1. Soddisficibilità (Pag 62) ✓
2. ϕ∈2–SAT (Pag 64) ✓
3. Definizioni di NP Equivalenti(Pag 67) ✓
4. 3COL ∈ NP (Pag 138 Exyss) ✓
5. 3SAT < CLIQUE (Pag 140 Exyss)
6. Teorema Cook-Lewin (SAT è NP COMPLETO) (Pag 70) ✓
7. SAT < 3SAT (Pag 148 Exyss)
8. unSAT coNP-Completo (Pag 75)

2) Voglio dimostrare $\phi \in$ 2-SAT $\iff$ nessuna variabile nella stessa comp fort connessa con la sua negata

questo mi permetterà di dire 2-SAT $\in P$

$\Rightarrow$) Sia ϕ una CNF con $\phi = \{x_1 \dots x_m\}$

Creo un grafo G che ha come nodi le variabili

e $(x_1, x_{x_2}) \in E(G)$ se c'è la clausola in ϕ

Voglio dim che $x \in \bar{x}$ non nella stessa $\underbrace{\text{comp fort connessa}}_{\text{CFC}}$

Questo è facile poiché se ho

$$E(G) \ni (a,b) \Rightarrow \bar{a} \vee b$$

ma ho anche $(b,a) \Rightarrow \bar{b} \vee a$

Prendo ora $x_1 \to x_2 \to \dots \to \bar{x}_1 \to x_m$

I vertici di una CFC $\Rightarrow (x_1, \bar{x}_1)$ e $(\bar{x}_1, x_1) \in E(G)$

$\Rightarrow \underbrace{\bar{x}_1 \vee \bar{x}_1}_{= \bar{x}_1} \wedge$ e $\underbrace{x_1 \vee x_1}_{\overset{?}{=} x}$ $\Rightarrow$ la formula non è SAT (nessun valore da 1) △

$\Leftarrow$) ORA SO CHE x_1 e $\bar{x}_1 \notin$ CFC

ANDIAMO A CONDENSARE OGNI CFC C IN UN SINGOLO VERTICE C
E FACCIO UN ORD. TOPOLOGICO
PER OGNI x HO 2 POSSIBILITÀ

- $x \in C_i$ e $\bar{x} \in C_j$ con $i < j$
  in questo caso $x = 0$ $(\bar{x} = 1)$
- $x \in C_i$ e $\bar{x} \in C_j$ con $i > j$
  in questo caso $x = 1$ $(\bar{x} = 0)$

COMBINANDO QUESTO CLAIM SIA UN $(A,B) \in E(G)$ NON
POSSO MAI AVERE $A=1$ E $B=0$
PER ASSURDO DICO INVECE $a=1$ e $B=0$

$\bar{a}=0$ $\bar{B}=1$

$\Rightarrow$ $a \in C_i$ e $B \in C_j$ con $j < i$

$$C_i \longrightarrow C_j$$

MA HO $\bar{B} \in C_k$ con $k > i > j$

$$\overset{\bar{a}}{C_i} \longrightarrow \overset{a}{C_j} \longrightarrow \overset{\bar{b}}{C_k}$$

È FACILE NOTARE L'ASSURDO POICHÉ SE HO $(a,B) \in E(G)$
ALLORA ANCHE $(\bar{B},\bar{a}) \in E(G)$ quindi $\bar{B} \to \bar{a}$ CHE VIOLA L'ORD. TOPOL.

$\Rightarrow$ x e $\bar{x}$ NON NELLA STESSA CFC ALLORA L'ASSEGNAMENTO
SODDISFA ϕ

$\square$ $\Rightarrow$ 2-SAT $\in P$

3) ENTRAMBE LE DEF DI NP SONO EQUIVALENTI

$\Rightarrow$) $L \in DEC$ SIA N LA NTM CHE DECIDE L IN TEMPO POLINOMIALE

CREO UNA TM V DEFINITA COME SEGUE:

V: INPUT $\langle w, c \rangle$

SIMULA $N(w)$ CON SCELTE c

SE ACCETTA ACCETTA

È CHIARO CHE $w \in L(N) \Leftrightarrow \exists$ UN RAMO DI N CHE ACCETTA w
$\Leftrightarrow \exists k \in \Sigma^*$ t.c. $\langle w, k \rangle \in L(V)$

$\Rightarrow$ L HA UN VERIFICATORE

□

$\Leftarrow$) L HA UN VERIFICATORE IN TEMPO POLINOMIALE V

CREO UNA NTM N CHE OPERA COME SEGUE:

- INPUT $\langle w \rangle$
- SCEGLIE NON DETERMINISTICAMENTE UNA $c \in \Sigma^*$ di lunghezza n^k
- SIMULA V SU $\langle w, c \rangle$ SE V ACCETTA ACCETTA

È CHIARO CHE $w \in L(N) \Leftrightarrow \exists c \in \Sigma^*$ t.c. UN RAMO DI N CON SCELTE c ACCETTA $w \Leftrightarrow w \in L(V)$

È OVVIO L'ESECUZIONE SIA POLINOMIALE PERCHÉ SIMULA V che È POLINOMIALE.

□

(4) 3col $\in$ NP

$3col = \{ \langle G \rangle \mid G$ grafo 3 colorabile $\}$

Sia V una NTM che prende in input $\langle \langle G \rangle, C \rangle$

- interpreta $C = \{c_1 \dots c_m\}$ con $m = |V(G)|$ e $\forall i \in [1, \dots m]$
  $c_i = \{R, G, B\}$

- ripete questo step per ogni $(c_i, c_j) \in E(G)$:

  se $c_i = c_j$ rifiuta

  accetta

è ovvio che

$\langle G \rangle \in 3col \iff \exists$ colorazione valida per ogni nodo $\iff \exists C = \{c_1 \dots c_m\} \mid \langle \langle G \rangle, C \rangle \in V$

Per costruzione dell'algoritmo $3col \in NP$

5 $3SAT \leq_m^p CLIQUE$

Creo un grado G,

Conto inoltre k = numero di clausole or in phi

partendo dalla phi per ogni variabile (xi) nella formula creo un vertice in G (se ho più volte la stessa variabile creo più volte lo stesso vertice), collego ogni vertice agli altri tranne che i vertici uguali e i suoi negati, organizzo poi i nodi in k-triple.

Prendo poi C={x1,…xi..} in cui inserisco un letterale valido per ogni clausola di phi

Per costruzione del grafo G una qualsiasi k-CLIQUE sarà valida => 3SAT < CLIQUE

quindi phi appartiene a 3SAT => f(phi)= <G,k> appartiene a CLIQUE

6) COOK-LEVIN (SAT NP-COMPLETO)

$NP\text{-}HARD = \{ L \in \Sigma^* \mid \forall A \in NP \; A \leq_m^P L \}$

$CoNP = \{ L \in D_{\Sigma} \mid \bar{L} \in NP \}$

L = NP COMPLETO SE $L \in NP$ E $L \in NP\text{-}HARD$

1° SAT $\in$ NP

SIA V UNA TM DEFINITA COME SEGUE:

- PRENDE IN INPUT $\langle\langle \phi \rangle, c\rangle$
- INTERPRETA C COME UN ASSEGNAZ. DI VARIABILI $c = \{c_1, \ldots, c_m\}$
- LANCIA ϕ SU C
- SE ϕ ACCETTA V ACCETTA

è CHIARO CHE

$$\phi \in SAT \iff \exists c \in \Sigma^* \text{ t.c. } \langle\langle \phi \rangle, c\rangle \in V$$

$\Rightarrow$ V è UN VERIFICATORE PER SAT $\triangle$

2°) NON SCRITTA

$$\phi_{START} = x_{1,1,\#} \wedge x_{2,1,w_1} \wedge \ldots \wedge x_{n^k-2,1,\sqcup} \wedge x_{n^k-1,1,\#}$$

$$\phi_{ACC} = \bigvee_{1 \leq i,j \leq n^k} x_{i,j,q_{accept}}$$

$$\phi_{LOC} = \bigvee_{1 \leq i,j} x_{i,j} \times$$

$$\bigwedge_{\substack{s,t\in S\\ s\neq t}} (\overline{x_{ijs}}) \vee (\overline{x_{ijt}})$$

$$\phi_{Loc} = \bigwedge_{1\le i\le j\le m^k} \left(\bigvee_{s\in S} x_{i,j,s}\right) \wedge \left(\bigwedge_{\substack{s,t\in S\\ s\neq t}} (\overline{x_{i,j,s}}) \vee (\overline{x_{i,j,t}})\right)$$

$$\phi_{Move} = \bigwedge_{1\le i\le j\le n^k} \{i,j\} \text{ legal}$$

$\forall A$ $A \in NP$ $\leq_m^p$ SAT $\square$

7 $\quad SAT \leq_m^p 3SAT$

6)

UNSAT $\in$ coNP-Compl.

UNSAT = $\overline{SAT}$

Poiché SAT $\in$ NP $\iff$ $\overline{SAT}$ $\in$ coNP

ora voglio dim che $\forall A \in$ coNP $A \leq_m^p \overline{SAT}$

Per Cook Levin so che SAT $\in$ NP-completo

Poiché $L \in NP \leq_m^p SAT \Rightarrow \overline{L} \leq_m^p \overline{SAT}$

$\overline{L} \in$ coNP poiché $L \in$ NP

$\Rightarrow \overline{SAT}$ = UNSAT $\in$ coNP-completo

9. PATH Decidibile in log^2 n F ✗
10. DSPACE(f(n)) ⊆DTIME(2O(f(n))) - F ✓
11. Teorema di Savitch (Pag 80) - F ✓
12. P ⊆NP ⊆PSPACE ⊆EXP ⊆NEXP ⊆EXPSPACE (Pag 162) ✓
    1. PSPACE= NPSPACE ✓
13. L e NL (Pag 165 Exyss) - F ✓
    1. NL Seconda def (Pag 166 Exyss) - F ✓
14. Path è NLog Completo (Pag 80-82 di marco) - F ✓
15. TQBF è PSPACE-difficile (Pag 84 marco)
16. L=coL (Pag 167) - F ✓
17. Teorema di Immerman-Szelepcsényi

10 $\quad DSPACE(f(n)) \subseteq DTIME(2^{O(f(n))})$

Sia M una TM t.c. $DSPACE(f(n)) \Rightarrow L(M)$

M usa una conf del tipo work q_i TAPE : J

con work tape contenuto del nastro di lavoro

q_i stato attuale

J posizione della testina

Sia $t(n)$ il n° massimo di tempo di conf di M

poiché M è un decisore e le conf non si ripetono $t(n)$ sarà anche il numero max di configurazioni

$$t(n) = |\Gamma|^{f(n)} \cdot |Q| \cdot n$$

$$f(n) \geq \log n \Rightarrow 2^{f(n)} \geq n$$

si ha che

$$t(n) = |\Gamma|^{f(n)} \cdot |Q| \cdot n \leq |\Gamma|^{f(n)} \cdot |Q| \cdot 2^{f(n)} = 2^{O(f(n))}$$

□

(11) SAVITCH

$$NSPACE(f(n)) \subseteq DSPACE(f^2(n))$$

Sia N una NTM t.c. $L(N) \in NSPACE(f(n))$

e N ha un solo stato accettante e si trova nella configurazione $w_c q_c$ TAPE: J per il teorema di relazione spazio tempo ($DSPACE(f(n)) \subseteq DTIME(2^{O(f(n))})$) sappiamo che il n° max di configurazioni di N è $2^{O(f(n))}$

Creo così un grafo $G_{N,w}$ che ha come vertici le conf di N e ha archi tra i vertici secondo la δ.

Creiamo una TM M che opera come segue

- prende in input w
- simula l'algoritmo PATH? su $G_{N,w}$ con PATH?($G_{N,w}, q_{start}, q_{acc}, \log m$)
- accetta se PATH? accetta

ricordo che sappiamo che PATH? ha costo $\log^2 m$

è chiaro che $w \in L(M) \iff$ PATH?($G_{N,w}, q_{start}, q_{acc}, \log m$) accetta
$\iff$ esiste un cammino da q_{start} a q_{acc} nel grafo di N
$\iff w \in L(N)$

$\Rightarrow L(N) = L(M)$

vediamo ora il costo

$$s(n) = \log^2 m = \log^2(2^{O(f(n))}) = O(f^2(n))$$

$\Rightarrow NSPACE(f(n)) \subseteq DSPACE(f^2(n))$

(12.1) PSPACE = co PSPACE

co PSPACE = $\{L \in DEC \mid \bar{L} \in PSPACE\}$

$\Rightarrow$)

Sia $A \in$ PSPACE e sia D il decisore di A

sia $\bar{D}$ una TM che opera come segue

- prende in input w
- simula D su w
- se D accetta rifiuta o viceversa

è chiaro che

$$w \in L(\bar{D}) \iff w \notin L(D) = A$$

$$L(\bar{D}) = \bar{A}$$

$\Leftarrow$) uguale alla prima

2) PSPACE = NPSPACE

$\Rightarrow$ PSPACE $\subset$ NPSPACE è banale

$\Leftarrow$ NPSPACE $\subset$ PSPACE

uso T. di Savitch

$$L \in NPSPACE \Rightarrow \exists k \in \mathbb{N} \mid L \in NSPACE(n^k) \subseteq SPACE(n^{2k})$$

polinomiale

$$\Rightarrow L \in PSPACE$$

B) NL

$$L = \bigcup \text{DSPACE}(\log n)$$

$$NL = \bigcup \text{NSPACE}(\log n)$$

NL ha anche un altra DEF

$$NL = \{ L \in DEC \mid L \text{ ha verificatore read-once in } \log n \}$$

vogliamo dimostrare l' uguaglianza tra le DEF

$\Rightarrow$ sia A un $L \in DEC$ e sia N la NTM che lo decide in tempo logaritmico

N durante la sua computazione avrà una serie di rami di computazione

sia V un TM che opera come segue

- INPUT $\langle w, c \rangle$
- simula w su N con scelte c
- se $N(w)$ accetta V accetta

è chiaro che

$w \in L(N) \iff$ esiste un ramo di comp che acc
$\iff \exists c \in \Sigma^*$ t.c. $\langle w, c \rangle \in L(V)$

il costo computazionale è uguale (senza solo N che opera in log.)

Δ

$\Leftarrow$) SIA A un LeNP | HO UN VERIFICATORE V CHE OPERA IN t_{p}

SIA ORA N UN TM CHE OPERA COME SEGUE

- INPUT $\langle w \rangle$
- SCEGLIE NON DETERMINISTICAMENTE UNA STRINGA C
- SIMULA $\langle w, c \rangle$ SU V
- SE V ACCETTA ALLORA N ACCETTA

È CHIARO CHE

$$w \in A \Leftrightarrow \exists c \in \Sigma^* \mid \langle w, c \rangle \in L(V)$$
$$\Leftrightarrow \exists \text{ un ramo di } N \text{ che sceglie } c$$
$$\Leftrightarrow w \in L(N)$$

ANCHE QUI IL COSTO CAP È OVVIO

$\square$

1(c) **PATH è NL-COMPLETO**

$$NL\text{-}Complete = \{ L \in Dec \mid L \text{ è NL completo} \}$$

L è NL-COMPLETO se $L \in NL$ e $\forall A \in NL$ $A \leq^L_m L$

1° **PATH $\in$ NL**

SIA N UNA NTM DEFINITA COME SEGUE

- PRENDO IN INPUT $\langle G, s, t \rangle$
- SE $s = t$ ACCETTA
- CREA UNA VARIABILE $curNode = s$
- $m = |V|$
- PER i DA 1 A m:
  - SCEGLI NON DET UN NODO $v \in V(G)$
  - SE $(curNode, v) \notin E(G)$ RIFIUTA
  - SE $v = t$ ACCETTA
  - ALTRIMENTI $curNode = v$
- RIFIUTA

È CHIARO CHE

$$\langle G, s, t \rangle \in L(N) \iff \exists \text{ CAMMINO } s \to t \text{ in } G \iff \langle G, s, t \rangle \in PATH$$

IL COSTO CHE È LOGARITMICO PER COSTRUZIONE

2°) SIA $A \in NP \rightarrow A \leq_m^L PATH$ DECISO DA N'

$f : \Sigma^* \rightarrow \Sigma^*$

SIA F LA TM PER CALCOLARE f

- Prendo in input w
- COSTRUISCO IL GRAFO DI CONF $G_{N',w}$
- RESTITUISCO $\langle G_{N,w}, c_{START}, c_{ACCEPT} \rangle$

NOTO CHE $w \in L(N') \iff \exists$ un RAMO w N' CHE ACCETTA

$\iff \exists\ c_{START} \rightarrow c_{ACCEPT}$ in $G_{N,w}$

$\iff \langle G_{N,w}, c_{START}, c_{ACCEPT} \rangle \in PATH$

$\square$

15) $3COL \le_m^P SAT$

Perché so che SAT è NP completo per Cook Levin mi basta trovare un algoritmo per dimostrare $3col \in NP$:

Sia V una TM definita come segue

- Input $\langle\langle G\rangle, c\rangle$ con G grafo
- interpreta $c = \{c_1 \dots c_m\}$ con $m = |V(G)|$ e $c_i = \{R, G, B\}$
- $\forall (v_i, v_j) \in E(G)$:
  se $c_i = c_j$ rifiuta

accetta

È chiaro che

$\langle G\rangle \in 3col \iff \exists$ colorazione $c_1 \dots c_m$ valida
$\iff \exists\, c = \{c_1 \dots c_m\} \in \Sigma^*$ t.c. $\langle\langle G\rangle, c\rangle \in V$

Il costo computazionale è chiaramente polinomiale $\Rightarrow 3col \in NP$

$\Rightarrow 3col \le_m^P SAT$ $\square$