- DECIDIBILITÀ
- ACCENNI DI INDECIDIBILITÀ
- ESERCIZI VARI

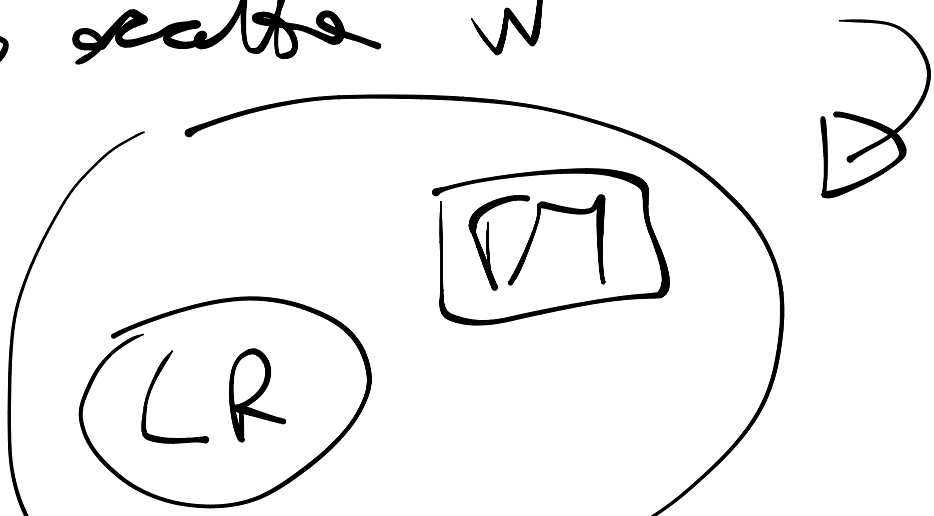
- DFA
- ER

→ A_{NFA} :

$\langle B, w \rangle$ = B ε-NFA

w stringa

B accetta w



M = INPUT $\langle B, w \rangle$

↑ ε-NFA

① Trasformo in DFA
con costr. sottoinsiemi

② Eseguo M su $\langle B, w \rangle$

③ Se M accetta, accetta
Se M rifiuta, rifiuta

HALTING PROBLEM

↓

A_{TM} → esegue il problema dell'arresto

└ M → $\langle w \rangle$ → LOOP
→ accetta /

rifiuta

$EQ_{DFA} = \{(A,B) \mid A$ e B sono DFA e $L(A) = L(B)\}$

→ CHIUSI RISPETTO regolari
ALLA DIFF. SIMMETRICA

→ M sul problema
→ A e B sono regolari
→ M accetta, accetta

$\boxed{\text{Descrizione}}$

M / N

M esegue su tutto il nostro A

N esegue su tutto il nostro B

∩

Intersezione

$M \cap N$ → accetta se
M e N accettano,
altrimenti rifiuta

$\boxed{CFL}$ → Chomsky

→ FNC → $(2n-1)$

$INFINITO_{DFA} = \{(A) \mid A$ è un DFA
e $L(A)$ è un linguaggio infinito$\}$

$INFINITO_{DFA}$ → decidibile

$\{ \neq, \sqcap, \ldots \}$

→ infinito dal punto di
vista di codifica

→ A_{DFA} → posso usare
problemi noti

T → Sui input

K → n. di stati

DFA D → accetta le stringhe di lunghezza "K" o più

DFA M → Esegue la rimanente parte di input

$$L(T) = L(A) \cap L(D)$$

→ T accetta se accetta, altrimenti rifiuta

→ Da un certo punto in poi, decide

$A = \{ \langle M \rangle \mid M$ è un DFA

che non accetta nessuna
stringa contenente
un #(n.) dispari di 1}
→ [DECIDIBILITÀ]

[A_{DFA} / A_{REX} / A_{CFG}
A_{PDA}]
→ modello con n. finito
di passi

DFA → accetta un n.
dispari di 1

M → Su input <A>

M → Esegue (A)
- Ne prendiamo il

complemento ⟨A⟩

- accetta se $\bar{A}$ accetta, rifiuta altrimenti

M → ⟨A⟩

→ esegue e prende il M. disponi di 1

→ A accetta, rifiuta altrimenti accetta

$\boxed{A_{DFA}}$ ← usiamo un modello che sappiamo decidibile per eseguire il nostro algoritmo

- $ALL_{\varepsilon}CFG = \{\langle G\rangle \mid G$ è una CFG che genera

$\varepsilon, \Sigma \in L(G)\}$

→ G in FNC (conversione)

CFG – PDA

→ PDA (convertiamo la CFG in PDA)

(Potrei usare A PDA)

→ Se M accetta, accetta altrimenti infinito

NFA → sottoinsiemi (DFA = NFA)

INDECIDIBILITÀ

→ HALTING PROBLEM

(se il programma si ferma su tutti gli input)

→ DIAGONALIZZAZIONE

$x_1 \quad x_2 \quad x_3$

$x_1 \quad x_2 \quad x_3$

$\mathbb{R}$ è infinito

Non per forza arriveremo alla fine → non sempre in grado di contare e di eseguirlo in # passi finiti

$A_{TM} = \{ \langle M, w \rangle \mid M$ accetta la stringa $w \}$

A_{TM} si ferma su un certo n. di passi, ma non per forza termina

$H \langle M, w \rangle$ — accetta se M rifiuta viceversa

[accetta accetta
rifiuta rifiuta]

$\underbrace{A_{TM}}$ / $\underbrace{\bar{A}_{TM}}$

indecidibile

→ uso questi due modelli per descrivere il mio

problema come indecidibile

B = <M, w> | M è una TM a due nastri che scrive un simbolo non vuoto sul secondo nastro eseguendo w }. Formularlo come problema e dire che sia indecidibile

T → input <M, w>
⇒ M per costruire T come TM a due nastri
T < input x >
→ M su x (simbolo)
→ M accetta

Created by Notein

⇒ scrivo un simbolo non usato sul secondo nastro

$T \Rightarrow \langle M, w \rangle$

verifichiamo che M esegue correttamente

→ verifica se M scrive un simbolo non usato sul nastro

→ INDECIDIBILE

→ Possiamo usare A_{TM} per "decidere" questo problema

A_{TM} → va in loop o non si ferma la mia M

(come usare per descritto)

R -L TM

→ TM e nastro semi-infinito

→ 2 mosse. | [. . . . . → (∞)

→ (R3) → spostarsi a dx di 3 celle

→ (L2) → spostarsi a sx di 2 celle

→ fare in modo di farla restare sempre dentro al nastro

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L3, R2\}$$

(1)

$\delta(q, e) \to$ CODIFICARE

INIZIO → $\delta(q_0, e)$

→ INIZIO NASTRO $\triangleright$

(L2 / R3)

MOVIMENTO PER "ASSICURARSI" CHE NON ESCA DAL NASTRO

$\delta(q, a) = (r, b, R)$

→ 3 celle a dx

2 celle a sx

→ prosegue in questo modo e mantiene l'ordine

→ raggiunge lo st. di accett → ACCETTA

→ rifiuta altrimenti

- $\delta(q, a) = (r, b, L)$

mantenendo l'ordine

ci spostiamo regolarmente a sx

- se incontro l'inizio del nastro

SINGOLO → TM R3/L2

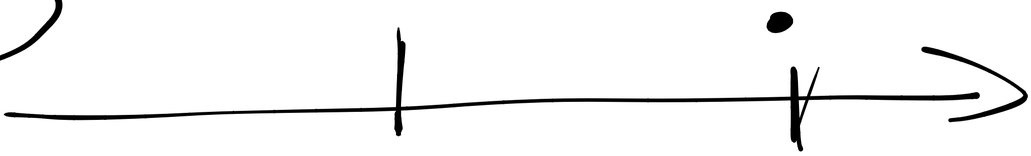
←

(OPPOSTO)

nonostante → siamo equivalenti

(BINARIO) ↑ ↓ ← →

VARIANTI



(TERNARIO) ↑ ↓ ↘ ← →

Esempio di (uso dei simboli)

TM spesso - nastro

→ simile a TM a nastro singolo semi-infinito

(L, R, S)

S = sento in una parte non nulla del nastro

- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$

Created by Notein

- $\delta(q, a) = (r, b, L)$

→ SPOSTAM. A SX

- $\delta(q, a) = (r, b, S)$

← (ZIGZAG) →

(SALTO → SPOSTAM. ITERATIVO)

Ci spostiamo a sinistra

(#) marchiamo dove siamo arrivati

(.) marchiamo la fine dello spostam. a sx

→ Macchina che conta il n. di simboli

→ (Delta = Rami i loro
n. di simboli a cui pongo
a metà)

(COMPLEMENT$_{DFA}$ =
DFA sono l'uno il
complemento l'uno dell'altro)

$$\bar{A} = A$$

→ $\{ (A / B \mid A$ e B DFA
s.t. $L(A) = L(\bar{B}) \}$

T = In input $\langle \quad , w \rangle$

$\langle A, B \rangle$

① $A \cap B = \cancel{\emptyset}$

↗ CHIUSA

② A → esegue tutto A

$\overline{B}$ → esegue tutto B

(simbolo per simbolo che sono diversi)

→ $M \rightarrow \langle A, \overline{B} \rangle$

⇐⇒

- $A, B \in$ COMPLEMENT$_{DFA}$

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

essendo A e B DFA finiti → funziona!

- $A, B \notin$ COMPLEMENT$_{DFA}$

→ no stato di fine

→ no rispetto delle proprietà