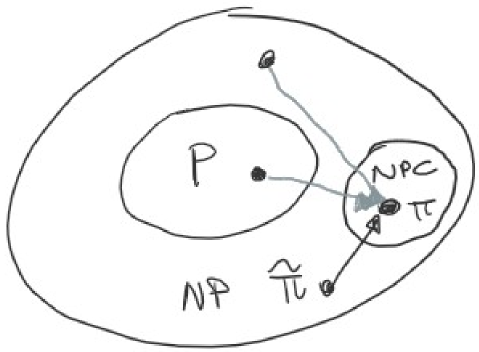
Teorema Cook-Levin

NP Completo

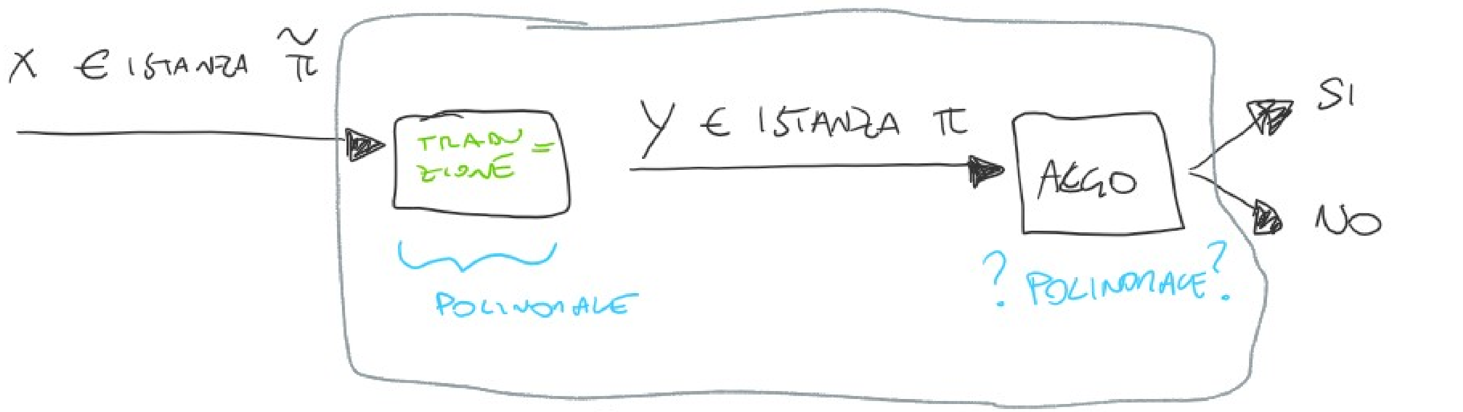
INTRODUZIONE



π è NP completo se

① π APPARTIENE AD NP (DECISIONE)

② $\forall\, \hat{\pi} \in NP$ ESISTE UNA RIDUZIONE POLINOMIALE da $\tilde{\pi} \xrightarrow[P]{} \pi$



? ALGORITMO POLINOMIALE PER $\hat{\pi}$

$|x| = n$

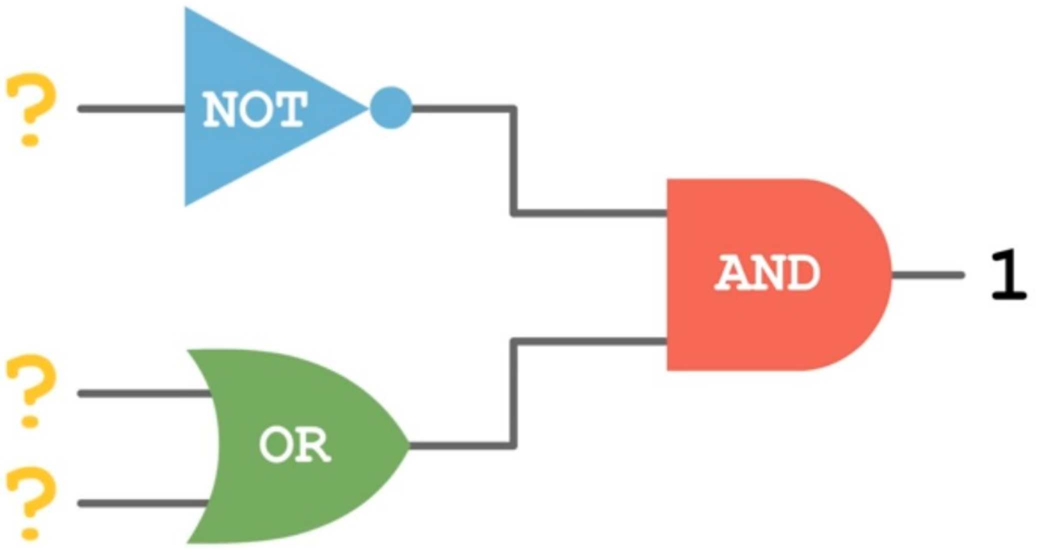
$|y| = O(n^d)$

Cook-Levin Theorem

Il teorema di Cook-Levin afferma che il problema **SAT** è **NP-completo** attraverso la tecnica che mostra che qualsiasi problema in NP può essere **ridotto** ad esso

CIRCUIT-SAT

Mattoncini lego



In un problema SAT:

“dato un circuito” → esiste un input che me lo risolve (ottengo 1)?

SAT

TEOREMA SAT È NP-COMPLETO

Se $L \in NP$ e sia M una macchina di Turing che decide L in tempo minore di $p(n)$ [dove p è polinomiale]

UN PROBLEMA NP-COMPLETO SARÀ ASSOCIATO ALLA MACCHINA DI TURING CHE LO DESCRIVE.

IDEA

POSSIAMO RAPPRESENTARE LA SEQUENZA DELLA CONFIGURAZIONE DELLA MACCHINA M COME UN TABLEAU.

OGNI CONFIGURAZIONE SARÀ UNA RIGA.

			NASTRO					
	STATO	POSIZIONE TESTINA	Sqr 1	...	Sqr n	...	Sqr p(n)	
STEP 1	q_0	0	w_1		w_n	$\sqcup$	$\sqcup$	CONFIGURAZIONE INIZIALE
STEP 2								
STEP P(N)	q ACCETTA							

ESISTENZA DI UNA SEQUENZA CHE PORTA AD UNO STATO ACCETTANTE $\equiv$ SODDISFARE TUTTI I VINCOLI CHE SONO ESPRESSI NELLA TABELLA.

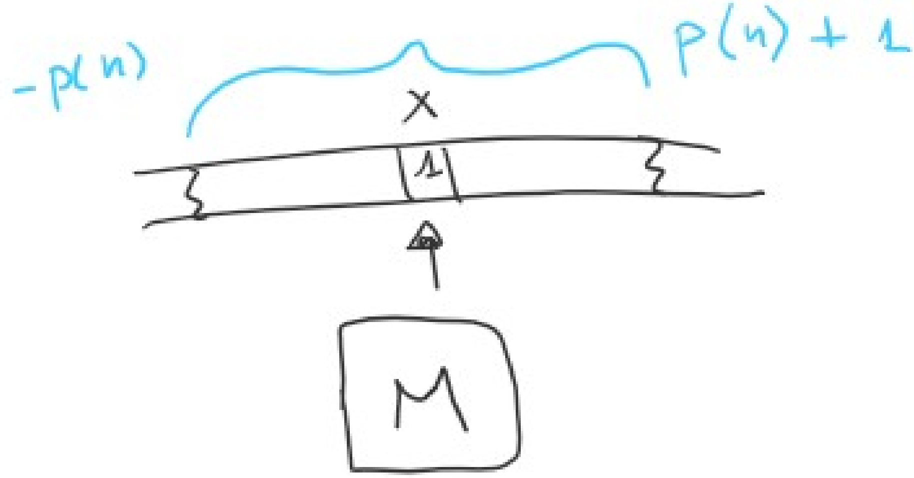
DIM

M = MACCHINA DI TURING NON DETERMINISTICA CHE MI DESCRIVE UN PROBLEMA POLINOMIALE

$M \to T_M(n) \le p(n)$ TEMPO

$M \to S_M(n) \le p(n)$ SPAZIO

UNA M.d.T. CHE ACCETTA X COME INPUT



$\boxed{\text{VARIABILI}}$

$i \Rightarrow$ TEMPO

$Q[i,k] \Rightarrow$ al tempo i Q è nello stato k.

$H[i,j] \Rightarrow$ al tempo i la testina è in j

$S[i,j,k] \Rightarrow$ al tempo i, alla posizione j contiene un simbolo S_k.

TEMPO

$$0 \leq i \leq p(n)$$

STATI

$$Q = \{q_1, q_2, \dots, q_v\} \qquad v = \#\ n^\circ \text{ stati}$$

$$0 \leq k \leq v \qquad [\text{polinomio}]$$

TESTINA

$$-p(n) \leq 1 \leq p(n)+1 \qquad [\text{polinomiale}] \cong p(n)^2$$

NASTRO

ALFABETO

$$0 \leq k \leq \#\Gamma \qquad [\text{polinomiale}]$$

$\boxed{\text{CLAUSOLE}}$ (es $x_i \vee \bar{x}_j \vee \dots$)

SERVONO PER COGLIERE LE RESTRIZIONI SULLA COMPUTAZIONE.

$G_1 \to$ LA MACCHINA È SOLAMENTE IN UNO STATO

$G_2 \to \forall i$ LA TESTINA È IN UNA SOLA POSIZIONE

$G_3 \to \forall i$ OGNI CELLA CONTIENE SOLO UN SIMBOLO

$G_4 \to$ IN $i=0$ LA MACCHINA È NELLA CONFIGURAZIONE INIZIALE.

$$\begin{cases} Q = q_0 \\ \text{sul nastro} = x \end{cases}$$

TUTTE LE CLAUSOLE DESCRIVONO UNA MACCHINA DI TURING

$G_5 \to$ MACCHINA ACCETTA ENTRO $p(n)$ PASSI

$G_6 \to \forall i$ LA CONFIGURAZIONE DI M AL TEMPO $(i+1)$ È OTTENUTA APPLICANDO LA δ

$\boxed{G_1}$ ($\forall i$ M è in un solo stato)

LA MdT È ALMENO IN UNO STATO

- $Q[i,0], Q[i,1], \ldots\ldots Q[i,r] \qquad 0 \leq i \leq p(n)$
- $\left(\overline{Q[i,j]} \wedge \overline{Q[i,j']}\right) \qquad \forall i, j \text{ con } i \neq j$

NON PUÒ ESSERE IN 2 STATI
NELLO STESSO ISTANTE

$\boxed{G_2}$ ($\forall i$ la testina è in una sola cella)

- $H(i,-p(n)) \vee H(i,-p(n)-1) \vee \dots \vee H(i,p(n)+1) \quad \forall i$
- $(\overline{H[i,j] \wedge H[i,j']}) \quad \forall i,j$ con $i \neq j$

$\boxed{G3}$ (ogni cella contiene solamente un simbolo)

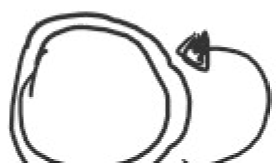
- $S_{ij0} \vee S_{ij1} \vee S_{ij2} \vee \ldots \quad \forall i, j$
- $(\overline{S_{ijk}} \vee \overline{S_{ijk'}}) \quad \forall i, \forall j, \forall k \neq k'$

$\boxed{G4}$ (configurazione iniziale)

$$\begin{cases} Q[0,0] \wedge H(0,1) \\ \underbrace{S[0,1,k_1] \wedge S[0,2,k_2] \wedge \ldots \wedge S[0,x_n,k_n]}_{\text{simboli sul nastro contengono } x = k_1 k_2 k_3} \\ S[0,p(n)+1,\sqcup] \wedge \ldots \quad \leftarrow \text{blank sul nastro} \end{cases}$$

$\boxed{G5}$ (la macchina accetta entro $p(n)$ passi)

$(Q_{p(n)}\, Y)$ con Q_Y è uno stato accettante



$\boxed{G6}$ (la funzione δ è corretta)

Intuizione:

1) se la testina non è su una cella J $\rightarrow$ quella cella non si modifica

$$\overline{S(i,J,\ell)} \wedge H(i,J) \wedge S(i+1,J,\ell)$$

2) devo riuscire a codificare la funzione delta

DX o SX

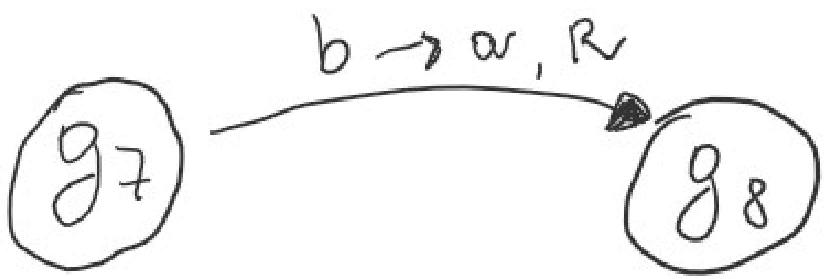
$$\left[\overline{H(i,J)} \vee \overline{Q(i,k)} \vee \overline{S(i,J,\ell)}\right], H(i+1, J+\bar{D}) \wedge$$

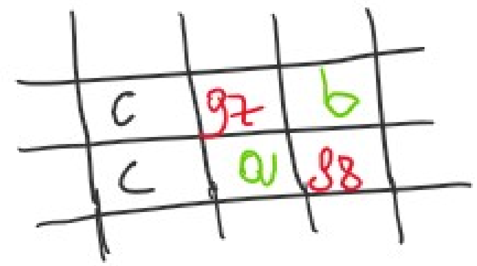
$$\left[\overline{H(i,J)} \vee \overline{Q(i,k)} \vee \overline{S(i,J,\ell)}\right], Q(i+1, k') \wedge$$

$$\left[\overline{H(i,J)} \vee \overline{Q(i,k)} \vee \overline{S(i,J,\ell)}\right], S(i+1, J', \ell') \wedge$$

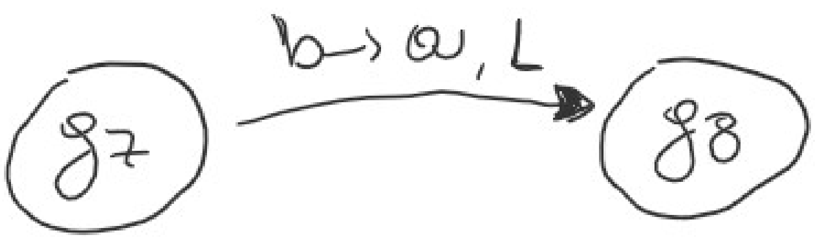
INTUIZIONE

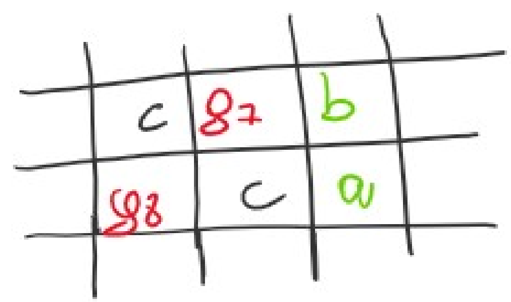
UNA TRANSIZIONE (di una MDT NON DETERMINISTICA)



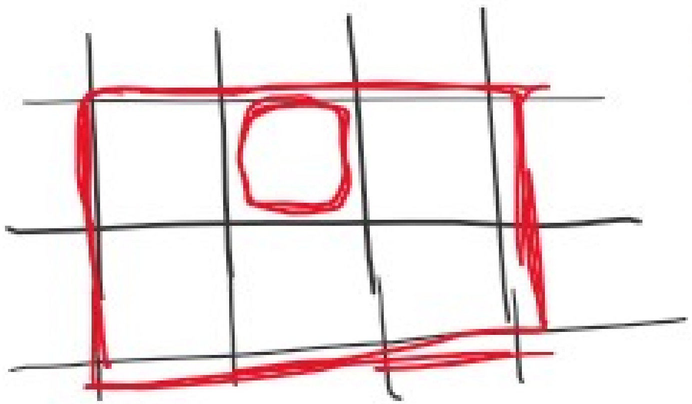


c = qualsiasi simbolo





1) LEGGIAMO UN SIMBOLO
2) SCRIVIAMO UN SIMBOLO
3) CI SPOSTIAMO



LA FINESTRA È CENTRATA NELLA CELLA i j

LA FUNZIONE DI TRANSIZIONE CI DICE QUALI SONO LE "FINESTRE" CORRETTE.

CON UN ALFABETO $\Gamma = \{a, b, c, d, \sqcup\}$

$$q_7 \xrightarrow{b \to a,\ R} q_8$$

a	q_7	b
a	a	q_8

b	q_7	b
b	a	q_8

c	q_7	b
c	a	q_8

d	q_7	b
d	a	q_8

$\sqcup$	q_7	b
$\sqcup$	a	q_8

PER OGNUNA DI QUESTE FINESTRE C'È UNA FORMULA CHE LA DESCRIVE

b	q_7	b
b	a	q_8

$$x_{i,j-1,b} \wedge x_{i,j,q_7} \wedge x_{i,j+1,b} \wedge$$

$$x_{i+1,j-1,b} \wedge x_{i+1,j,a} \wedge x_{i+1,j+1,q_8}$$

W_{37}

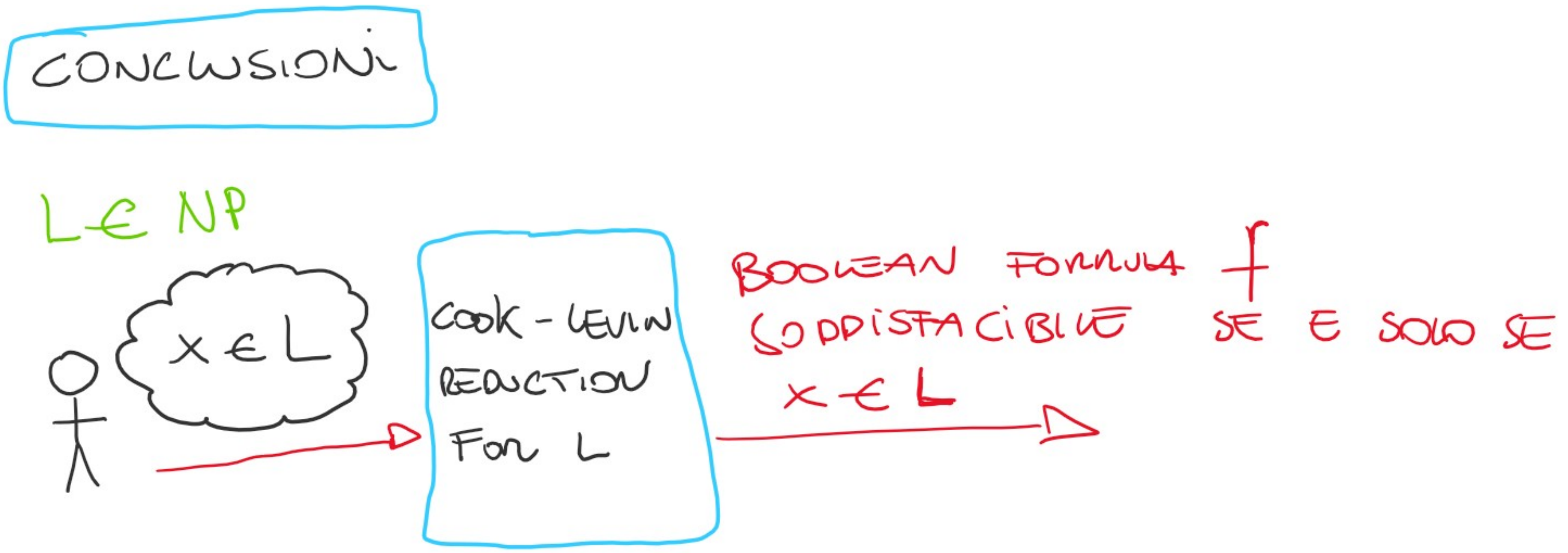
DATO UN i,j DEVE CORRISPONDERE UNA DELLE FINESTRE AMMESSE

$$W_1 \vee W_2 \vee \dots\dots \vee W_{37} \vee \dots\dots \vee W_{532}$$

DESCRIVE LA FINESTRA

$$\text{DELTA} = \bigwedge_{1 \le i,j \le p(n)} \left(\bigvee_{\substack{\text{TUTTE LE} \\ \text{532 FINESTRE}}} \begin{pmatrix} x \wedge x \wedge x \wedge \\ x \wedge x \wedge x \end{pmatrix} \right)$$

SAT è NP-completo



SAT appartiene ad NP

esiste una macchina MdT non deterministica che può decidere tutti i problemi in SAT.

Ogni linguaggio in NP è riducibile a SAT in tempo polinomiale


P = NP

Riduzione: VC $\longrightarrow$ HC

↓

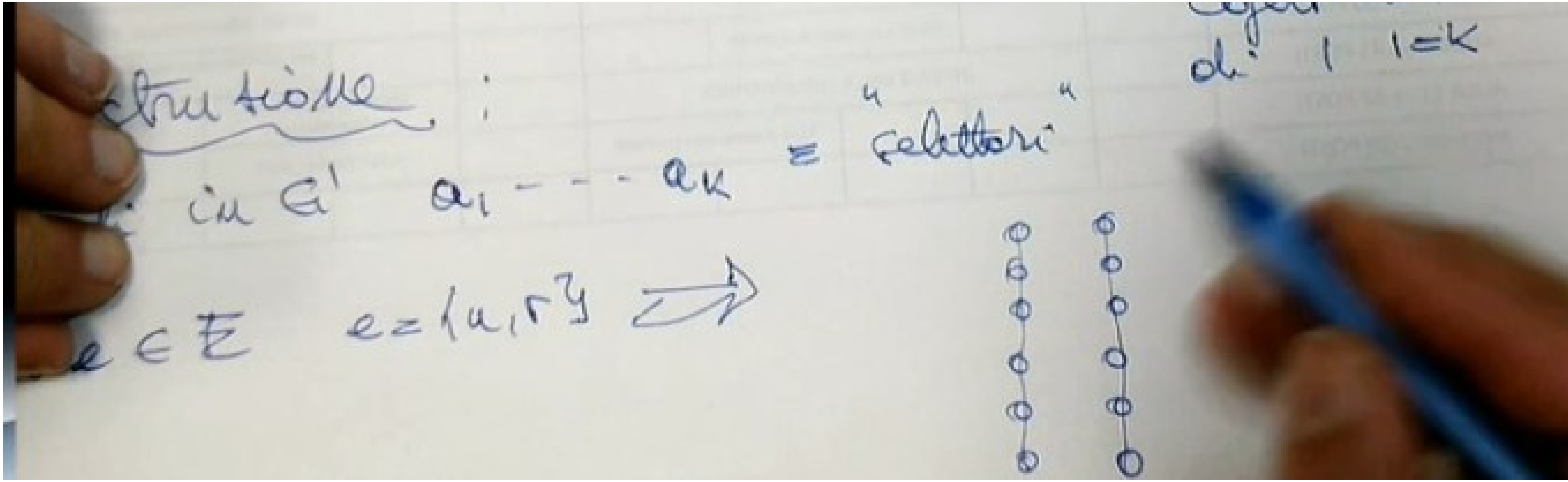
sia NP-Completo

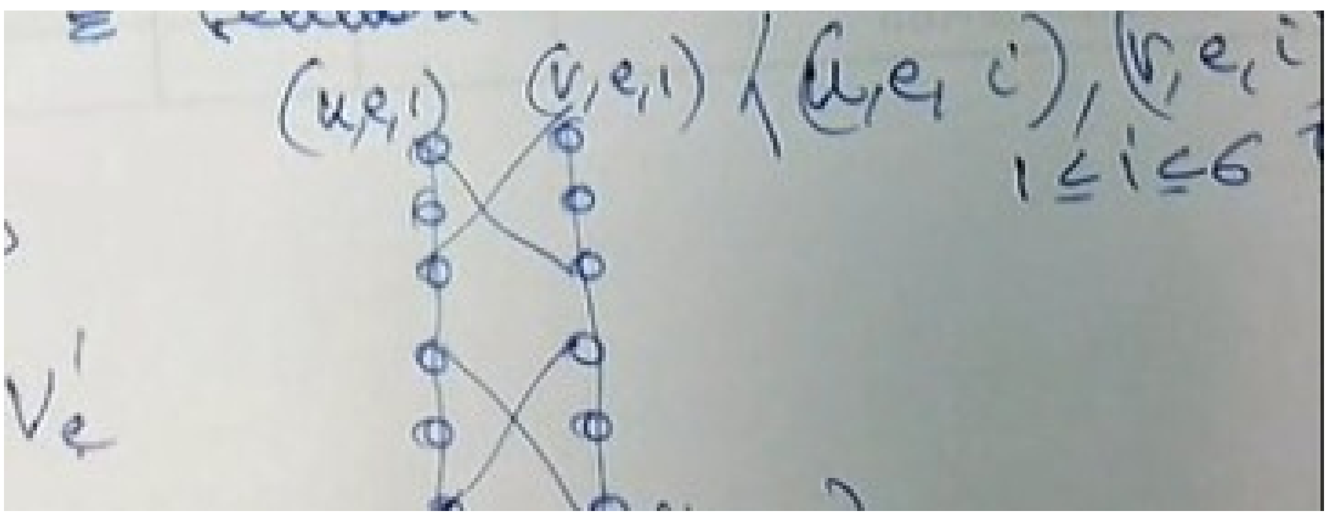
Ist di Vertex Cover: $G=\langle V,E\rangle$, $K\leq|V|$

↓

Ist di HC $G'=\langle V',E'\rangle$

in G' $\exists$ un circ. H sse in G c'è una copert. di vertici di $|\ |=K$


struzione :
in G' $a_1 - - - a_K$ = "selettori"
$e \in E$ $e=\{u,v\} \Longrightarrow$
di 1 $i \leq K$


$(u,e,1)$ $(v,e,1)$ $\{(u,e,i),(v,e,i$
$1 \leq i \leq 6$
V'_e