5/4/2024 LEZ 9

M.d.t, PAROLE E NUM. NATURALI

→ LEZ OGGI: SPIEGAZIONE DI COME NON TUTTI I PROBLEMI NON POSSONO ESSERE RISOLTI V M.d.t.

TEOREMA

$$\boxed{\text{INSIEME DI } |M.d.t| < \text{INSIEME } |\text{LINGUAGGI}|}$$

DIM → concetto simile a cantor, dispensa 4.

- PRIMA DI TUTTO DOBBIAMO DIM. CHE $|M.d.t|$ È NUMERABILE → $\exists$ f BIUNIVOCA TRA $|M.d.t| \to \mathbb{N}$ (contabile)

- CONS. LE M.d.t. COME DESCRIZIONE:

  $T: b(q_0) - b(q_1) \otimes b(q_{11}), S_{11}, S_{12}, b(q_{12}), D \otimes$

  così & begin 1 0 → $9\ b(q_0)\ 2\ b(q_1)\ 3\ b(q_{11})\ 5\ S_{11}\ 5\ S_{12}\ 5\ b(q_{12})\ 5\ 6\ 7.$

  b(q) codifi. binaria

  POSSIAMO SCRIVERLO COME NUMERO, QUINDI POSSIAMO ASSEGNARGLI UN VALORE $\in \mathbb{N}$ → OGNI M.d.t. UNICA QUINDI È NUMERABILE.

- ANCHE LE PAROLE SONO NUMERABILI (5.1), ALLORA FACCIAMO UN GIOCO... (DISP. 5)

- POSSO FARE UNA MATRICE CON P = PAROLE (COLONNE), T_h (RIGHE)

	P_1	P_2	P_3	P_4	P_5	....
T_{h1}	1	0	1	1	0	→
T_{h2}	0	0	1	0	1	
T_{h3}	1	1	1	0	0	
T_{h4}	0	0	1	0	1	
⋮						

L

$$\begin{cases} 1 = P \text{ ACC. DA } T \\ 0 = P \notin \text{ ACC. DA } T \end{cases}$$

- OGNI RIGA È UN LINGUAGGIO
- LA DIAGONALE PUÒ ESSERE UN L.

PRENDIAMO QUINDI

• $L_D = 1010 \xrightarrow{\text{SIA}} L_{\bar{D}} = 0101$

$\boxed{\text{ESISTE UNA } T_{h_i} \text{ CHE RICONOSCE } L_D\text{?}}$ NOPE

DATO CHE $\forall T_{h_i}\ \exists P_i \in L_{\bar{D}} \wedge T_{h_i}(P_i) = q_R$ X COSTRUZIONE OF $L_{\bar{D}}$

• L_D È UN LINGUAGGIO NON ACCETTABILE.

QUINDI ↳ TECNICA DELLA DIAGONALIZZAZIONE

[|M.d.t| < |LINGUAGGI| → NON TUTTI I LINGUAGGI SONO RICON. DA M.d.t.

MA COME POSSIAMO COSTRUIRE UN LINGUAGGIO DEL GENERE?

VEDIAMO COSA TURING VOLEVA DIMOSTRARE CON LA MACCHINA, CIOÈ SE ∃ UN PROCEDIMENTO CHE RICONOSCERE SE UN PROBLEMA È RISOLVIBILE O NO.

X CAPIRE..

SIA

$$L_H = \{ \langle i, x \rangle \in \underbrace{\mathbb{N} \times \mathbb{N}}_{} : i = \text{COD. MACCH. DI TURING} \wedge T_i(x) \text{ TERMINA} \}$$

TUTTI E 2 POSSIAMO RAPP. COME

TURING DIMOSTRÒ CHE L_H NON È DECIDIBILE

non esiste un programma che ci dica se la amcchina (codice) termina o no

DIM

1) L_H È ACCETTABILE

∃ MACC. TURING CHE ACC. L_H.

CONOSCIAMO GIÀ UNA MACCHINA SIMILE

T_U -> MACC. UNIVERSALE

- SIA U^i: INPUT i, x DOVE

(1) VERIFICA SE i È COD DI UNA M.d.t: SE NO -> q_R

(2) SIMULA $U(i,x)$: SE TERMINA IN q_A o q_R

$\Downarrow$

ACCETTA

L_H ACC. MA NON DECIDE, POICHÉ:

- SE i $\notin$ M.d.t, RIGETTA -> OK

MA - SE $x \notin L$ E $U(i,x)$ $\not$ TERMINA -> L_H NON TERMINA.

$\Downarrow$

(2) L_H È NON DECIDIBILE

DIM X ASSURDO

SUPPONIAMO CHE $\exists$ T CHE DECIDE L_H

OSSIA $\forall$ COPPIA $\langle i,x \rangle$ $\left[T(i,x) = q_A \text{ SE } \langle i,x \rangle \in L_H \right.$

q_R SE // // $\notin L_H$

SE T $\exists$, DA T POSSO COSTRUIRE UNA MACCHINA T', E.C.

- T'(i,x) SIMULA T(i,x) E ASPETTA CHE TERMINA

- SE T(i,x) ACC. -> T'(i,x) TERMINA IN q_R

- SE // RIFIUTA -> // // // q_A

LA MACCHINA COMPLEMENTARE QUINDI

DA T' COSTRUISCO T'', DOVE SIMULA T'(i,x), MA:

- SE T'(i,x) FINISH IN q_A -> ANCHE T''(i,x) VA IN q_A

- SE T'(i,x) FINISH IN q_R -> T'(i,x) LA MANDO IN LOOP

DA T^H CREO $T^*(i)$ (1 INPUT), DOVE:

$$T^*(i) = T^H(i,i) = \begin{cases} \nexists \text{ FINISH IF } \langle i,i \rangle \in L_H \\ q_A \text{ IF } \langle i,i \rangle \notin L_H \end{cases}$$

QUINDI

SE $T \; \exists \rightarrow T^* \; \exists \rightarrow \exists \; k \in \mathbb{N} : T^* = T_k$

MA ALLORA:

$$T_k(k) = T^*(k) \, ?$$

2 CONTRADDIZIONI

- SE $T_k(k) = T^*(k)$ NON TERMINA $\rightarrow \langle k,k \rangle \in L_H$
  $\Downarrow$
  $T_k(k)$ TERMINA

- SE $T_k(k) = T^*(k) = q_A \Rightarrow \langle k,k \rangle \notin L_H \rightarrow T_k(k)$ NON TERMINA.

QUINDI $\nexists$ NESSUNA DI QUESTE MACCHINE

$\Downarrow$

L_H NON È DECIDIBILE $\blacksquare$