- Weknow: 3-SAT is NP hard.
- We want to show that CLIQUE is also NP hard.

i.e, If 3-SAT is hard, how to show CLIQUE is also hard?

- Enough to show: if CLIQUE $\in P$, then 3-SAT $\in P$.

i.e., Show a polynomial-time reduction from 3-SAT to CLIQUE.

-Inputs are of different types.

3-SAT: Boolean formula ϕ (that contains 3-literal clauses).

CLIQUE: Graph G, integer k.

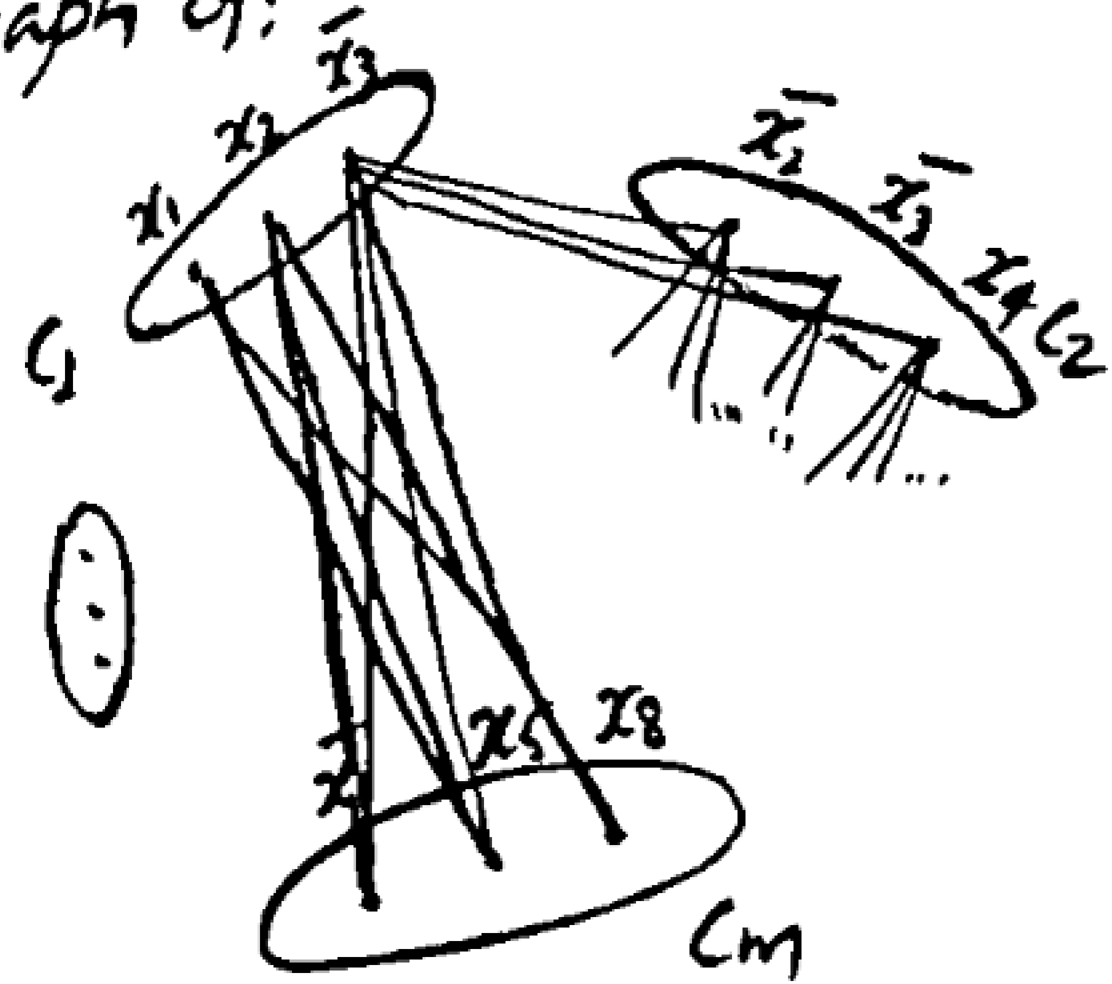
Constructing the graph G:

$C_1: \; x_1 + x_2 + \bar{x}_3$

$C_2: \; \bar{x}_2 + \bar{x}_3 + x_4$

$\vdots$

$C_m: \; \bar{x}_1 + x_5 + x_8$



- 3 vertices for every clause;
  one for each literal in that clause.

- Add edges between literals appearing in 2 different clauses, but no edge if one is a negation of the other. (eg: x_1 and $\bar{x}_1$)

- What is k?

$k = m$, the total no. of clauses.

<u>Claim:</u> 3-SAT is satisfiable

if and only if

G has a clique of size k.

Proof: Argue both ways.

(i) G has a clique of size k $\Rightarrow$

Make all the literals corresponding to the k vertices true. ($k = m$)

One literal each in every clause becomes true. $\Rightarrow \phi$ evaluates to true.

(YES answer is correct)

- Possible because we did not add edge $x_i - \bar{x}_i$.

(ii) To show 'NO' answer is correct, It is enough to show that if ϕ is satisfiable, there must be a clique of size m in the graph G. (can not say NO).

For this, pick one literal from each clause that becomes true (in a satisfying truth assignment of ϕ). $\square$

· Some other hard problems:

Set cover: Given in the exam.

Input: A set $U = \{x_1, x_2, \ldots\ x_n\}$

and m subsets,

$S_1, S_2, \ldots S_m \subseteq U$.

and a number k.

Question: Can k subsets 'cover' U?

- Note that vertex cover problem can be reduced to set cover, in poly. time.

$$VC \leq_p SC.$$

- VC input instance : Graph G, number k.
- SC instance: U = set of all edges of G.
  Subsets: For each vertex v, the set of edges incident on v.

- Assuming SAT is NP-hard, we showed

3-SAT, CLIQUE, Ind. Set, Vertex Cover, Set cover are all NP-hard.

- Their other versions also:
  - Find a clique of size k.
  - Find the size of the maximum clique.

- Decision version:

  Is there a clique of size k in G?

- Optimization version:
  - Find the size of the max. clique.
  - Find the max. sized clique.

(For vertex cover, find the min. vertex cover).