# INDECIDIBILITÀ

Il risultato fondamentale che mostreremo è che come ci sono linguaggi:

- Non sono regolari
- Non sono acontestuali

ci sono anche linguaggi non Turing-dec. / ric.

Il concetto di algoritmo mediante con Turing.

Nel 1900 Hilbert pose questo problema:

"processo in base al quale il prob. può

essere risolto in numero finito di passi" per questo test: determinare se un polinomio ha radici intere.

Es.: $6X^3 + YZ^2 + 3XY^2 - X^3 - 10$

$$X = 5; \quad Y = 3; \quad Z = 0$$

In termini di calcolabilità:

$$L = \{ \langle p \rangle : p \text{ polinomio con radici intere} \}$$

è DECIDIBILE / RIC. La motivazione

$\langle \cdot \rangle$ significa una qualsiasi codifica di un polinomio $p(\cdot)$. Nel 1970 è stato dimostrato che L non è DECIDIBILE. Nell'ottica della tesi Church-Turing: esistono problemi che nessun computer può risolvere.

Iniziamo con esempi di problemi decidibili.

Esempio: stabilire se DFA D accetta una stringa. Corrisponde:

$$A_{DFA} = \{ \langle D, w \rangle : D \in DFA \text{ e } D \text{ accetta } w \}.$$

Come è fatta una codifica <D, w> ?

$$\delta(q, a) = p$$

In altri termini devo codificare $(Q, \Sigma, \delta, q_0, F)$. Ad es. una rapp. di $\delta$ è l'elenco di tutte le regole della tabella per $\delta$ separate da "#".

Una codifica può essere VALIDA o NO.

Vediamo come decidere $A_{DFA}$:

- Su input <D, w> controllo che sia una codifica valida di $D \in DFA$.

Se no, rifiuto.

- Altrimenti, simulo $D$ su $w$. Wlog, assumo la TM ha un nastro contenente $\delta$, un nastro per input $w$, un nastro per tracciare lo stato.
- Se la simulazione termina su stato accettante, accetta altrimenti rifiuta.

$$\langle D, w \rangle$$

$$\langle \delta \rangle : (q_0, w_1) \to p \ \cdots$$

$$w = w_1 \, w_2 \ \cdots \ w_n$$

$$\cancel{q_0} \ p$$

$$\forall q \in Q, \ \forall a \in \Sigma$$

$$\langle \delta \rangle : \ q\text{''}+\text{''}a\text{''}\to\text{''}p\#\ \cdots$$

Se ho $|Q|$ stati per codificare

$q \in Q$ necessito di $\log_2 |Q|$ simboli

se $\Gamma = \{0, 1\}$

Altro esempio:

$A_{NFA} = \{<N, w> : N \in NFA$ e $N$ accetta $w\}$

Per decidere $A_{NFA}$:

- Prima traduco $<N, w>$ in $<D, w>$ tale che $D \in DFA$

e $L(D) = L(N)$.

- Dop. lancio la TM per $A_{DFA}$.

Stesso principio:

$$A_{REX} = \{ \langle R, w \rangle : R \text{ espressione regolare che genera } w \}$$

$A_{REX}$ è TURING-DECIDIBILE. Basta tradurre $\langle R, w \rangle$ in $\langle D, w \rangle$ t.c. $L(D) = L(R)$.

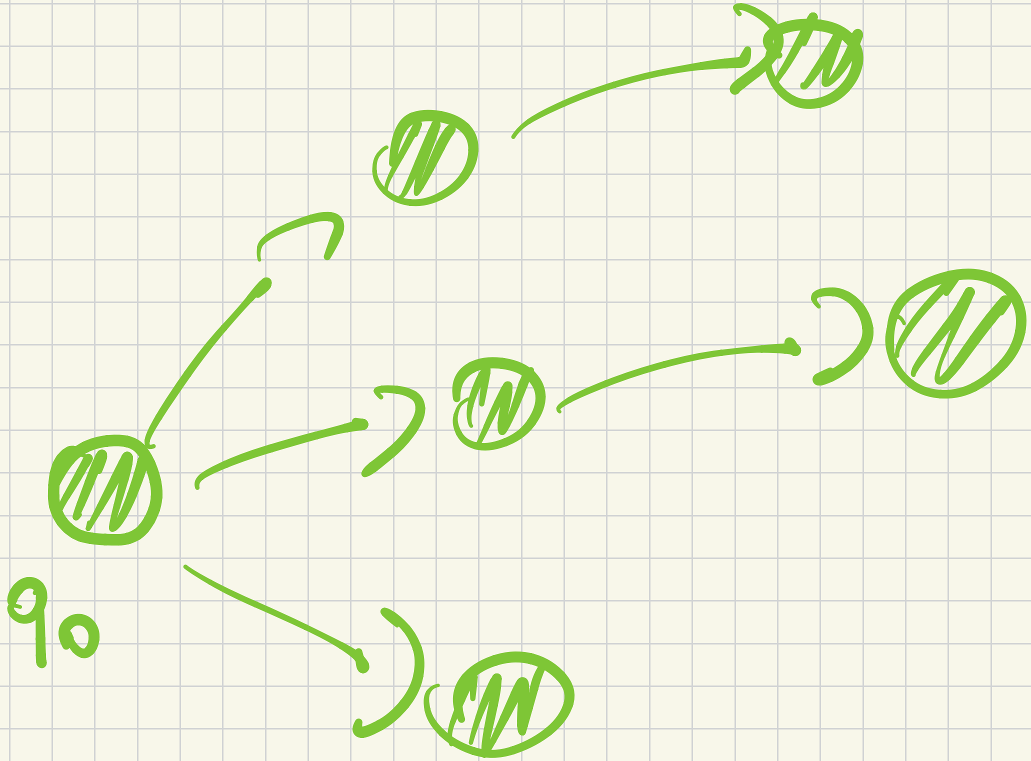
Altro esempio:

$$E_{DFA} = \{\langle D \rangle : D \in DFA \text{ e } L(D) = \emptyset\}$$

Problema: la TM non può provare tutti i $w \in \Sigma^*$.

Tuttavia posso ridurre questo problema a quello di cercare un cammino $q_0 \rightsquigarrow q$ t.c. $q \in F$ nel diagramma di stato per $D$:

- Dato <D> rifiuto se non è una codifica valida.
- Altrimenti, ripeto fino a che non vengono marcati nuovi stati:
  - Marco tutti gli stati che hanno transizione provenienti da stato marcato
- Se esiste $q \in F$ marcato allora rifiuto. Altrimenti accetto.




$\exists\ \varphi \in \Gamma$ marcato ???

Altro esempio:

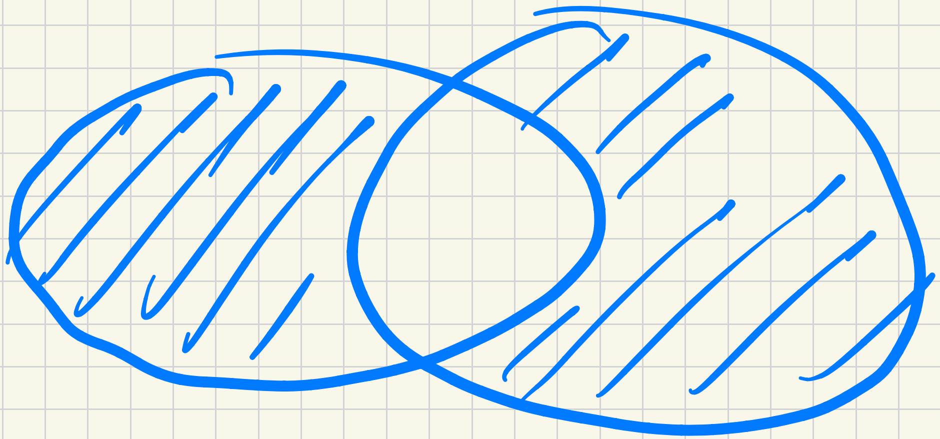
$$EQ_{DFA} = \{ \langle A, B \rangle : A, B \in DFA \text{ e } L(A) = L(B) \}.$$

$L(A) = L(B)$ se e solo se

$$L(A) \,\Delta\, L(B) = \emptyset$$

$$\parallel$$

$$\left( L(A) \cap \overline{L(B)} \right) \cup$$



$(L(B) \cap \overline{L(A)})$

Per decidere $EQ_{DFA}$: Definisco $C$ t.c. $L(C) = L(A) \triangle L(B)$.

Poi uso la TM per decidere se $\langle C \rangle \in E_{DFA}$.

Es. Dimostrare che

$$A_{CFG} = \{ \langle G, w \rangle : G \in CFG \text{ t.c. } w \in L(G) \}$$

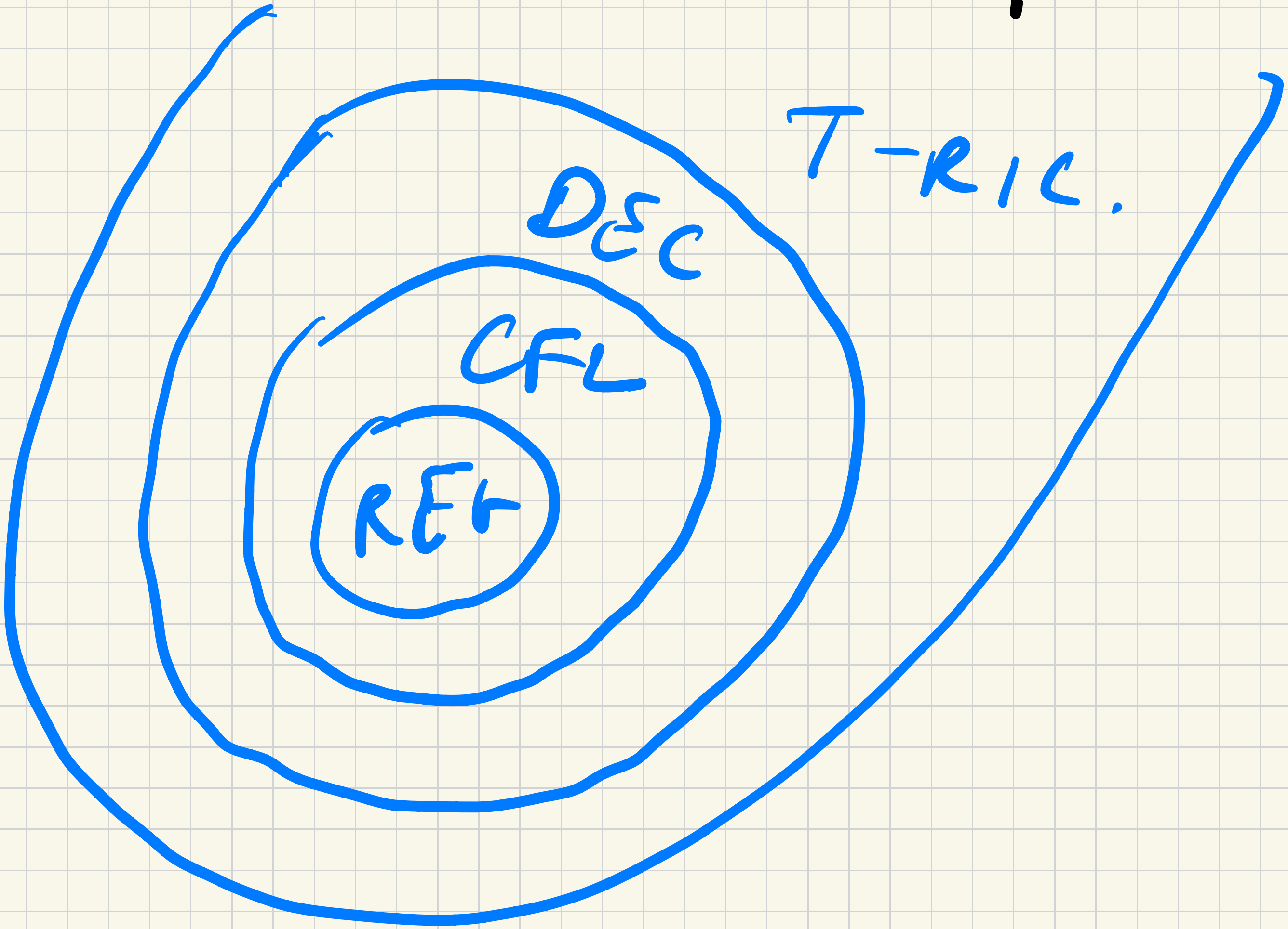
È DECIDIBILE.

Oppure:

$$E_{CFG} = \{ \langle G \rangle : G \in CFG \text{ e } L(G) = \emptyset \}$$

È DECIDIBILE.

In pratica la situazione è questa:




$$REG \subset CFL \subset DEC \subset T\text{-}RIC$$

Vediamo ora che esistono linguaggi

che non sono TURING-DEC. / RIC.
La tecnica che si usa è quella della DIAGONALIZZAZIONE.

**TEO**. $A_{TM} = \{ <M, w> : M \in TM$ e $M(w) = Acc \}$

non è DECIDIBILE.
Prima di dimostrarlo, vediamo che $A_{TM}$ è TURING RICONOSCIBILE. Infatti consideriamo la seguente TM $U$:

- Su input $\langle M, w \rangle$ con $M \in TM$ e $w \in \Sigma^*$
- Simula $M$ su input $w$
- Se $M$ accetta, allora accetta l'input, altrimenti rifiuta.

Se $M$ va in loop, anche $U$ andrà in loop (riconoscitore). Se $U$ potesse capire se $M$ va in loop o meno sarebbe decisore,

ma questo come vedremo è anche simula=
bile.

Unica cosa da chiarire: Come funziona $U$?

Ad es.: Uso due nastri, sul primo salvo $<M, w>$; sul secondo tengo traccia della configurazione $<(a, q, b)>$ di $M$ dove $q$ è lo stato, $(a, b)$ è il nastro e la testina punta sul primo carattere di $b$.

La codifica: $<M, w>$. Sia $M =$

$(\Sigma, \Gamma, Q, \delta, q_0, q_{ACC}, q_{REJ})$ e sia

$n = |\Sigma|$ e $m = |\Gamma|$, $s = |Q|+3$.

Denotiamo con $(i)_2$ la codifica binaria di $i$; e $\langle \delta \rangle$ la codifica di $\delta$ come sequenza di regole $\langle R \rangle$ separate da ",".

$$R = ((q, a), (r, b, Z))$$

$$Z \in \{L, R\}$$

$$\langle M, w \rangle = (n)_2, (m)_2, (s)_2, \langle \delta \rangle ; w$$

$$< \delta > = < R_1 >, < R_2 >, \dots$$

La simulazione: Nastro 1 contiene $<M, w>$ e nastro 2 vuoto. Poi nel nastro 2 scrivo $< q_0, w >$.

Al passo $t$: Nastro 1 contiene $<M, w>$ e nastro 2 contiene configurazione

$$< (a, q, b_1, \underbrace{b_2 \dots b_l}_{b'}) >$$

ovvero $b = b_1 b'$

Cerco sul nastro 1 una regola del tipo: $\langle (q, b_1), (r, y, z) \rangle$ e aggiorno il nastro 2 con la nuova configurazione.

Ad ogni passo: controllo se $r = q_{ACC}$ e se lo è ACCETTO. Se $r = q_{REJ}$ invece rifiuto.

(La configurazione $(a, q, b)$ significa che lo stato è $q$, nastro $= ab$ e

Terminare su primo carattere di b. )

Dimostriamo $A_{TM}$ È INDECIDIBILE.

DIM. Sia H una TM che DECIDE $A_{TM}$. Ovvero:

$$H(\langle M, w \rangle) = \begin{cases} \text{ACC se } M \text{ accetta } w \\ \text{REJ se } M \text{ rifiuta } w \\ \text{(oppure LOOP)} \end{cases}$$

Costruisco una nuova TM D che

uso H.

TM D:

- Su input $\langle M \rangle$
- Esegue H su $\langle M, \langle M \rangle \rangle$
- Se H ACCETTA, D RIFIUTA

e VICEVERSA

| | $\langle M_1 \rangle$ | $\langle M_2 \rangle$ | $\langle M_3 \rangle$ | $\langle M_4 \rangle$ | $\cdots$ | $\langle D \rangle$ |
|---|---|---|---|---|---|---|
| $M_1$ | ACC | | | | | |
| $M_2$ | | ACC | | | | |
| $M_3$ | | | REJ | | | |
| $M_4$ | | | | REJ | | |
| $\vdots$ | | | | | | |
| $D$ | REJ | REJ | ACC | ACC | | ??? |

De une parte :

$$D(\langle M \rangle) = \begin{cases} ACC & \text{sse } H \text{ REJ } \langle M, \langle M \rangle \rangle \\ & \text{sse } M \text{ REJ } \langle M \rangle \\ REJ & \text{sse } H \text{ ACC } \langle M, \langle M \rangle \rangle \\ & \text{sse } M \text{ ACC } \langle M \rangle \end{cases}$$

Quando eseguo $D(\langle D \rangle)$?

$$D(\langle D \rangle) = \begin{cases} ACC & \text{sse } D \text{ REJ } \langle D \rangle \\ REJ & \text{sse } D \text{ ACC } \langle D \rangle \end{cases}$$

CONTRAD-
DIZIONE

$\rightarrow\leftarrow$ Ma $D$ è un DECISORE esso stesso. Manca via d'uscita:

H non chiude ovvero $A_{TM}$ non

è DECIDIBILE. $\blacksquare$