Iniziamo con esempi di linguaggi decidibili.

$$A_{DFA} = \{ \langle D, w \rangle : D \text{ è un DFA}, w \in L(D) \}$$

TEO A_{DFA} è DECIDIBILE.

Dim. Occorre esibire una TM M decisore tale che M accetta tutti e soli gli input $\langle D, w \rangle \in A_{DFA}$.

Ad alto livello:

1) Su input $\langle D, w \rangle$, interpreta D come DFA e w come stringa. (Se fallisce, rifiuta.)

2) Simula esecuzione di D su w

3) Se D accetta w, allora M accetta altrimenti rifiuta.

La codifica: $D = (Q, \Sigma, \delta, q_0, F)$ contiene tutti gli elementi in sequenza, ed es. separati da ",". La δ la posso vedere come una sequenza di transizioni separate da "#".

Per simulare D posso usare 2 ulteriori nastri, uno contenente w ed uno contenente lo stato attuale di D (inizio q_0).

Ad ogni passo leggo lo stato e il carattere in input e scorriamo il primo nastro per eseguire la transizione specificata da $<\delta>$.

$\blacksquare$

Similmente: $A_{NFA} = \{ <N, w> : N \in NFA$
$w \in L(N) \}$

Idea implementativa:

1) La TM M trasforma $<N, w>$ in $<D, w>$ t.c. $D \in DFA$ e $L(D) = L(N)$. (Come studiato in precedenza.)

2) La TM M esegue i passi della precedente dimostrazione su $<D, w>$.

Altro esempio: $A_{REX} = \{\langle R, w \rangle : R \in REX$ ed R genera $w\}$

Idea implementativa:

1) La TM converte R in NFA equivalente come dimostrato a lezione.

2) La TM esegue la TM di sopra su $\langle N, w \rangle$.

Esempio interessante: il test del vuoto.

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

TEO E_{DFA} è DECIDIBILE.

DIM. Associo al DFA D un grafo (il diagramma di stato); gli stati sono i vertici e c'è un arco tra due vertici sse c'è una possibile transizione tra gli stati corrispondenti.

La TM:

1) Marca lo stato iniziale.

2) Ripete fino a che non vengono più marcati stati: Marca ogni stato che ha una transizione proveniente da uno stato già marcato.

3) Se almeno uno stato in F è marcato

rifiuta; altrimenti accetta. ∎

Altro esempio:

$$EQ_{DFA} = \{ \langle D_1, D_2 \rangle : D_1, D_2 \in DFA,$$
$$L(D_1) = L(D_2) \}.$$

TEO. EQ_{DFA} è DECIDIBILE.

DIM. Idea: consideriamo la Δ simmetrica tra $L_1 = L(D_1)$ ed $L_2 = L(D_2)$.

$$L_1 \Delta L_2 = (L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1})$$

$$L_1 = L_2 \iff L_1 \Delta L_2 = \emptyset$$

La TM: Dato $\langle D_1, D_2 \rangle$ definisce $\langle D \rangle$ t.c. $L(D) = L_1 \Delta L_2$.
Poi eseguo la TM per E_{TM} su $\langle D \rangle$

Esempio simile con le grammatiche.

$$A_{CFG} = \{ \langle G, w \rangle : G \in CFG, \ G \text{ produce } w \}.$$

TEO. A_{CFG} è DECIDIBILE.

Dim. Se G è CFG, posso assumere wlog che G è in CNF.

Fatto: Se $w \in L(G)$ e $|w| = n$ allora # passi per generare w è $2n-1$.

La TM: partendo da S prova tutte le derivazioni di lunghezza $2n-1$ dove $|w| = n$. Se una di esse produce w, accetta, altrimenti rifiuta.

Prova del fatto: Se $n = 1$; è vero perché $2 \cdot n - 1 = 1$.

Per induzione se è vero fino a $n-1$, e lo dimostro per $n \geq 2$.

$$S \to AB$$

$$A \overset{*}{\Rightarrow} u \quad ; \quad B \overset{*}{\Rightarrow} v \qquad w = uv$$

$$|w| = n$$

$$|u| = k \quad ; \quad |v| = n-k$$

$$k \geq 1$$

\# passi per w: $1 + (2k-1) + 2(n-k) - 1$

$$= \cancel{1} + \cancel{2k} - \cancel{1} + 2n - \cancel{2k} - 1$$

$$= 2n - 1 \quad \checkmark$$

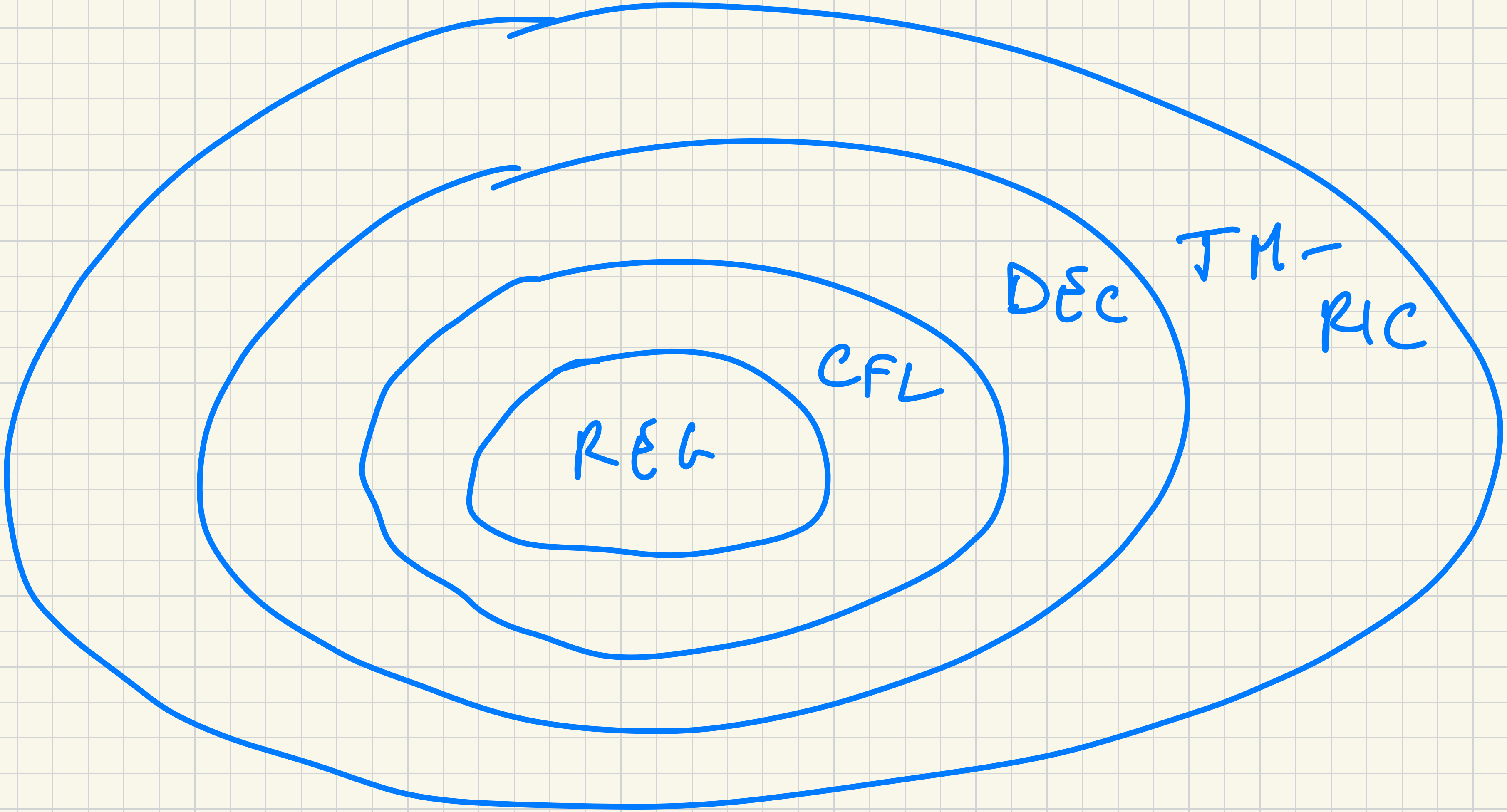
$$E_{CFG} = \{ \langle G \rangle : G \in CFG \\ L(G) = \emptyset \}$$

La TM:

1) Data $\langle G \rangle$ marca tutti i terminali
2) Ripete fino a quando non vengono marcate nuove variabili: Marca la variabile A t.c. in G c'è la regola $A \to V_1 \dots V_k$ con $V_1, \dots, V_k$ marcati.
3) Accetta sse S non è marcata.

$$EQ_{CFG} = \{ \langle G, H \rangle : G, H \in CFG, \ L(G) = L(H) \}$$

$\Rightarrow$ non è decidibile!



Vogliamo mostrare che esistono linguaggi NON DECIDIBILI. Ad es.:

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

TEO. A_{TM} è Turing-riconoscibile.

Dim. Simulo esecuzione di M su input w. Se M accetta, accetto. Se M rifiuta, rifiuto. Ovviamente, se M andrà in loop, andrò in loop.

Dettagli implementativi: la TM universale

U, ovvero la TM che simula esecuzione di un'altra TM M su input w. Avrà due nastri: (i) nastro 1 contiene $\langle M, w \rangle$; (ii) contiene la codifica $\langle (a, q, b) \rangle$ della configurazione attuale di M.
(Come definito precedentemente: $a \| b$ il contenuto del nastro, q lo stato, la testina legge il primo carattere di b.)
La codifica $\langle M, w \rangle$, dove $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$. Siano: $n = |\Sigma|$; $m = |\Gamma|$; $s = |Q|$; avremo

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

"," e ";" sono caratteri speciali

$(n)_2$ è la rapp. binaria di n

$$\langle \delta \rangle = R_1 ; R_2 ; R_3 ; \ldots$$

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

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

La TM U:

1) Inizializza sul nastro 2 la configurazione iniziale $\langle (q_0, w) \rangle$.

2) Al passo t: sia $\langle (a, q, b) \rangle$ la configurazione sul nastro 2.

(i) Estrae q dalla configurazione;

(ii) Scansione il primo nastro in cerca di regola $\langle (q, x), (r, y, z) \rangle$.

Se $x \neq b[1]$, passo alla regola successiva.

Se $x = b[1]$ aggiorno la configurazione sul nastro 2 in accordo alla regola letta.

3) Se il nastro 2 contiene q_{acc} o q_{rej} accetta o rifiuta di conseguenza.

Se $w \in L(M)$, U accetta sicuramente.

Se $w \in L(M)$, U rifiuta o va in loop e quindi U è un riconoscitore. ☒

Per dimostrare che A_{TM} non è DECIDIBILE introduciamo una tecnica: la DIAGONALIZZAZIONE.

Prima di applicarla alle TM, vediamo come si usa per dimostrare alcune semplici proprietà degli insiemi. Ad es.: $\mathbb{R}$ non è numerabile.

Numerabile: A è numerabile se $\exists$ una biezione $f: \mathbb{N} \to A$. L'insieme dei numeri pari è numerabile $\hookrightarrow \mathbb{E}$

$\exists\ f: \mathbb{N} \to \mathbb{E}$ una biezione

$$f(n) = 2n$$

Anche $\mathbb{Q}$ è numerabile.

$\mathbb{R}$ non è numerabile. Supponiamo lo sia, ovvero $\exists\, f : \mathbb{N} \to [0,1]$. Faremo vedere che esiste $\alpha \in [0,1]$ che non è accoppiato con nessun elemento di $\mathbb{N}$. Pertanto f non può esistere, ovvero $[0,1]$ non è numerabile.

$$f(1) = 0{,}5105 11\ldots$$
$$f(2) = 0{,}413217\ldots$$
$$f(3) = 0{,}824345\ldots$$
$$f(4) = 0{,}732055\ldots$$

La cifra n-sima di d è diversa dalla i-sima cifra decimale di $\overline{f(i)}$.

Es.: $d = 0,437 \ldots$

$4 \neq 5$; $3 \neq 1$; $7 \neq 4$; ...

Affermo che d non può essere nella lista $f(1), f(2), f(3), \ldots$ Supponiamo lo sia: $\exists\, k \in \mathbb{N}$ t.c. $f(k) = d$. Ma allora la k-sima cifra di d è uguale alla k-sima cifra di $f(k)$ e questo contraddice la definizione di d. $\boxtimes$

Con la stessa tecnica: l'insieme $\mathcal{B}$ di tutte le stringhe di lunghezza infinita anche non è numerabile.

TEO. Esistono linguaggi che non sono decidibili e che non sono Turing-riconoscibili.

DIM. L'insieme delle TM è NUMERABILE, ma l'insieme dei linguaggi è NON NUMERABILE.

Prendiamo Σ^*; notare che $\{\langle M\rangle : M \in TM\} \subseteq \Sigma^*$. Facciamo vedere che Σ^* è numerabile. Questo implica che l'insieme delle TM è numerabile.

Ordino le stringhe su Σ^* secondo ordinamento "$<$":

- Sia $<_l$ l'ordinamento tra i caratteri di Σ esteso a parole lunghe l. (l'ordine alfabetico o lessicografico.)

- $x < y$ sse $\begin{cases} |x| < |y| \\ |x| = |y| \text{ e } x <_l y \end{cases}$

  $x, y \in \Sigma^*$

È un ordinamento totale: ovvero posso disporre gli elementi di Σ^* su una linea.

$$x_1 < x_2 < x_3 < x_4$$

$$\Sigma^* = \{ x_1, x_2, x_3, \dots \}$$

La funzione $f : \mathbb{N} \to \Sigma^*$

$f(i) =$ i-esimo elemento nell'ordinamento "<" su Σ^*.

$\Rightarrow \Sigma^*$ è numerabile.

L'insieme di linguaggi invece NO. Associo ad ogni linguaggio una stringa di lung. infinita.

$$\Sigma^* = \{ s_1, s_2, s_3, \dots \} \qquad s_1 < s_2 < s_3$$

$$\Sigma^* = \{ \ \varepsilon \quad 0 \quad 1 \quad 00 \quad 01 \quad 10 \quad 11 \quad 000 \ \dots$$

$$L = \{ \quad 0 \quad \quad 00 \quad 01 \quad \quad \quad 000 \ \dots$$

$$\chi_L = 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \ \dots$$

$\Rightarrow \exists$ biezione $f : \mathcal{L} \to \mathcal{B}$ dove $\mathcal{L}$ è insieme linguaggi; $\mathcal{B}$ è insieme di stringhe di lunghezza infinita

$$f(L) = \chi_L \qquad \blacksquare$$