LA CLASSE P

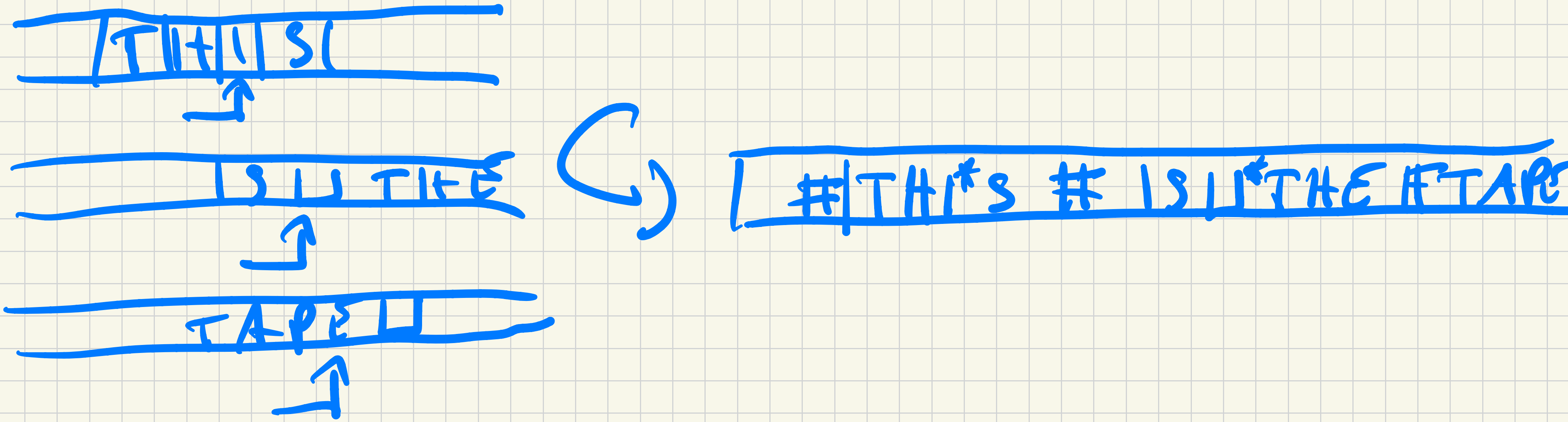
Prima di definirla, vediamo alcuni esempi relativi alla complessità di tempo.

Varianti del modello TM:

- la TM M che può fare {L, R, S} si può simulare con TM M' "classica" e la complessità di tempo sarà $\leq 2T(n)$ $= O(T(n))$.

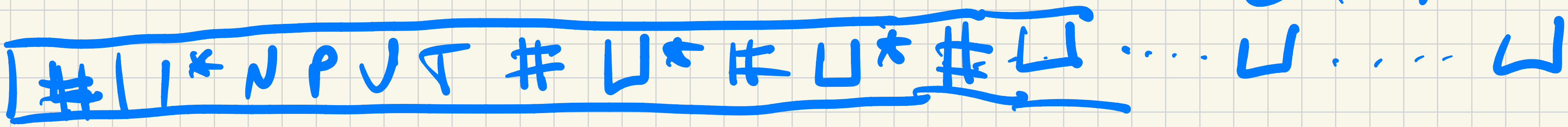
- la TM M con K nastri (e.g. $K=3$) si può simulare con TM M' "classica" e la complessità di tempo sarà $O(T^2(n))$.

Vediamo perché:



Inizializzazione:

Tempo:
$\Theta(n)$



Simulazione:

Per simulare 1 passo devo scansionare il nastro e ricordarmi i simboli marcati e aggiornare di conseguenza nastri e posizioni testine. Inoltre se necessario devo liberare una cella traslando tutto a DX di 1 pos.

Quanto è lungo il nastro? Il tempo limite lo spero! Quindi al massimo sarà $3\,T(n) = O(T(n))$.

$\#$ 1 passo $\Rightarrow O(T(n))$

Made with Goodnotes

$\# T(n)$ passi $\Rightarrow O(T^2(n))$

Totale $\quad O(n) + O(T^2(n))$

$$= O(T^2(n))$$

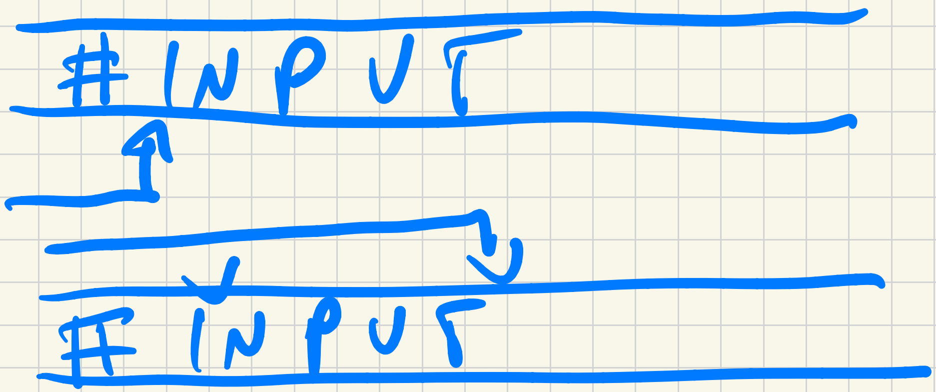
$$(T(n) \gg n).$$

Nella pratica il modello di TM può avere un impatto importante. Ad es.

Consideriamo il linguaggio PALINDROMES delle parole palindrome su $\{a,b\}^*$.

Se un vocabolario: PALINDROMES è decidibile da TM singolo nastro in tempo $O(n^2)$.

Con 2 nastri: tempo $O(n)$



THM Ogni TM singolo nastro necessita $\Omega(n^2)$ tempo per decidere PALINDROMES.

Per questo corso, queste differenze non contano.

DEF Sia $t : \mathbb{N} \to \mathbb{R}^+$, definiamo:

$$DTIME(t(n)) =$$

$$= \{ L : \exists \text{ TM che decide } L \text{ in tempo } O(t(n)) \}$$

Ovvero, $PALINDROMES \in DTIME(n^2)$

$$\notin DTIME(n).$$

DEF. P è la classe di linguaggi decidibili da TM in tempo polinomiale:

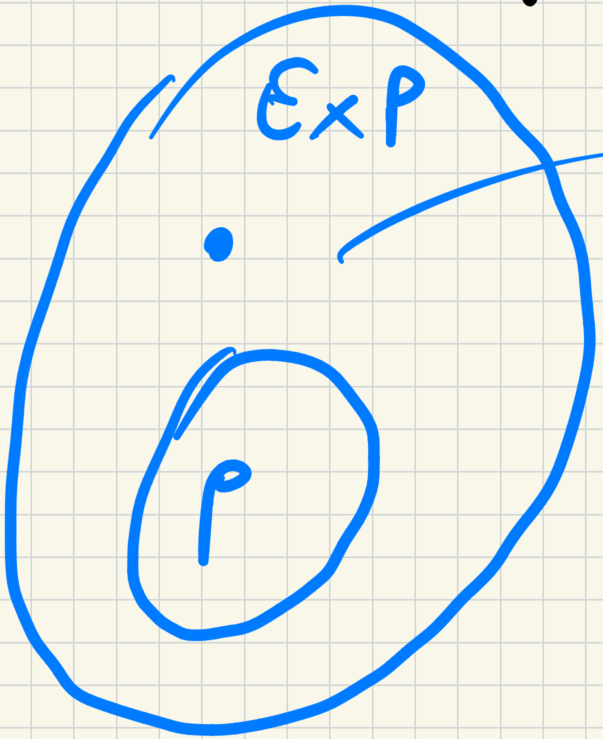
$$P = \bigcup_{K \in \mathbb{N}} DTIME(n^K)$$

Questa definizione è robusta perché è invariante rispetto al modello di TM.

DEF EXP è la classe di linguaggi decidibili da TM in tempo esponenziale:

$$EXP = \bigcup_{K \in \mathbb{N}} DTIME(2^{n^K})$$

Una cosa importante che mostreremo:



$\exists\, L$ t.c. $L \notin P$

ma $L \in EXP$

$(P \neq EXP)$.

TEOREMA DI GERARCHIA DI TEMPO.

$P \subsetneq EXP$

La classe EXP è molto grande, ma ci sono linguaggi banalmente in EXP che in realtà si dimostrano essere in P usando strategie più smart.

Esempio:

grafo diretto.

- $PATH = \{ \langle G, s, t \rangle : \exists \; s \rightsquigarrow t \text{ in } G \}$

$\langle G \rangle$ è una qualunque, e.g. la matrice di adiacenza.

Tipicamente in questo caso è più

calcolo contare la complessità in funzione di $n = |V|$ e $m = |E|$ dove $G = (V, E)$.

$s \rightsquigarrow t$: $v_0, v_1, \dots, v_\ell$ t.c. $v_0 = s$

$v_\ell = t$ e $(v_i, v_{i+1}) \in E$.

La lunghezza del cammino è $\leq n$ e

inoltre # cammini $\leq n^n = 2^{n \log n}$

ovvero PATH $\in$ EXP.

In realtà, PATH $\in$ P:

– Marca s.

- Ripeto:
  - Scansiono tutti gli archi (u, v)
  - Se u è marcato e v NO, marco v.
  - Mi fermo quando smetto di marcare.
- Accetto sse t è marcato.

Tempo $O(n \cdot m) = O(n^3)$

$m = O(n^2)$.

PATH $\in$ P.

Altro esempio:

- $2COL = \{ \langle G \rangle : G$ è 2-colorabile$\}$.

Scopo: assegnare ai vertici "BLU" o "ROSSO" in modo che nodi connessi da archi abbiano colori diversi.

possibili 2-coloring $O(2^n)$

$\Rightarrow$ $2COL \in EXP$.

Ma in realtà $2COL \in P$:

- Prendo un vertice e lo coloro BLU.
- Poi coloro i suoi vicini ROSSI.
- Poi coloro i vicini BLU
- ....

Se trovo contraddizione rifiuto. Altrimenti ripeto per ogni componente connessa in G.
Se ho colorato tutto, accetto.

→ 3COL? Ovviamente 3COL $\in$ EXP ma nessuno sa se 3COL $\in$ P! Miglior algoritmo ha tempo $O(1.333^n)$.

– Altro esempio: CLIQUE. Tutti i grafi che contengono triangolo ($\triangle$).

CLIQUE $\in$ P: # $\triangle$ $O(n^3)$ e controllare se è $\triangle$ costa $O(1)$.

Nessuno sa se CLIQUE si può decidere in $O(n^2)$.

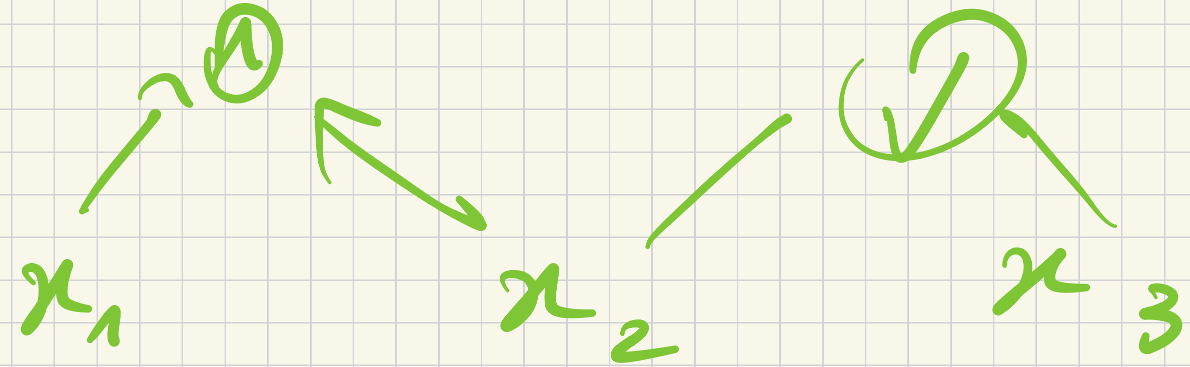
CODICE OPIS : 9304RZSJ

SATISFIABILITY

Problema fondamentale in teoria della complessità. Ci sono tante varianti.

DEF Un circuito booleano C è un grafo diretto ACICLICO con n input $x_1 \dots x_n$ e 1 output. I vertici si chiamano porte: $\land$, $\lor$, $\neg$.

FAN-OUT ARBITRARIO.



Se FAN-OUT $= 1$, C è una FORMULA.

DEF CIRCUIT-EVAL $= \{ \langle C, x \rangle : C(x) = 1 \}$.

$\langle C \rangle$: una qualsiasi codifica in funzione di # input, # porte.

$$|\langle C \rangle| = O(n \log n)$$

CIRCUIT-EVAL $\in P$.

DEF $\text{CIRCUIT-SAT} = \{ \langle C \rangle : \exists\, x \in \{0,1\}^n \text{ t.c. } C(x) = 1 \}$.

Ovviamente, CIRCUIT-SAT $\in$ EXP perché posso decidere in tempo

$$O(\text{poly}(n) \cdot 2^n)$$

CIRCUIT-SAT $\in P$? Non lo sa nessuno

[Spoiler: CIRCUIT-SAT $\in P$ sse

P = NP.)

Varianti:

- FORMULA-SAT: C è una formula.
- CNF-SAT: C è una CNF cioè un grande "$\wedge$" di clausole ("$\vee$" di letterali).
- K-SAT: tutte le clausole hanno k letterali.

E.S. 3-SAT ovvero $K=3$. Ovviamente 3-SAT $\in$ EXP. (Ma 3-SAT $\in$ P sse P = NP. Stesse cose per CNF-SAT $\leq$ FORMULA-SAT.)

Migliori algoritmi:

- 3SAT $\in$ DTIME$(1,34^n)$

- 4SAT $\in$ DTIME$(1,5^n \cdot poly(n))$

- ...

THM $2SAT \in P$.

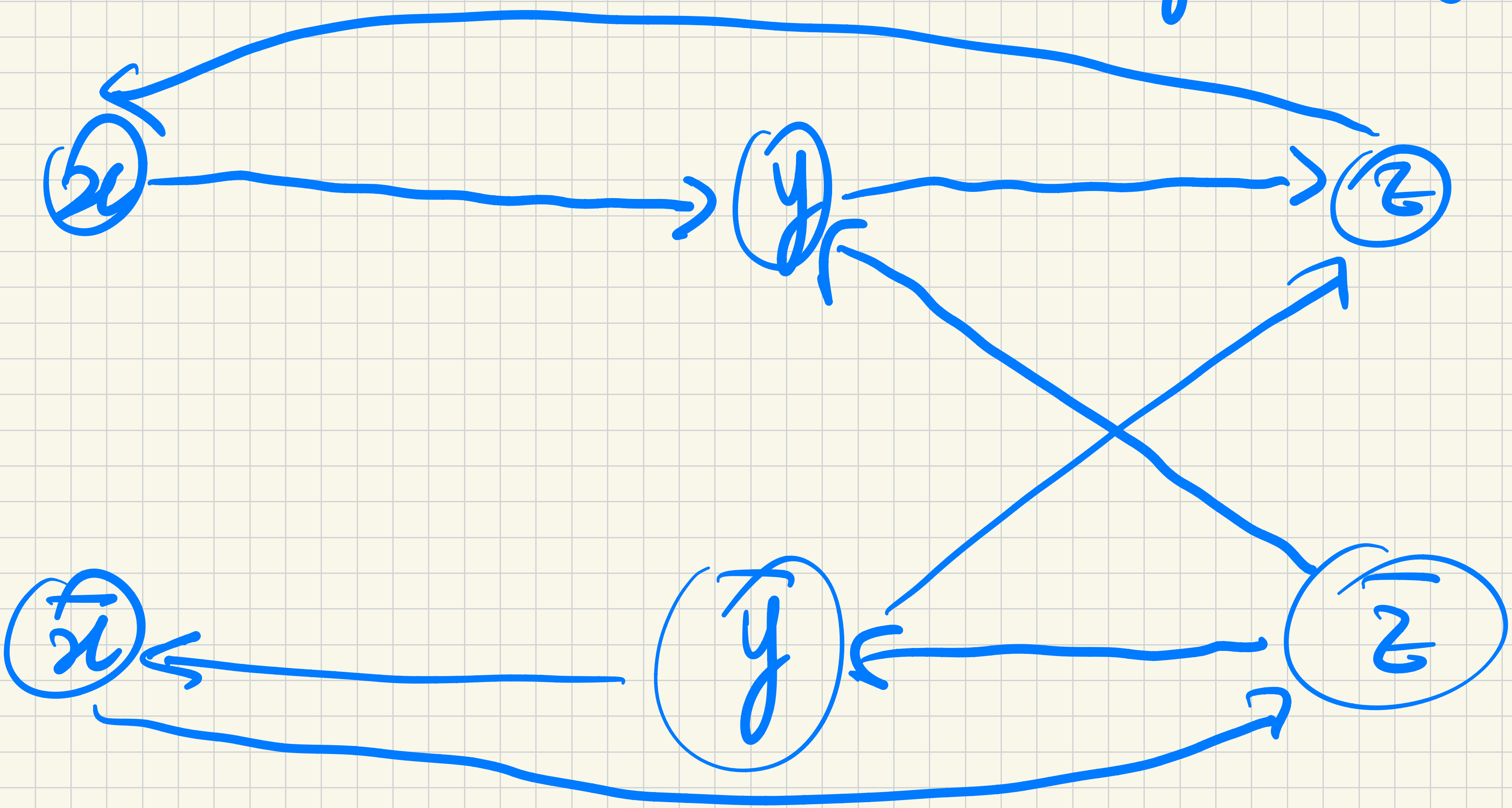
Proof. Sia $\phi(x_1, \dots, x_n)$ la nostra formula 2SAT con n variabili ed m clausole. Faremo corrispondere alla formula un grafo. Per ogni clausola $x \vee y$ considero 2 implicazioni logiche: che equivalenti "$\bar{x} \to y$" e "$\bar{y} \to x$"

x	y	$x \vee y$	$x \to y$
0	0	0	1
0	1	1	1
1	0	1	0

1	1	1	1
		$\bar{y} \to x$	$\bar{x} \to y$
		0	0
		1	1
		1	1
		1	1

Per ogni $l_1 \vee l_2$ in ϕ costruisco grafo G con nodi $\{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n\}$ aggiungendo archi $\bar{l}_1 \to l_2$ e $\bar{l}_2 \to l_1$

<u>Es</u> : $\phi(x) = (\bar{x} \vee y) \wedge (\bar{y} \vee z)$
$\wedge (x \vee \bar{z})$
$\wedge (y \vee z)$



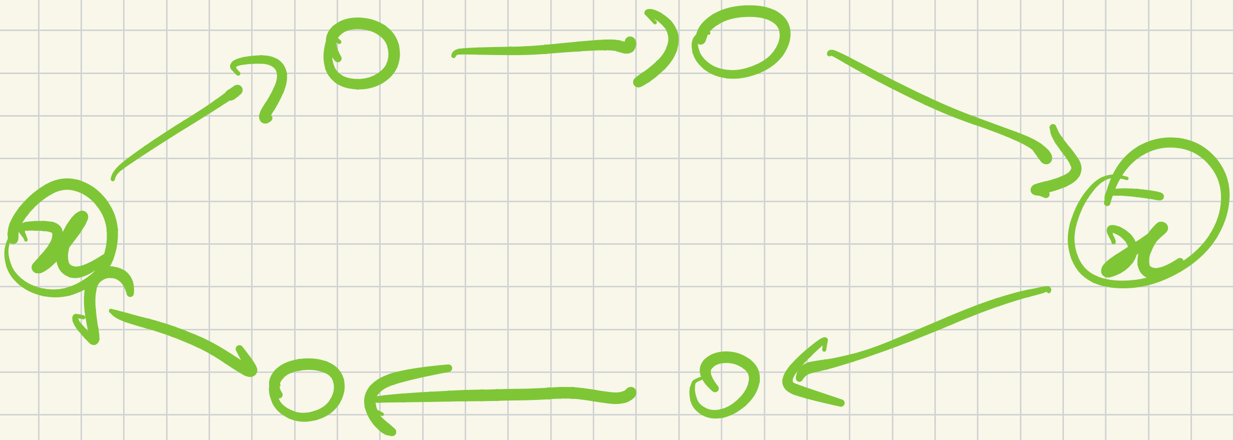
LEMMA Φ è soddisfacibile sse nessuna componente fortemente connessa di G contiene una variabile e la sua negazione.

DIM. Fortemente connesso: ogni nodo raggiungibile e partire da qualsiasi altro nodo.

($\Longrightarrow$) Sia Φ soddisfacibile e ab un arco in G. Se $a = T$

b pure deve essere T, questo perché nella ϕ c'è $(\bar{a} \vee b)$.

Se nel grafo c'è il cammino $x \rightsquigarrow \bar{x}$ allora è necessario che x sia F.



Ma allora, non può esserci allo stesso tempo il cammino $x \rightsquigarrow \bar{x}$ e $\bar{x} \rightsquigarrow x$.

Questo perché, per lo stesso argomento

$\bar{x}$ dovrebbe essere F ovvero x è V.

$\rightarrow\leftarrow$

$\Rightarrow$) Ogni componente fortemente connessa non contiene x e $\bar{x}$.

($\Leftarrow$) Assumiamo nessuna componente connessa comprenda $x, \bar{x}$. Ordiniamo le componenti fortemente connesse $C_1, \dots, C_t$

Assegnamento: $\forall\, x$, setto $x = T$ sse x appare dopo di $\bar{x}$.

Altrimenti, $x = F$.

OSS Per nessun arco ab di G il vertice a è assegnato T e b è assegnato F.

Questo implica che la ϕ è soddisfatta.

Supponiamo ab arco con $a = T$ e $b = F$. e che $a \in C_i$.

L'arco ab è presente perché c'è la clausola $\bar{a} \vee b$ che ha pure generato $\bar{b}\bar{a}$. Siccome $a = T$ e $\bar{a} = F$ allora $\bar{a}$ appare prima ovvero $\bar{a} \in C_j$ con $j < i$.

Analogamente, siccome $b = F$ e $\bar{b} = V$ allora $\bar{b}$ appare dopo C_j ovvero $\bar{b} \in C_k$ per $k > j$.

$\Rightarrow$ l'arco ab contraddice l'ordine delle componenti. Perché ci sarebbe un arco tra C_k e C_j

È dovuto all'arco $\bar{b}\bar{a}$ $\rightarrow\leftarrow$ $\boxtimes$