PASCAL MINIMO NON DETERMINISTICO

INPUT:

$X = x_1, \dots, x_n$ MEMORIZZATA IN $N(N[1] = x_1, \dots, N[n] = x_n)$, $x \in \Sigma^*$

COSTANTI:

$P = \{P_1, \dots, P_k\}$ CON $P_i = \langle q_{i1}, S_{i1}, S_{i2}, q_{i2}, m_i \rangle, q_0, q_A, q_R$

$q \leftarrow q_0$; $C \leftarrow 1$; $PC \leftarrow 1$; $UC \leftarrow n$

WHILE $(q \neq q_A \vee q \neq q_R)$ DO BEGIN:

$\Psi = \{P_i \in P : P_i = \langle q, N[C], \_, \_, \_ \rangle\}$ → Ψ È L'INSIEME DELLE MULTIQUINTUPLE DI P_i

→ $|\Psi|$ È COSTANTE IN QUANTO È AL MASSIMO UGUALE AL GRADO DI NON DETERMINISMO

IF $(\Psi \neq \emptyset)$ THEN BEGIN:

SCEGLI $P_i = \langle q_i, N[C], S_{i2}, q_{i2}, m_i \rangle \in \Psi_i$

$N[C] \leftarrow S_{i2}$, $q \leftarrow q_{i2}$, $C \leftarrow C + m$

IF $(C < PC)$ THEN ....

IF $(C > UC)$ THEN.....

END IF

END WHILE

RETURN q

OSS

1) L'ISTRUZIONE SCEGLI È L'ISTRUZIONE CHE IMPLEMENTA IL NON DETERMINISMO

2) L'ALGORITMO È BASATO SUL MODELLO DEL GENIO CHE SCEGLIE LA QUINTUPLA DA ESEGUIRE

3) POSSIAMO FIDARCI DEL GENIO SOLO SE AL TERMINE DELL'ALGORITMO CI PORTA IN q_A

4) USANDO LA DEFINIZIONE DI NP COME MACCHINA ACCETTANTE A NOI NON IMPORTA DI QUALSIASI RISULTATO CHE NON SIA q_A

5) SE $\exists$ UNA COMPUTAZIONE CHE PORTA IN q_A ALLORA $\exists$ UN GENIO CHE NE LA SELEZIONA

PROBLEMA 3-SAT

$$I_{3\text{-}SAT} = \{ \langle X, f \rangle : f = C_1 \wedge C_2 \wedge \dots \wedge C_m ;\ \forall i \in [0,m] \left[C_i = l_{i1} \vee l_{i2} \vee l_{i3},\ \forall_{ij}\ l_{ij} \in X \vee \neg l_{ij} \in X \right] \}$$

$$S_{3\text{-}SAT}(X,f) = \{ a : X \to \{\text{VERO}, \text{FALSO}\} \}$$

$$\pi_{3\text{-}SAT}(X, f, S(X,f)) = \exists\, a \in S_{3\text{-}SAT}(X,f) : f(a(X)) = \text{VERO}$$

ALGORITMO 3-SAT

```
INPUT: X ; f = C1 ∧ C2 ∧ ... ∧ Cm
q <- q0; c <- 1; pc <- 1; uc <- n

FOR (i <- 1; i <= |X|; i <- i+1) DO
    SCEGLI a(xi) IN {VERO, FALSO}
FOR (i <- 1; i <= |X|; i <- i+1) DO
    SOSTITUISCI xi CON a(xi) IN f
IF (f = VERO) THEN
    q <- qA
RETURN q
```

FASE DI SCELTA → ESEGUITA IN $O(|X|)$

PUO' ESSERE FATTA SOLO FRA UN NUMERO COSTANTE DI POSSIBILITÀ

FASE DI VERIFICA → ESEGUITA IN $O(|X||f|)$

SI CONTROLLA SE L'ISTANZA CREATA NELLA FASE DI SCELTA PORTI IN q_A

$O(|f|)$

CLASSE DI COMPLESSITÀ

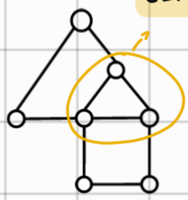
L'ALGORITMO ACCETTA 3-SAT IN TEMPO NON DETERMINISTICO $O(|X||f|)$

3-SAT $\in$ NP

Problema del Sottoinsieme Clique

Per ogni coppia di nodi esiste un'arco

Sottoinsieme di nodi connessi di cardinalità almeno k



Clique di $k=3$

$$\mathcal{I}_{CL} = \{<G(V,E),k> : G \text{ grafo}, k \in \mathbb{N}\}$$

$$S_{CL}(G,k) = \{V' \subseteq V\}$$

$$\pi_{CL}(G,k,S(G,k)) = \exists V' \in S(G,k) : \forall v_i, v_j \in V' \left[(v_i,v_j) \in E\right] \wedge |V'| \geq k$$

Algoritmo clique

```
INPUT: G=(V,E), k ∈ N

V' ← ∅
FOR EACH (v ∈ V):
    SCEGLI SE v ∈ V'
CLIQUE ← TRUE
IF (|V'| < k) THEN
    CLIQUE ← FALSE
FOR EACH (v ∈ V')
    FOR EACH (u ∈ V')
        IF ((u ≠ v) ∧ (u,v) ∉ E) THEN:
            CLIQUE ← FALSE
RETURN CLIQUE
```

Fase di scelta $\rightarrow$ Eseguita in $O(|V|)$

Eseguita a livello di S_{CL}

$O(|V|)$

$O(|V|^2)$

$O(|E|)$

Fase di verifica $\rightarrow$ Eseguita in $O(|V|^2|E|)$

Eseguita a livello di π_{CL}

Classe di complessità

L'algoritmo accetta il problema clique in tempo non deterministico $O(|V|^2|E|)$

Il problema clique $\in NP$

PROBLEMA DEL CICLO HEMILTONIANO

UN CICLO HEMILTONIANO È UN CICLO CHE PASSA PER OGNI NODO DEL GRAFO UNA ED UNA VOLTA SOLA

$$\mathcal{I}_{HC} = \{ <G=(V,E)> : |V| = n \}$$

$$S_{HC}(G) = \{ P : <v_1, v_2, \dots, v_n> : \forall i [v_i \in V] \}$$ → CAMMINO COMPRENDENTE TUTTI I NODI DEL GRAFO

$$\pi_{HC}(G, S(G)) = \exists P \in S_{HC}(G) : \forall i \neq j \, [v_i \neq v_j] \wedge \forall i = 1 \dots n \, [(v_i, v_{i+1}) \in E \wedge (v_n, v_1) \in E]$$

$$\hookrightarrow \exists P \in S(G) : \eta(G, S(G))$$

ALGORITMO CAMMINO HEMILTONIANO

```
INPUT: G=(V,E)

FOR (i <- 1; i <= n; i <- i+1) DO
    FOR (v ∈ V) DO
        SCEGLI SE v = v_i O v ≠ v_i
```

SERVE A SCORRERE LE POSIZIONI DI P

FASE DI SCELTA → $O(n^2)$

```
CICLE <- TRUE
FOR (i <- 1; i <= n; i <- i+1) DO
    FOR (j <- 1; i <= n; i <- i+1) DO
        IF (v_i ≠ v_j) THEN CYCLE <- FALSE
FOR (i <- 1; i <= n; i <- i+1) DO
    IF ((v_i, v_i+1) ∉ E) THEN CYCLE <- FALSE
IF ((v_n, v_1) ∈ E ∧ CYCLE = TRUE) THEN
    RETURN CYCLE
```

FASE DI VERIFICA → $O(n^2)$

CLASSE DI COMPLESSITÀ

L'ALGORITMO ACCETTA IL PROBLEMA DEL CAMMINO HEMILTONIANO IN TEMPO ND $O(n^2)$

IL PROBLEMA DEL CAMMINO HEMILTONIANO $\in NP$

DEFINIZIONE ALTERNATIVA DI NP → PROVVISORIA

DATO Π : $\pi_\Pi(\underbrace{X}_{\text{ISTANZA SI}}, S_\Pi(X)) = \exists\, Y \in S_\Pi(X) : \eta_\Pi(\underbrace{Y}_{\text{Y SOSTITUISCE } S(x)}, X) = \text{VERO}$

$\eta(x,y)$ È VERIFICABILE IN TEMPO DETERMINISTICO POLINOMIALE $\wedge$ $\underbrace{Y}_{\text{SOLUZIONE POSSIBILE}}$ È COSTRUITO IN TEMPO NON DETERMINISTICO POLINOMIALE $\Rightarrow [\Pi \in NP]$

ALGORITMO NON DET. STAMPO

INPUT: $x \in \mathcal{J}_\Pi$

COSTRUISCI NON DETERMINISTICAMENTE $Y \in S_\Pi(x)$

VERIFICO DETERMINISTICAMENTE $\eta(Y)$