TUTORATO 9

- COMPLESSITÀ

  $TIME(f(n))$

- CLASSE $\mathbb{P}$

  ↳ DECIDIBILITÀ

  IN $\mathbb{P}$ ← VERIFICA INT. POLINOMIALE

LIMITI ASINTOTICI

$f: \mathbb{N} \to \mathbb{N}$

↳ CALCOLABILE

↓ FARLA IN TEMPO FINITO

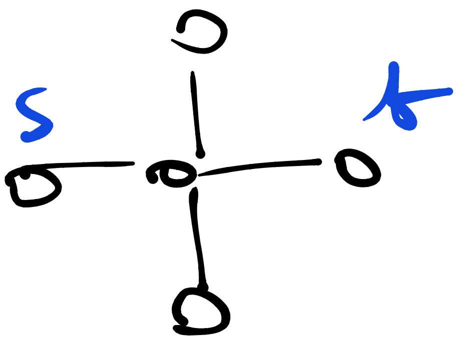
[LB] SOL. [UB]

LIMITATI IN T. FINITO

PROBLEMI IN TEMPO POLINOMIALE

→ T. FINITO

GRAFO →



VERTICI → (K)

PATH = {(G, S, A) | G grafo che contiene un cammino da "S" a "A"}

RAGGIUNGERE "A"
PARTENDO DA "S"

↓

P <=> DECIDIBILITÀ
(#. PASSI FINITO)

- DESCRIZIONE
CON TM "M"

- M = "In input w"

- Copiamo sul nastro tutte
le possibili stringhe rappre.
del grafo G -> O(V+E)

VERTICI/ARCHI

- Scansione un n.
di volte ≤ K -> O(K)
(e quindi tutto il nastro)

- se M accetta → accetta
- se M rifiuta → rifiuta

RELPRIME = { (x, y) |
1 è il MCD di x e y }

TRY: T = su input ⟨x, y⟩
→ ∀ DIV. SUL NASTRO
di ⟨x, y⟩

→ CONTINUARE A
COMPUTARE
TUTTI I FATTORI DI ⟨x, y⟩

→ SE M HA COMPUTATO
TUTTI I FATTORI,
SVOLGI MCD (x, y)

→ SE M ACCETTA,
ACCETTA

→ SE M RIFIUTA,
1 è MCD

[ALTERN.] ↷

- CONTO TUTTI I DIVISORI PRIMI
- "ARRIVO" A $\langle x, y \rangle$
- M ACCETTA / RIFIUTA

- $ALL_{DFA} = \{ \langle A \rangle \mid$
A è un DFA che riconosce
$\Sigma^* \}$ → (ALFABETO DI TUTTE LE STRINGHE)

→ $ALL_{DFA} \in P$

↓

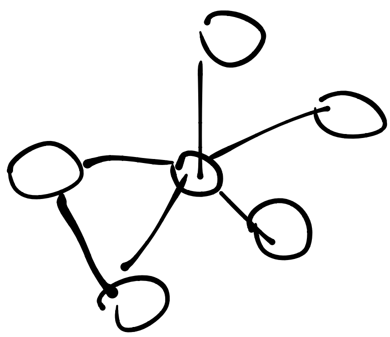
T = Su input <M, w>

- Marco stato iniziale (q_0)
- Ripetiamo finché non ci sono nuovi stati
- Marchiamo tutti gli stati non ancora raggiunti da quello iniziale

Complessità -> $O(\Sigma^*)$

- TRIANGLE

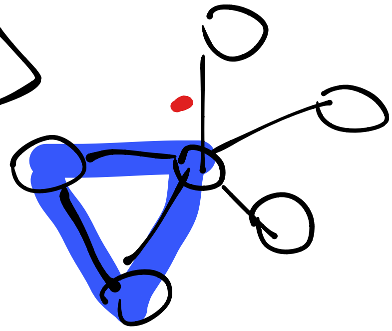
triangolo (sottografo e triangolo in un grafo) -> CLIQUE



TRIANGLE $\in$ P

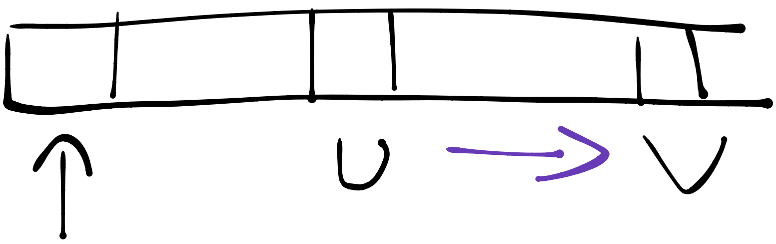
→ TM (M)

T → In input $\langle G \rangle$ (G = grafo)



- Parto da un vertice
- Marchiamo sul nastro i vicini con un simbolo

→ Se riusciamo a fare queste mosse partendo sempre dall'inizio del nastro (= vert. iniziale)



→ Esiste un cammino

da U a V

(stesso procedimento
di sopra)

→ complessità

$= O(|V|^3 \cdot |E|)$

→ (almeno una tripla
di vertici) ⇒ 3 volte

→ B = edges = archi

N.B → DESCRIZIONE
ALGORITMI!

(QUESTO CI SERVE)

ES. 4 SLIDE "RIDUCIBILITÀ"

4) A_{TM} NON È RIDUCIBILE A B_{TM}

$A_{TM} \cancel{\leq_m} B_{TM}$

$A \leq_m B$

→ ASSUMI CHE SIA DECIDIBILE

[$A_{TM} \leq_m B_{TM}$]

T → Su input $\langle M, w \rangle$

Esegui M su w

Se M accetta, accetta,

se M rifiuta, rifiuta

[Se B è decidibile, allora A è decidibile)

Ritorna $\langle M, w \rangle$

T su input $\langle M, w \rangle$

- Esegue $\langle M, w \rangle$
- Verifico che $\setminus \notin T$
- se T accetta, accetta (NO STRINGHE)
- se T rifiuta, rifiuta / va in loop

$$\boxed{\overline{A_{TM}} \leq B_{TM}}$$

ASSURDO! $\rightarrow$

SONO PARTITO DA UN PROBLEMA INDECIDIBILE

---

⑤ A è Turing-ric. e $A \leq_m \bar{A}$, A è decidibile.

$T \rightarrow$ Su input $\langle M, w \rangle$

- Esegui M su w
- Se M accetta, rifiuta
- Se M rifiuta, accetta

$\rightarrow$ Ritorna $\langle M, w \rangle$

$M' \rightarrow$ Su input $\langle w \rangle$

$\rightarrow$ Esegue su input $\langle w \rangle$

$\rightarrow$ Opposto di A

(Se M accetta, accetta / rifiuta)

$$\boxed{\begin{array}{l} \text{Se } w \in \overline{A} \iff w \notin A \\ \text{Se } w \in A \iff w \notin \overline{A} \end{array}}$$

Sapendo che $\bar{A}$ è Turing-ric.
anche A lo è.

- TM CON ALF BINARIO
$\Sigma = \{0,1\}$ $\Gamma = \{0,1 \ldots\}$

→ TERNARIO (VARIANTE)
$\Sigma = \{0,1,2\}$ $\Gamma = \{0,1,2 \ldots\}$

→ BINARIO

→ SCRIVO SOLO NASTRO SOLO

$\emptyset / 1 / \sqcup$ (BLANK)

CONS $TM_{BINARY} \Longleftrightarrow TM_{NORMALE}$

$0 / 1 / \sqcup$

$w = w_1, w_2 \dots w_N$

$\underbrace{(\,(w_1)}_{}\,(\,(w_2)\,(\,(w_3)$

CODIFICA BINARIA

→ sequenza di K simboli per rappresentare il mio alfabeto

$K = 2 \to 2^K = 4$

$K = 3 \to 2^K = 8$

→ Codifica binaria sostituisce per ogni simbolo un n. multiplo di 2 di simboli

∇

- $\delta(r, e) = (q, b, R)$

→ SCRIVO CODIFICA BINARIA SUL NASTRO

E SPOSTA DESTRA DI 2^k CELLE

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

CODIFICA BINARIA

→ TERNARIO

(DUPLICA RAGIONAMENTO

3^k (R → 3^k (CELLE)

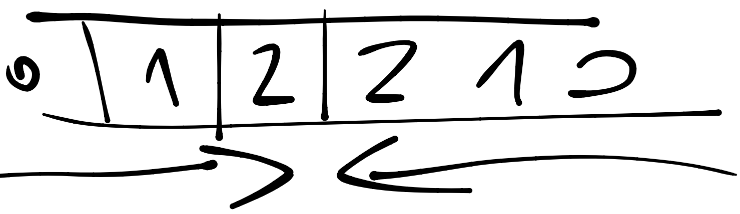
$L = \{\langle M, w \rangle \mid M$ accetta la stringa $ww^R\}$

Dimostra che L è indecidibile

$A_{TM} \le_m L$

$HALT_{TM} \le_m L$

- M' su input $\langle M, w \rangle$
  → simula M su w



→ EQ_{TM}

$M_1 = w$

$M_2 = w^R$

$\cap$ → ACCETTA

RIFIUTA

- se M si ferma sull'input $w w^R$,
- se M non si ferma,

non va in loop

- Ritorna $\langle M, w \rangle$

$\langle M, w \rangle \Longleftrightarrow L$

$\in L$ (ACCETTA ww^R)

$\notin L$ (VA IN LOOP) NO ww^R

VARIAMO (STRINGA FISSA)

$\rightarrow TM_{1010} / TM_{2023}$

TM_{42}

K-PDA

- 1-PDA > 0-PDA
  [PDA] [NFA]

- 2-PDA > 1-PDA
  [2 STACK] [1 STACK]

Dimostrare che 2-PDA siano più potenti di 1-PDA

- 2-PDA > 1-PDA
  [2 STACK] [1 STACK]

2 STACK (DOPPIA MEMORIA)

$2\text{-PDA} > 1\text{-PDA}$

[2 STACK] [1 STACK]

$\updownarrow$ $\updownarrow$

NASTRI 1 NASTRO

→ 1° NASTRO → INPUT DI 1-PDA

2° NASTRO → ESEGUE IL SUO E ACCETTA ALMENO 1-PDA

– EVENTUALMENTE DARE UN ESEMPIO DI UN LINGUAGGIO CHE NON È ACCETTATO DA 1-PDA

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

LING. NOTO NON REGOLARE PER CF (CCF $\longleftrightarrow$ PDA)

$$L = \{0^n 1^n \mid n \geq 0\}$$

LING. NOTO NON REGOLARE PER DFA