rifiuta sempre.

f è computabile.

($\Rightarrow$) Se $\langle M \rangle \in E_{TM}$, allora $L(M) = \emptyset$.

Inoltre $L(M_2) = \emptyset$ e quindi $L(M_1) = L(M_2)$

ovvero $\langle M_1, M_2 \rangle \in EQ_{TM}$.

($\Leftarrow$) Se $\langle M \rangle \notin E_{TM}$, allora $L(M_1) \neq \emptyset$.

Inoltre $L(M_2) = \emptyset$ e $L(M_1) \neq L(M_2)$ ovvero

$\langle M_1, M_2 \rangle \notin EQ_{TM}$. ✓

Usiamo le riduzioni per dimostrare anche che alcuni linguaggi NON sono Turing-riconoscibili.

TEO Se $A \leq_m B$ e B è Turing-riconoscibile, allora A è Turing-riconoscibile.

COR Se $A \leq_m B$ e A non è Turing-riconoscibile, allora B non è Turing-riconoscibile.

Idea: Sappiamo che $\overline{A_{TM}}$ non è Turing-riconoscibile.

LEMMA. Se $A \leq_m B$, allora $\bar{A} \leq_m \bar{B}$.

Dim. Esiste $f : \Sigma^* \to \Sigma^*$ t.c.

$$w \in A \quad \text{sse} \quad f(w) \in B$$

$$(w \in A \Rightarrow f(w) \in B \text{ e } w \notin A \Rightarrow f(w) \notin B.)$$

La mapping reduction è la stessa! $\forall w$:

$w \in \overline{A} \Rightarrow f(w) \in \overline{B}$ e

$w \notin \overline{A} \Rightarrow f(w) \notin \overline{B}$

TEO. EQ_{TM} non è Turing-riconoscibile.

DIM. Basta far vedere $A_{TM} \leq_m \overline{EQ_{TM}}$.

Per il lemma questo implica:

$$\overline{A}_{TM} \leq_m EQ_{TM}.$$

Costruisco $f: \Sigma^* \to \Sigma^*$ t.c.

$\langle M, w \rangle \in A_{TM}$ sse $f(\langle M, w \rangle) \in \overline{EQ_{TM}}$

$f(\langle M, w \rangle) = \langle M_1, M_2 \rangle$ dove M_1 rifiuta sempre e M_2 su ogni x esegue $M(w)$ e accetta sse M accetta w.

($\Rightarrow$) Se M accetta w ($\langle M, w\rangle \in A_{TM}$) allora $L(M_1) = \emptyset$ e $L(M_2) = \{0,1\}^*$ ovvero $L(M_1) \neq L(M_2)$ e $\langle M_1, M_2\rangle \in \overline{EQ_{TM}}$

($\Leftarrow$) Se $\langle M, w\rangle \notin A_{TM}$, allora M rifiuta w o loop. $L(M_1) = \emptyset = L(M_2)$. Ovvero $\langle M_1, M_2\rangle \in EQ_{TM}$ che è lo stesso che $\langle M_1, M_2\rangle \notin \overline{EQ_{TM}}$. ∎

ESERCIZI.

*) Dimostrare che il seguente L è indecidibile.

$$L = \{\langle M\rangle : M \in TM \text{ e } M \text{ accetta } w \in \{0,1\}^* \text{ con } |w| = \text{dispari}\}.$$

Costruisco una mapping reduction $A_{TM} \leq_m L$.
Ovvero una $f : \Sigma^* \to \Sigma^*$ t.c.

$<M, w> \in A_{TM}$ sse $f(<M, w>) \in L$

$f(<M, w>) = <M'>$

<u>M'</u>: - Su x, se $|x|$ è pari rifiuta
- Altrimenti esegue $M(w)$ e se M accetta (rifiuta), M' accetta (rifiuta) x.

f computabile. Poi:

($\Rightarrow$) $<M, w> \in A_{TM}$, ovvero $M(w)$ accetta.
M' accetta solo le stringhe x t.c. $|x| =$ dispari.
Ovvero, $<M'> \in L$.

($\Leftarrow$) $<M, w> \notin A_{TM}$, M rifiuta w o loop.

M' rifiuta sempre allora, ovvero $L(M') = \emptyset$ e quindi $\langle M' \rangle \notin L$.

*) $L = \{ \langle M \rangle :$ M è una TM e $L(M)$ sono stringhe $0^n 1^n 0^n : n \geq 0 \}$.

Non è decidibile. Costruisco $f : \Sigma^* \to \Sigma^*$ t.c. $\forall x \in \Sigma^*$:

$x \in A_{TM}$ sse $f(x) \in L$. $x = \langle M, w \rangle$

$f(\langle M, w \rangle) = \langle M' \rangle$.

L'idea: Costruisco M' t.c. se $M(w)$ accetta $L(M') = \{ 0^n 1^n 0^n : n \geq 0 \}$. E viceversa se $M(w)$ non accetta $L(M') \neq \{ 0^n 1^n 0^n : n \geq 0 \}$.

M': - Su x, simula M su w.
- Se M accettava $x = 0^n 1^n 0^n$ $n \in \mathbb{N}$, M' accetta x.
- Se M rifiutava w, M' rifiuta x.

($\Rightarrow$) Se $\langle M, w\rangle \in A_{TM}$, $M(w)$ accetta. M' accetta tutti gli $x = 0^n 1^n 0^n$ $n \in \mathbb{N}$. Ovvero $\langle M'\rangle \in L$.

($\Leftarrow$) Se $\langle M, w\rangle \notin A_{TM}$, $M(w)$ rifiuta o loop. M' rifiuta sempre e $\langle M'\rangle \notin L$.

*) Dimostrare: Se A è Turing-riconoscibile e $A \leq_m \bar{A}$ allora A è decidibile.
Siccome A è Turing-riconoscibile basta mostrare che A è coTuring-riconoscibile (ovvero $\bar{A}$ è Turing-riconoscibile).
Sappiamo che se $A \leq_m \bar{A}$, allora $\bar{A} \leq_m A$ e siccome A è Turing-ric. anche $\bar{A}$ lo è.

*) $L = \{ <T, T'> : T, T' \in TM$ e $L(T) \cup L(T') = \Sigma^* \}$

non è decidibile.
Riduzione su A_{TM}. Definisco $f : \Sigma^* \to \Sigma^*$ t.c. $<M, w> \in A_{TM}$ sse $f(<M, w>) \in L$.

$f(\langle M, w \rangle) = \langle T, T' \rangle$.

$\underline{T}$: - Su x, rifiuta.

$\underline{T'}$: - Su x, esegue M su w.

- Se $M(w)$ accetta (rifiuta), $T'(x)$ accetta (rifiuta).

($\Rightarrow$) $\langle M, w \rangle \in A_{TM}$, ovvero $M(w)$ accetta.

$$L(T) = \emptyset \text{ e } L(T') = \Sigma^*$$

$$\Rightarrow L(T) \cup L(T') = \Sigma^*$$

$$\Rightarrow \langle T, T' \rangle \in L.$$

($\Leftarrow$) $\langle M, w \rangle \notin A_{TM}$, M rifiuta w o loop.

$L(T) = L(T') = \emptyset$ e $\langle T, T' \rangle \notin L$.

ALMENO

*) $L = \{ \langle M \rangle : M \in TM$ e $L(M)$ contiene tutte le parole che iniziano per $0 \}$.

NON È DECIDIBILE

Costruisco $f : \Sigma^* \to \Sigma^*$ t.c. $\forall x$

$$x \in A_{TM} \text{ sse } f(x) \in L.$$

$$f(x) = f(\langle M, w \rangle) = \langle M' \rangle.$$

$\underline{M'}$: - Su x, esegue M su w.
- Se M accetta w, M' accetta x.
- Se M rifiuta w, M' rifiuta x

($\Rightarrow$) $\langle M, w \rangle \in A_{TM}$, allora $L(M') = \{0,1\}^*$
e $\{0,1\}^*$ contiene tutte le parole che iniziano per 0.
Ovvero $\langle M' \rangle \in L$.

($\Leftarrow$) $\langle M, w \rangle \notin A_{TM}$, $L(M') = \emptyset$ e $\langle M' \rangle \notin L$.

COMPLESSITÀ

Finora, abbiamo ignorato le risorse. Se L è decidibile, non significa che il corrispondente problema sia risolvibile praticamente.
Studiamo ora queste questioni per le TM.

- Capire le risorse necessarie per risolvere problemi.
- Classificare i problemi in base alle risorse necessarie a risolverli

Per noi: Tempo e spazio per le TM.

DEF. Il tempo di esecuzione o complessità di tempo di TM M è $f: \mathbb{N} \to \mathbb{N}$ t.c.

$f(n)$ è # di passi che M usa su ogni w di dimensione n.

Esempio: la TM M che su input w muove a Dx fino a che legge qualcosa ≠ da "$\sqcup$". Poi accetta.

Se w è lungo $n \geq 0$, M ha tempo di esecuzione $f(n) = n$.

Non ci interessa definire precisamente $f(n)$ ma ci basterà una stima asintotica.

<u>DEF</u>. Siano $f, g : \mathbb{N} \to \mathbb{R}^+$. $f(n) = O(g(n))$ se $\exists$ $c, n_0 \in \mathbb{N}$ t.c. $f(n) \leq c \cdot g(n)$ per ogni $n \geq n_0$.

<u>Es</u>. $n^3 + 100 n^2 + 6n = O(n^3)$.

DEF. $f, g : \mathbb{N} \to \mathbb{R}^+$. $f(n) = o(g(n))$ se $\lim_{n \to \infty} f(n)/g(n) = 0$. [Oppure, $\forall c \; \exists n_0$ t.c.

$$f(n) < c \cdot g(n) \quad \text{per } n \geq n_0.)$$

Es. Una TM per $L = \{0^k 1^k : k \geq 0\}$.

Abbiamo visto:

- Scansione input per controllare che abbia la forma $0^* 1^*$.
- Torna indietro e fa il seguente ciclo:
  - Ripete fino a che il nastro contiene almeno uno 0 e un 1: Scansione il nastro e cancella uno 0 con X e un 1 con X

- Se il nastro contiene solo x, accetta.
  Altrimenti rifiuta.

Tempo: $2n = O(n)$ all'inizio e alla fine.
Loop: $O(n/2)\,O(2n)$ perché ogni ripetizione richiede $O(2n)$ passi e ogni volta accoppio 0 con 1

$$O(n) + O(n^2) + O(n) = O(n^2).$$

DEF. Sia $t : \mathbb{N} \to \mathbb{R}^+$. $DTIME(t(n))$ è l'insieme di linguaggi decisi da TM deterministiche con tempo $O(t(n))$.

$L = \{0^k 1^k\} \in DTIME(n^2)$.

TEO Sia $t(n) \geq n$. Ogni TM multinastro con tempo $t(n)$ è simulabile da TM singolo nastro con tempo $O(t^2(n))$.

Dim. # nastri $k = O(1)$. Ci ricordiamo la simulazione della TM multinastro usando nastro singolo. In effetti:

- $O(n)$ per preparare il nastro.
- Per ogni passo della TM multinastro, la TM singolo nastro richiede $O(t(n))$ passi. (# passi è $t(n)$.)

Totale: $O(n) + O(t^2(n)) = O(t^2(n))$

DEF. Una NTM N decisore ha tempo di esecuzione $t(n)$ se il numero di passi che N

che N usa per ogni ramo di computazione è $\leq t(n)$ per input di lung. n.

TEO Sia $t(n) \geq n$. Ogni NTM N con tempo $t(n)$ ammette TM deterministica equivalente con tempo $2^{O(t(n))}$.

DIM. Ci ricordiamo la simulazione vista a lezione. Su x t.c. $|x| = n$ la lunghezza di ciascun cammino è $\leq t(n)$.

Se ogni nodo ha $\leq b$ figli ovvero le scelte non det. di N, # foglie $\leq b^{t(n)}$.

nodi $\leq 2 \cdot$ # foglie $= O(b^{t(n)})$.

$\Rightarrow$ Ricerca non ampiezza richiede tempo:

$$O(t(n) \cdot b^{t(n)}) = 2^{O(t(n))}.$$

DEF. $P = \bigcup_k DTIME(n^k)$

Questa corrisponde ai problemi risolvibili in maniera "efficiente"; ovvero polinomiale ($O(n^k)$) nella dimensione dell'input.

DEF $EXP = \bigcup_k DTIME(2^{n^k})$.

$PATH = \{ \langle G, s, t \rangle : G = (V, E)$ grafo orientato e esiste $s \rightsquigarrow t$ $\}$.

TEO $PATH \in P$.

DIM. Usiamo l'algoritmo che già abbiamo visto:

- Marca s.
- Ripeti fino a che non vengono marcati nuovi nodi: Scorriamo tutti gli archi e se c'è arco tra nodo marcato e nodo non marcato, marca quel nodo.
- Se t è marcato accetta, altrimenti rifiuta.

Fase iniziale e finale richiedono tempo polinomiale $O(n^k)$. Il loop è eseguito $|E| \le n$ volte perché ogni volta marca un nodo almeno. Ciascun passetto ha tempo di esecuzione polinomiale. ∎

Purtroppo non tutti i problemi sappiamo se sono in P. Esempio:

$$HAMPATH = \{\langle G=(V,E), s, t\rangle : G \text{ orientato e } \exists\ s \rightsquigarrow t \text{ HAMILTONIANO}\}$$

Cammino attraverso ogni nodo esattamente una volta! Osserviamo: HAMPATH $\in$ EXP perché posso provare tutti i cammini da s a t e verificare se sono HAMILTONIANI. Ma non sappiamo se HAMPATH $\in$ P.

Osservazione: HAMPATH è verificabile in tempo polinomiale. Ovvero se ti fornisco cammino $s \rightsquigarrow t$ si può verificain tempo polinomiale che s ~> t è HAMILTONIANO.

No.

DEF. Un verificatore per L è una TM V t.c. $L = \{ x : V \text{ accetta } \langle x, w \rangle \text{ per qualche } w \}$.

Il tempo di un V si misura in termini di x, ovvero V è polinomiale se eseguibile in tempo polinomiale in $|x|$.

DEF NP è la classe di linguaggi per cui esiste un verificatore di tempo polinomiale.

HAMPATH $\in$ NP.