NP-Hard and NP-Complete

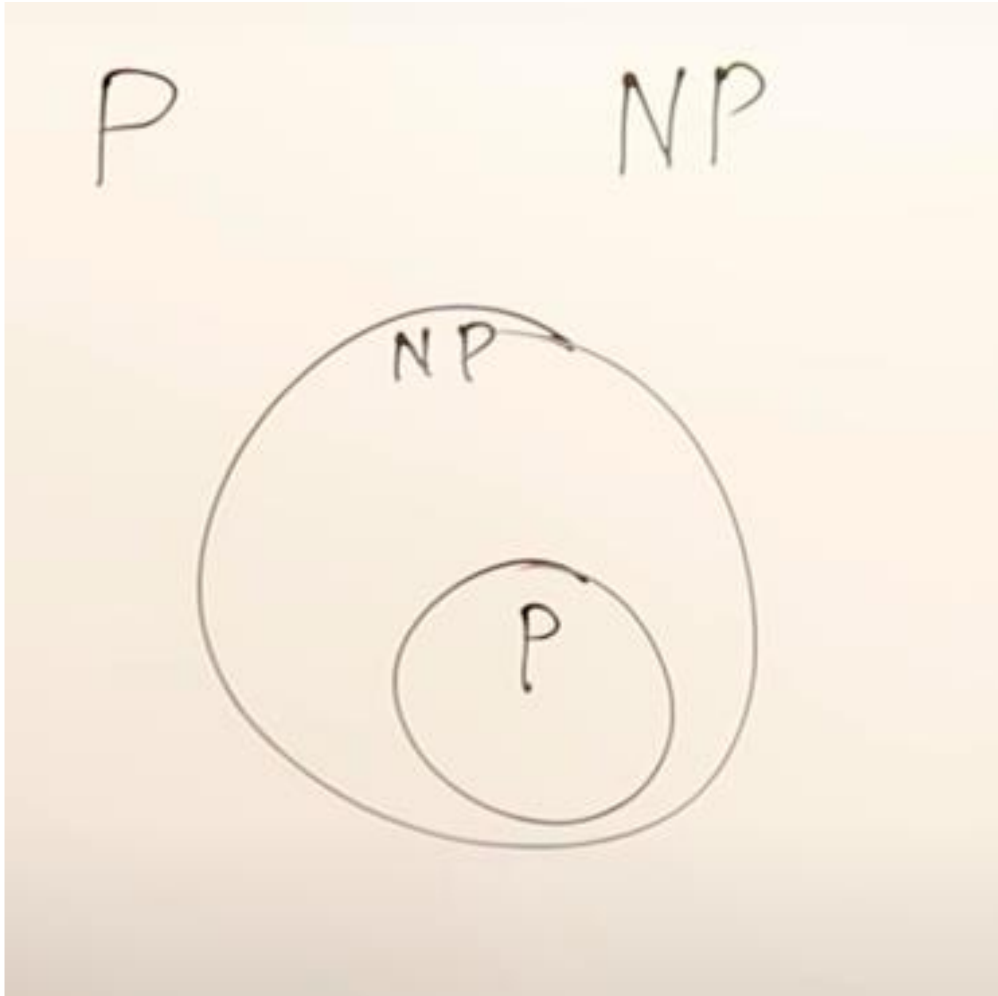
ploynomial Time	Exponential Time
Linear Search — n	0/1 Knapsack — 2^n
Binary Search — $\log n$	Traveling SP — 2^n
Insertion Sort — n^2	Sum of Subsets — 2^n
Merge Sort — $n \log n$	Graph Coloring — 2^n
Matrix Multiplication — n^3	Hamiltonian Cycle — 2^n

Nondeterministic

```
Algorithm NSearch(A, n, key)
{
    j = choice();        ——— 1
    if (key = A[j])
    {
        write(j);
        Success();       ——— 1
    }
    write(0);
    Failure();           ——— 1
}
```

Exponential Time

- 0/1 Knapsack — 2^n
- Traveling SP — 2^n
- Sum of Subsets — 2^n
- Graph Coloring — 2^n
- Hamiltonian Cycle — 2^n


P
NP
NP
P

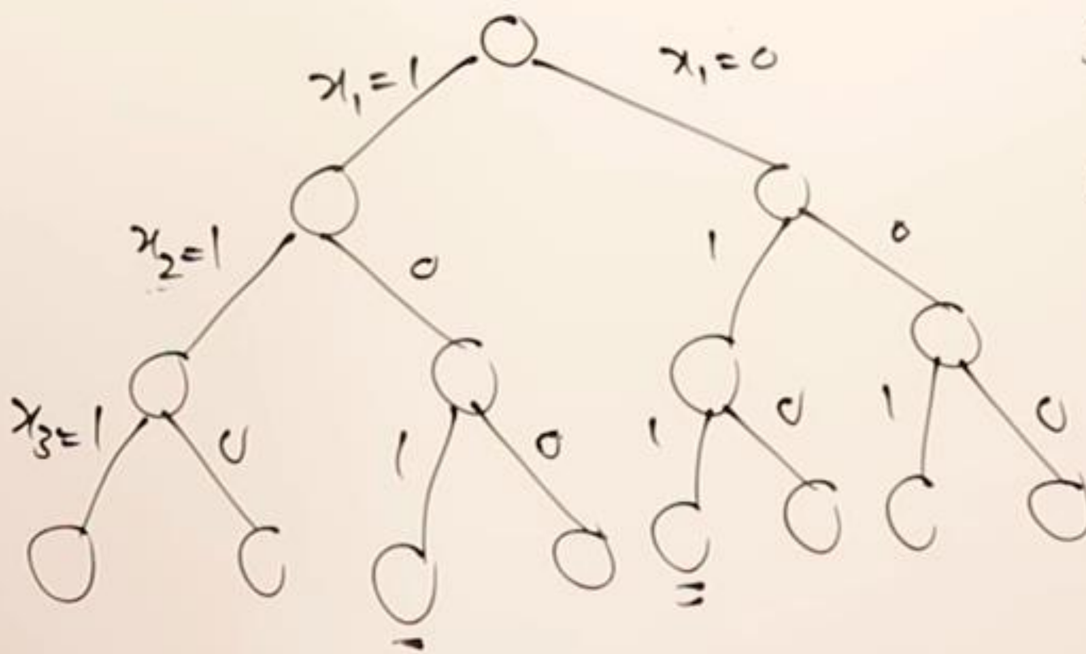
NP-Hard and NP-complete

CNF-Satisfiability — 2^n

$x_i = \{x_1, x_2, x_3\}$

$$CNF = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$$

C_1 $\qquad$ C_2



Exponential Time

- 0/1 Knapsack — 2^n
- Traveling SP — 2^n
- Sum of Subsets — 2^n
- Graph Coloring — 2^n
- Hamiltonian Cycle — 2^n

NP-Hard and NP-Complete

CNF-Satisfiability — 2^n

0/1 Knapsack

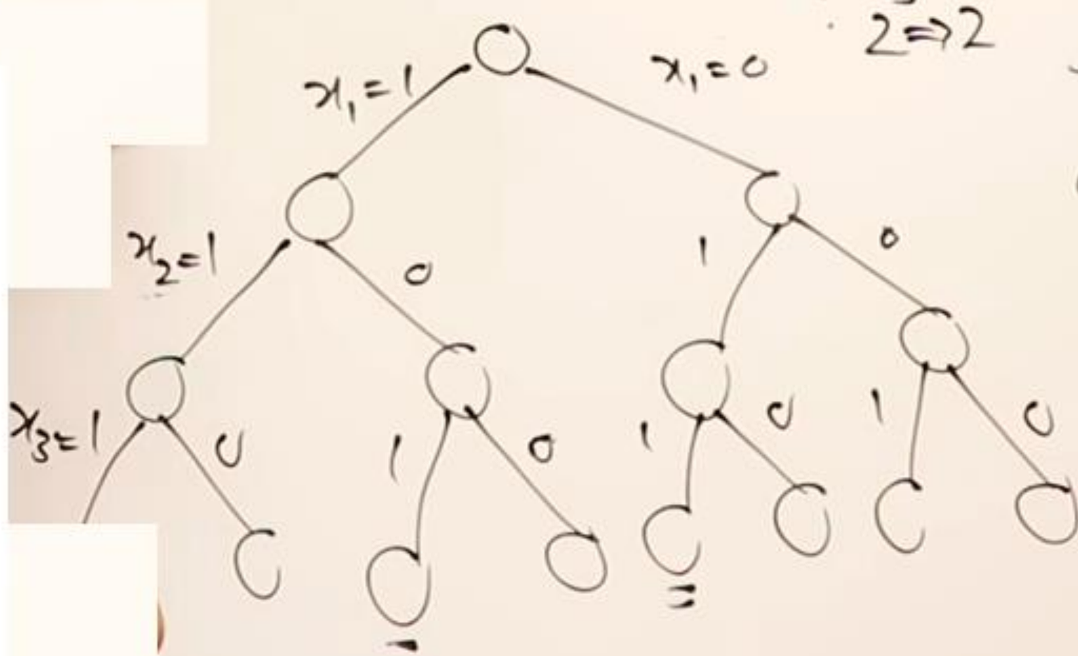
$P = \{10, 8, 12\}$ $n = 3$

$w = \{5, 4, 3\}$ $m = 8$

$x_i = \{0/1, 0/1, 0/1\}$

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
	⋮	

$2^3 \Rightarrow 2^n$



Exponential Time

0/1 Knapsack — 2^n

Traveling SP — 2^n

Sum of Subsets — 2^n

Graph Coloring — 2^n

Hamiltonian Cycle — 2^n

NP-Hard and NP-complete

Satisfiability — 2^n

Reduction

Exponential Time

0/1 Knapsack — 2^n

Sat $\propto$ 0/1 Knapsack

Traveling SP — 2^n

I_1 I_2

Sum of Subsets — 2^n

Graph Coloring — 2^n

Hamiltonian Cycle — 2^n

NP-Hard and NP-complete

NP-Hard $\rightarrow$ Satisfiability $- 2^n$

Reduction

NP-Hard Poly NP-Hard

$$\text{Sat} \overset{\text{Poly}}{\propto} \text{0/1 Knapsack}$$

I_1 I_2

$$\text{Sat} \propto \underset{\text{NP-Hard}}{L}$$

$$\text{Sat} \propto L_1 \qquad L_1 \propto L_2$$

Expontial Time

0/1 Knapsack $- 2^n$

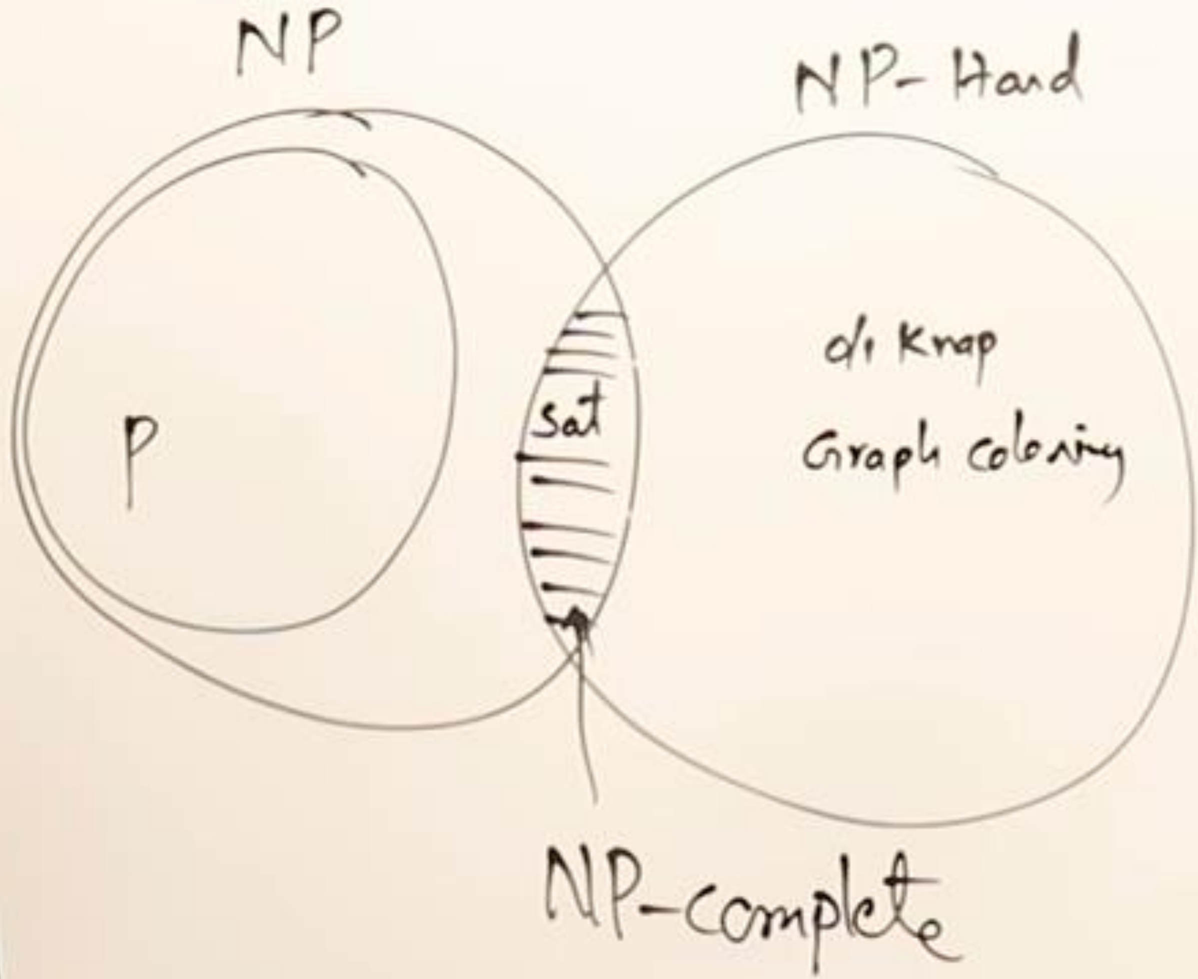
Traveling SP $- 2^n$

Sum of Subsets $- 2^n$

Graph Coloring $- 2^n$

Hamiltonian Cycle $- 2^n$

NP-Hard and NP-complete



Cook's Theorem

Theorem: The Satisfiability Problem (SAT) is NP-Complete

What is SAT?

A propositional logic formula ϕ is called Satisfiable if there is some assignment to its Variable that makes it evaluate to true.

→ $P \wedge q$ is satisfiable $P=1, q=1$
→ $P \wedge \neg P$ → $P=1$

⇒ SAT

A language $3SAT = \{\phi \mid \phi$ is Satisfiable $\underbrace{3\text{-CNF}}$ formula$\}$

Note:

In Boolean (0 or 1)

operator → Not $\neg$ (highest)

And $\cdot$

or $+$

3SAT

A language $3SAT = \{\phi \mid \phi$ is Satisfiable 3-CNF formula$\}$

3-CNF ↓

- It is in CNF (Conjunctive Normal Form) and Every clause has exactly 3 literals

$$\underbrace{(x \vee y \vee z)}_{\text{clause}} \wedge (\neg x \vee \neg y \vee z)$$

clause → OR

literals → x or $\neg x$

Theorem : SAT is NP-Complete

Proof consists of two step

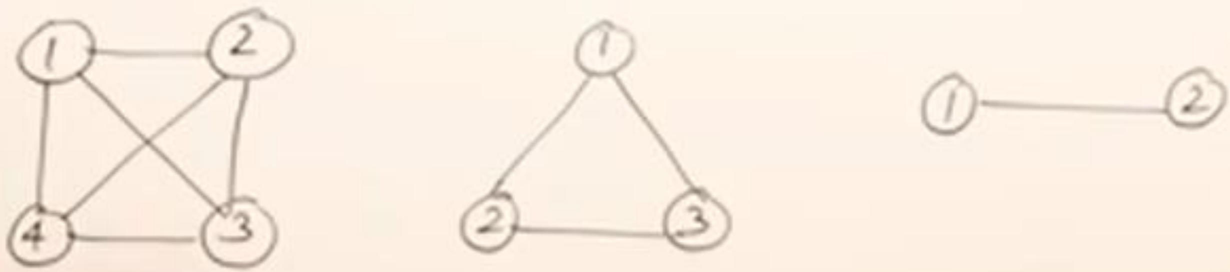
1) Convert the execution of a polynomial time NDTM to a bunch of well formed formulae such that formula satisfie iff the machine accept input

2) Show the sum of lengths of formulae is polynomial in the size of problem

- NP-Hard (L) - Can polynomially reduce any NP problem to L
- NP Complete - $L \in NP$
- $L \in NP \rightarrow$ NDTM for L that runs in polyn[illegible]

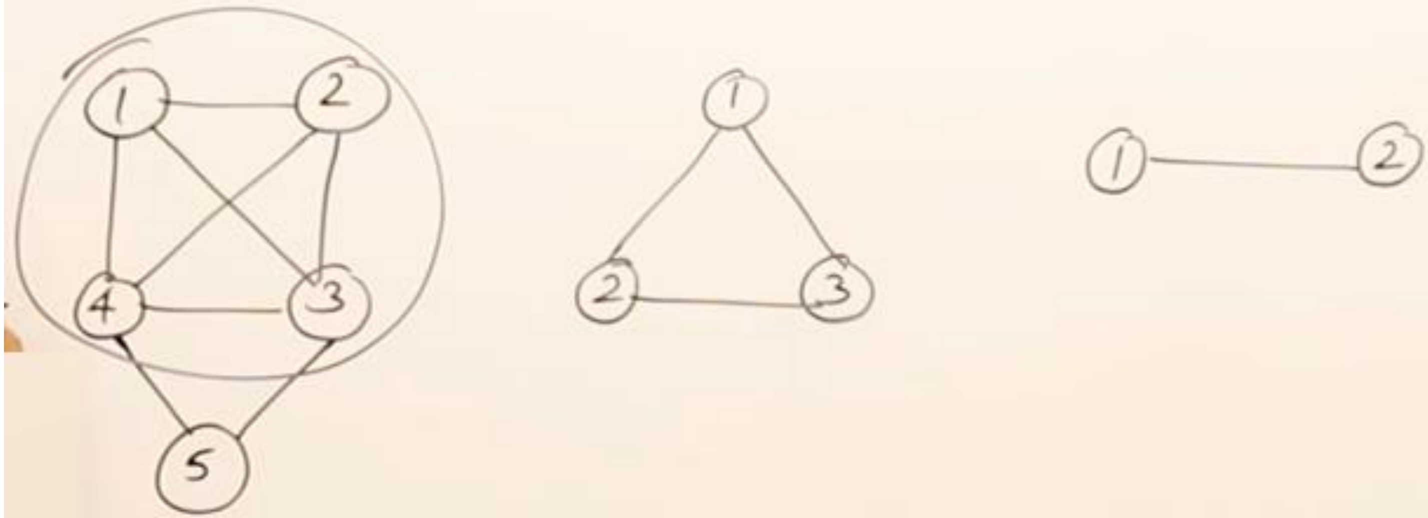
- NP-Hard(L) - Can polynomially reduce any NP problem to L
- NP-Complete - $L \in NP$
- $L \in NP \longrightarrow$ NDTM for L that runs in polynomial time
- An NDTM is the only model we have for NP Problem
- $SAT \in NP$
- Therefore, if we can polynomially reduce and an arbitrary polynomial NDTM to SAT, Its mean we have proven SAT is NP-Complete.

Clique Decision Problem (CDP)



1. what is clique?
2. what are Decision & Optimization Problems
3. what are NP-Hard Graph Problems
4. Procedure to Prove NP-Hard
5. Prove CDP is NP-Hard.

Clique Decision Problem (CDP)

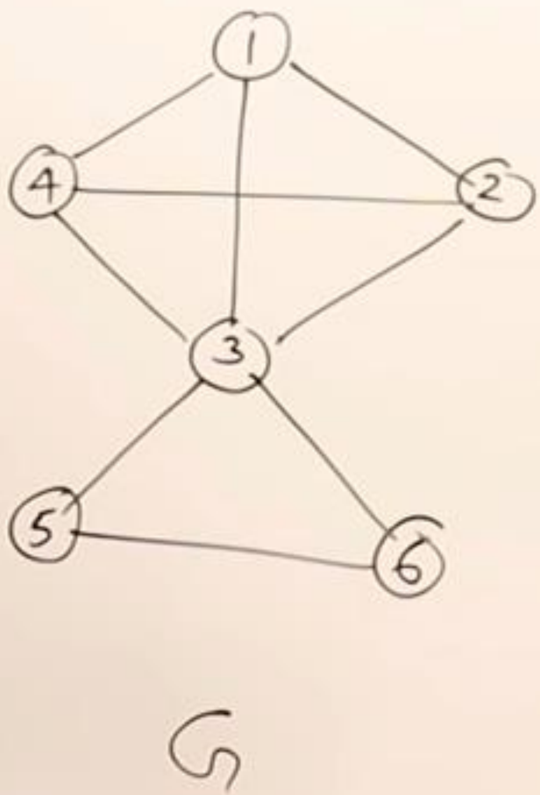


Complete Graph

$$|V| = n$$

$$|E| = \frac{n(n-1)}{2}$$

Clique Decision Problem (CDP)



$K = 4$

$K = 3$

$K = 2$

G

Decision Problem: find if a graph is having a clique of size k

Optimization Problem: find max clique in a graph

Clique Decision Problem (CDP)

$$P = L_2$$

$$\begin{array}{ccc} \text{Sat} & \propto & \text{CDP} \\ \downarrow & & \downarrow \\ I_1 & & I_2 \end{array}$$

Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1, x_2, x_3

$$F = \bigwedge_{i=1}^{K} C_i$$

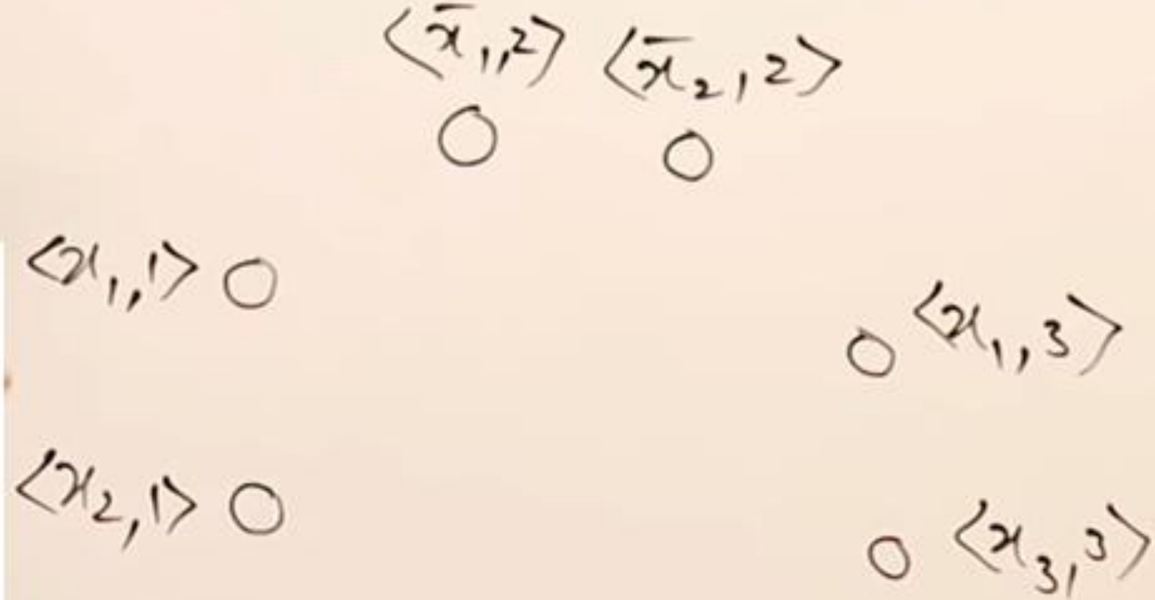
$$F = \underbrace{(x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{C_2} \wedge \underbrace{(x_1 \vee x_3)}_{C_3}$$

Sat $\propto$ CDP

x_1, x_2, x_3

$$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$$

$C_1 \qquad C_2 \qquad C_3$



Clique Decision Problem (CDP)

$$\text{Sat} \propto \text{CDP}$$

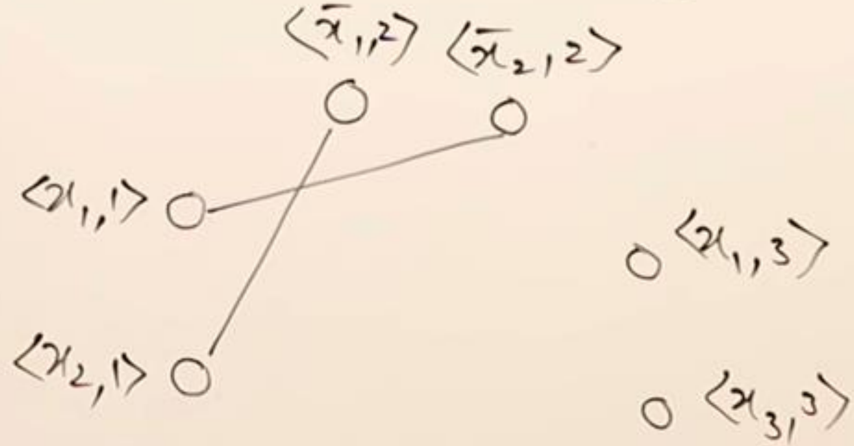
x_1, x_2, x_3

$$F = \bigwedge_{i=1}^{K} C_i$$

$$F = \underbrace{(x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{C_2} \wedge \underbrace{(x_1 \vee x_3)}_{C_3}$$

G $\quad V = \{ \langle a, i \rangle \mid a \in C_i \}$

$$E = \{ (\langle a,i \rangle, \langle b,j \rangle) \mid i \neq j \text{ and } b \neq \bar{a} \}$$

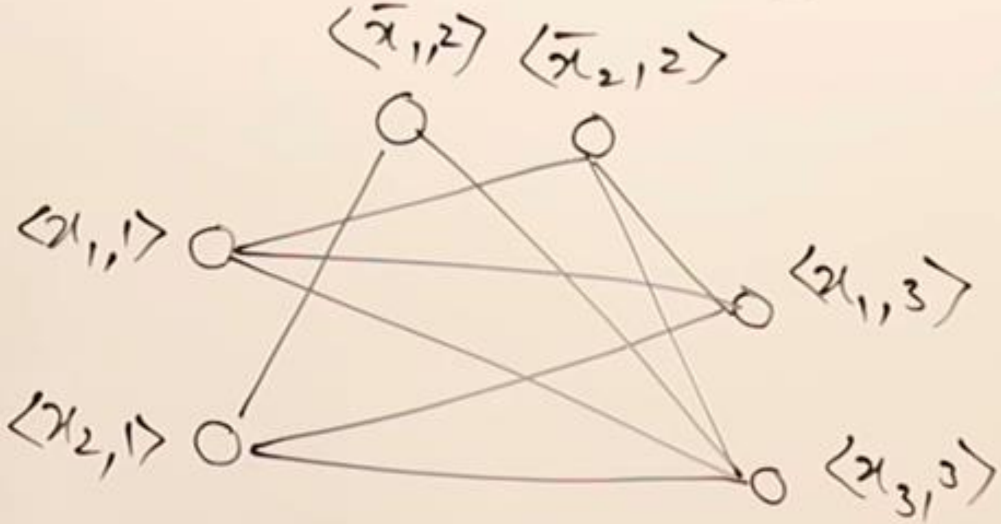


Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1, x_2, x_3

$$F = \underbrace{(x_1 \vee x_2)}_{C_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{C_2} \wedge \underbrace{(x_1 \vee x_3)}_{C_3}$$



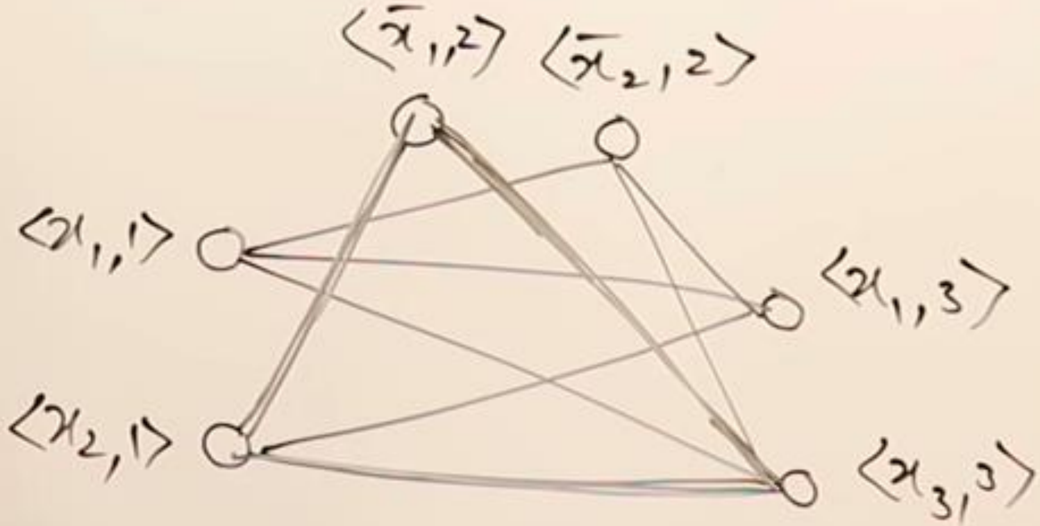
Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1, x_2, x_3 $\quad$ $K = 3$

$$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (x_1 \vee x_3)$$

C_1 $\quad$ C_2 $\quad$ C_3



$k = 3$

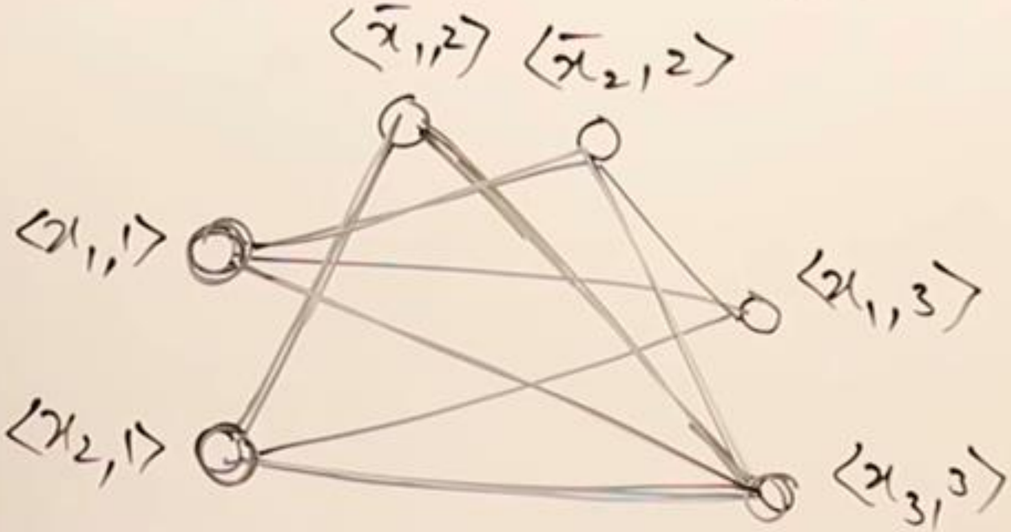
Clique Decision Problem (CDP)

Sat $\propto$ CDP

x_1, x_2, x_3 $\quad$ $k=3$

$$F = (\overset{0}{x_1} \vee \overset{1}{x_2}) \wedge (\overset{1}{\bar{x}_1} \vee \overset{0}{\bar{x}_2}) \wedge (\overset{0}{x_1} \vee \overset{1}{x_3}) = 1$$

$C_1 = 1$ $\quad$ $C_2 = 1$ $\quad$ $C_3 = 1$



$k=3$

x_1	x_2	x_3
0	1	1

Thank you