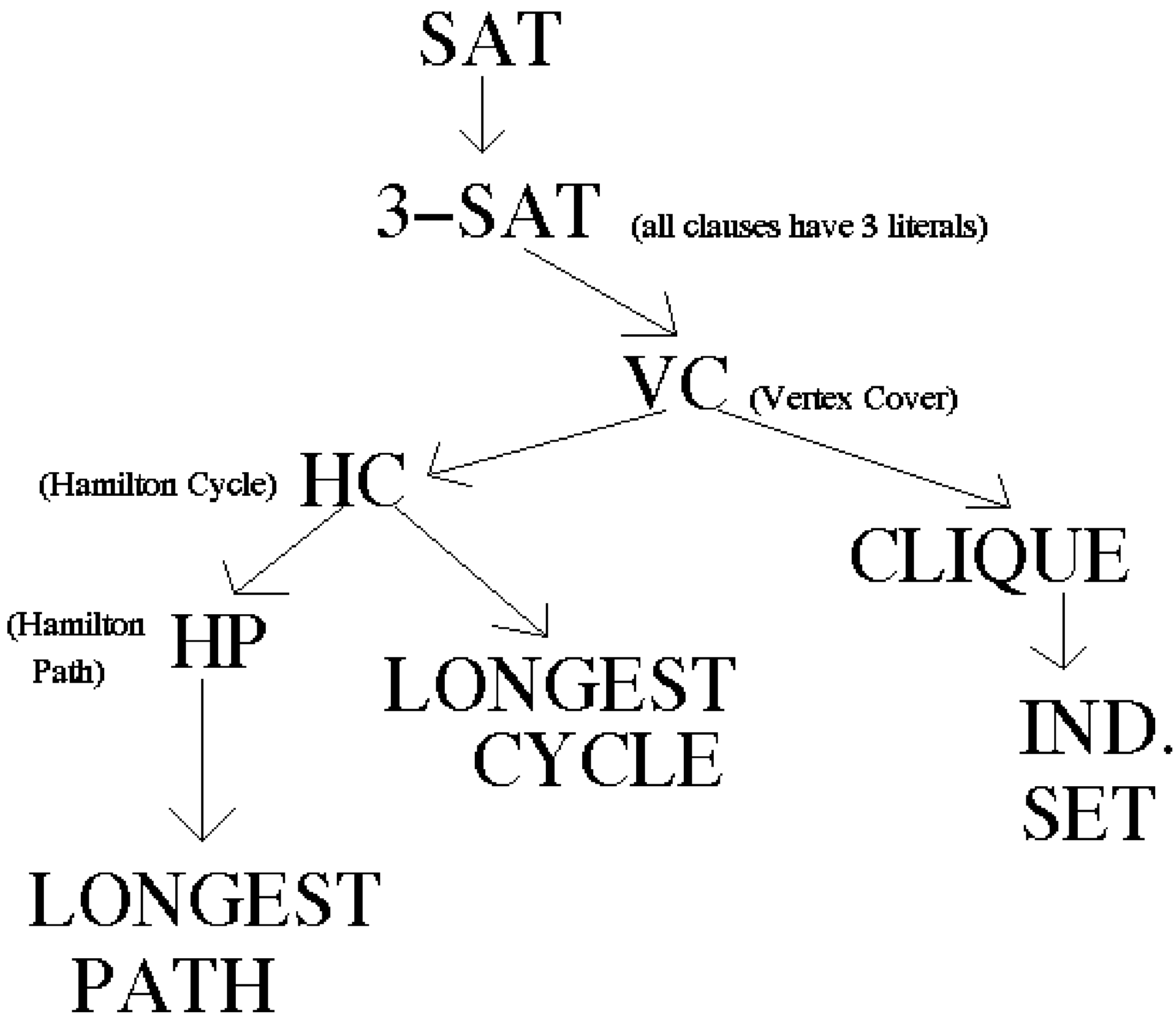


History of NP-completeness Reductions

3-SAT- each clause must contain exactly 3 variables (assignment- at most 3).

Given: SAT is NP-complete (proof CSC 320)

Theorem: **3-SAT is NP-Complete.**

The first step in any NP-completeness proof is to argue that the problem is in NP.

The problem 3-SAT is a yes/no question.

Certificate: truth assignment, can be checked in polynomial time.

Next, we show that a polynomial time algorithm for 3-SAT implies the existence of one for SAT.

To convert a SAT problem to 3-SAT:

1. Clauses of size 1.

SAT: $\{z\}$

3-SAT:

$$\{z, \quad y_1, \quad y_2\},$$
$$\{z, \neg y_1, \quad y_2\},$$
$$\{z, \quad y_1, \neg y_2\},$$
$$\{z, \neg y_1, \neg y_2\}$$

y_1 and y_2 are new variables.

2. Clauses of size 2.

SAT: $\{z_1, z_2\}$

3-SAT: $\{z_1, z_2, y\}$,
$\{z_1, z_2, \neg y\}$

y is a new variable.

3. Clauses of size 3.

Leave these as they are since they are already acceptable for 3-SAT.

4. Clauses of size 4 or more.

SAT: $\{z_1, z_2, z_3, \ldots z_k\}$, $k>3$

3-SAT:

$\{ z_1, z_2, y_1\}$,
$\{\neg y_1, z_3, y_2\}$,
$\{\neg y_2, z_4, y_3\}$,
...
$\{\neg y_{k-4}, z_{k-2}, y_{k-3}\}$,
$\{\neg y_{k-3}, z_{k-1}, z_k\}$

$y_1, y_2, \ldots y_{k-3}$, are new variables.

This does not constitute a proof of NP-completeness unless we can argue that the size of the new 3-SAT problem problem is polynomially bounded by the size of the old SAT problem. Consider each case:

Size of clause	# new literals	size before	size after
1	2	1	12
2	1	2	6
3	0	3	3
$k \geq 4$	k-3	k	k + 2(k-3)

In all cases, the size after is at most 12 times the original problem size.

2-SAT: All clauses have at most 2 literals.

There is a linear time algorithm for 2-SAT

so 2-SAT is in P.

The 3-SAT problem is as hard as SAT but unless P=NP, 2-SAT is easier than 3-SAT or SAT.

A set $S \subseteq V(G)$ is a vertex cover if every edge of G has at least one vertex in S.

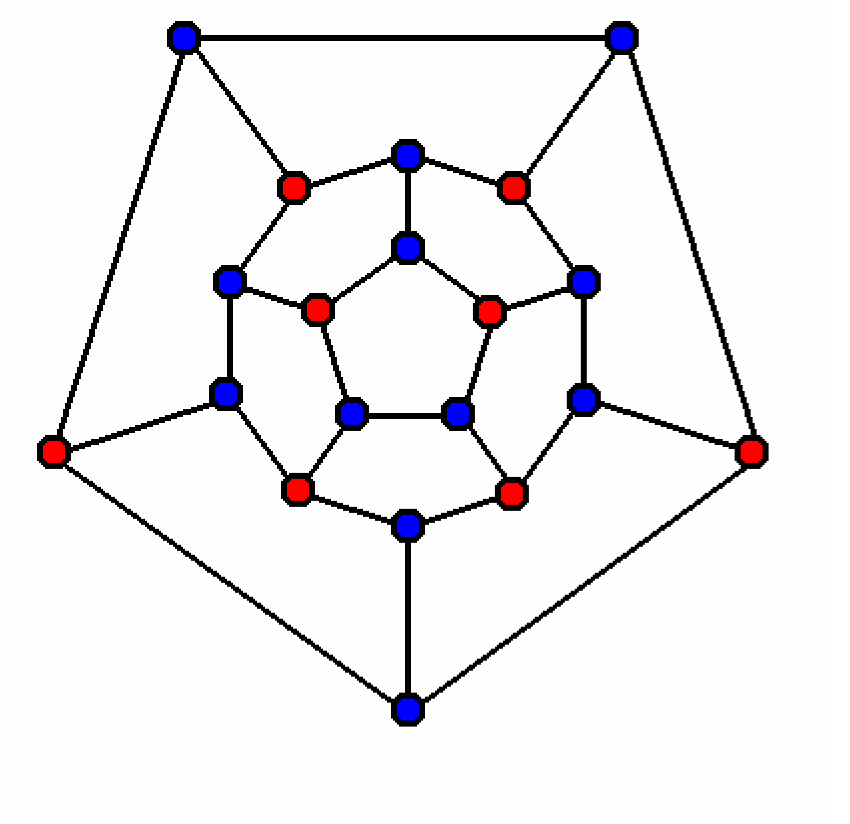
VERTEX COVER:

Given: G, k

Question: Does G have a vertex cover of order k?

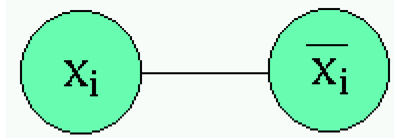
Blue: vertex cover

Red: independent set



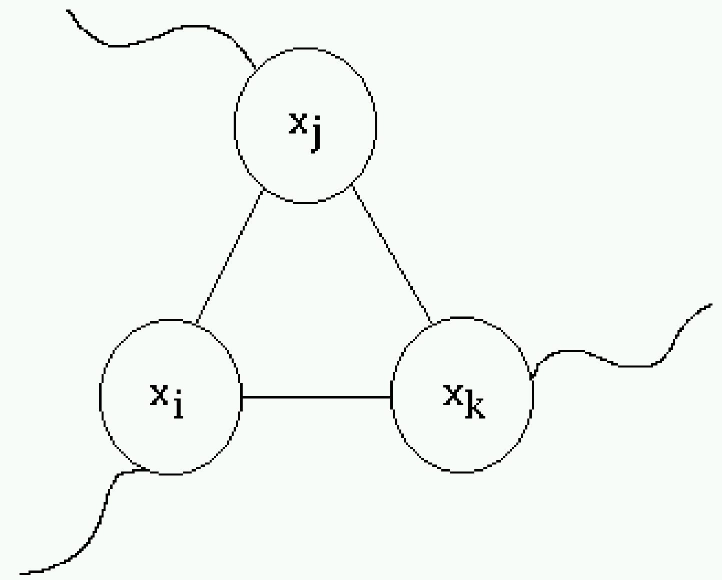
To solve 3-SAT using vertex cover:

1. For each literal x_i, include:



2. For each clause (x_i, x_j, x_k) use a gadget:

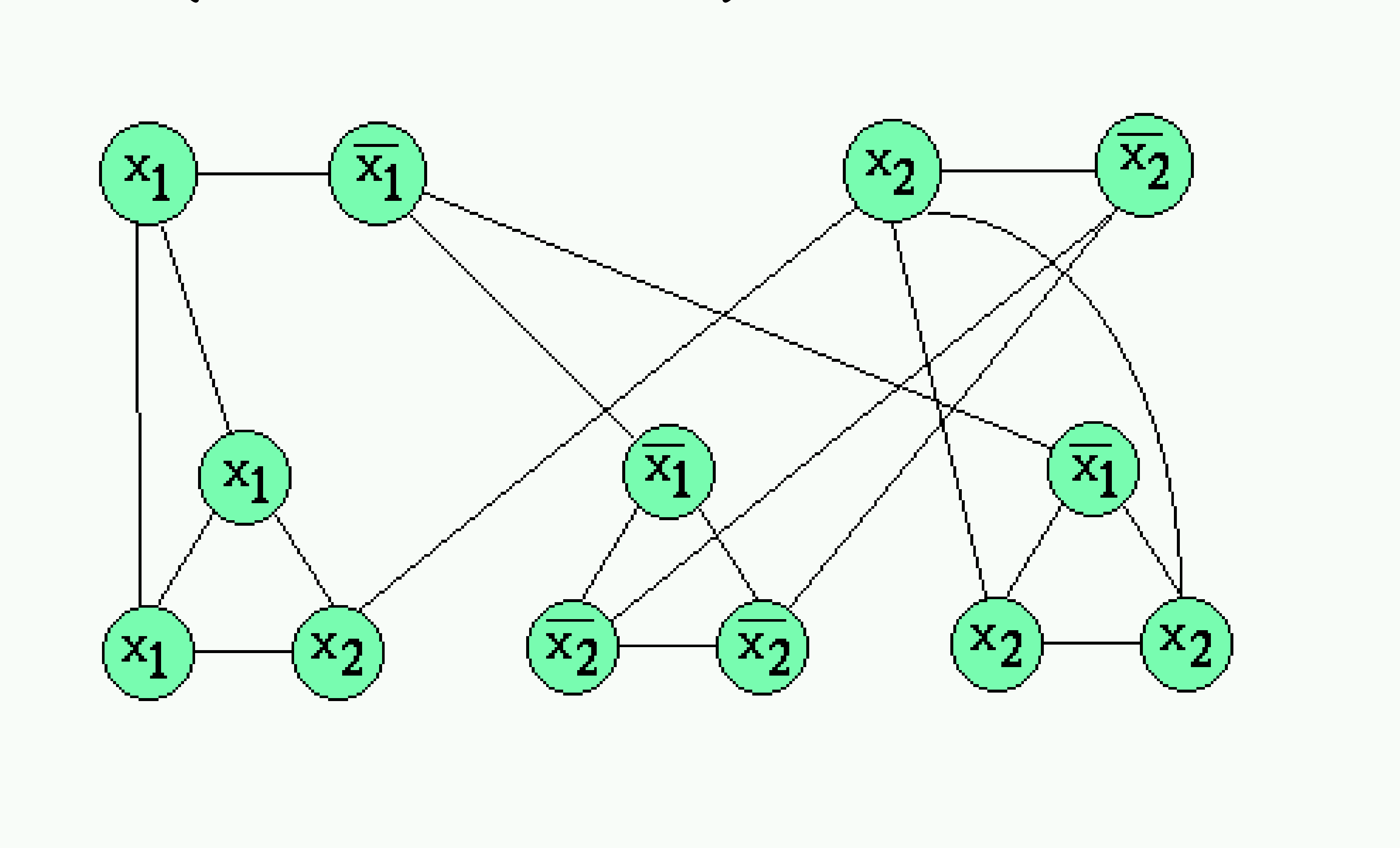
Each white vertex connects to the corresponding green one.



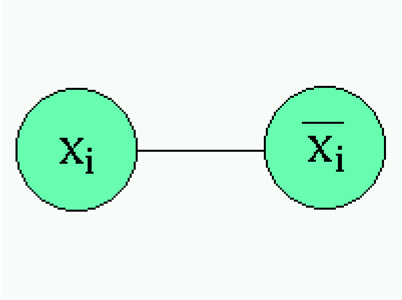
Pictures from:
http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2001/CW/npproof.html

3-SAT Problem:

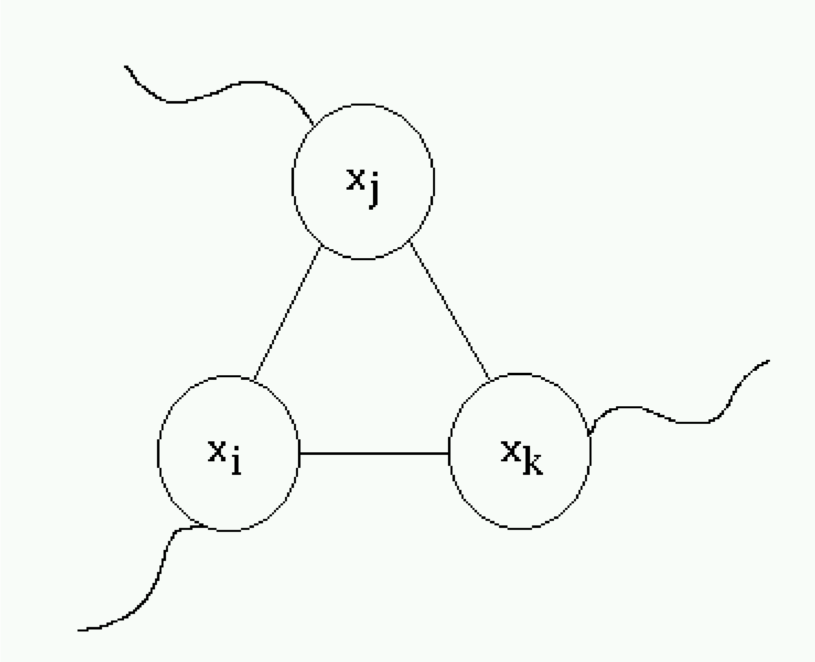
(x1 or x1 or x2) AND (¬x1 or ¬ x2 or ¬ x2) AND (¬x1 or x2 or x2)



At least one vertex from is in the vertex cover.



For each gadget, at least 2 vertices are in the vertex cover:

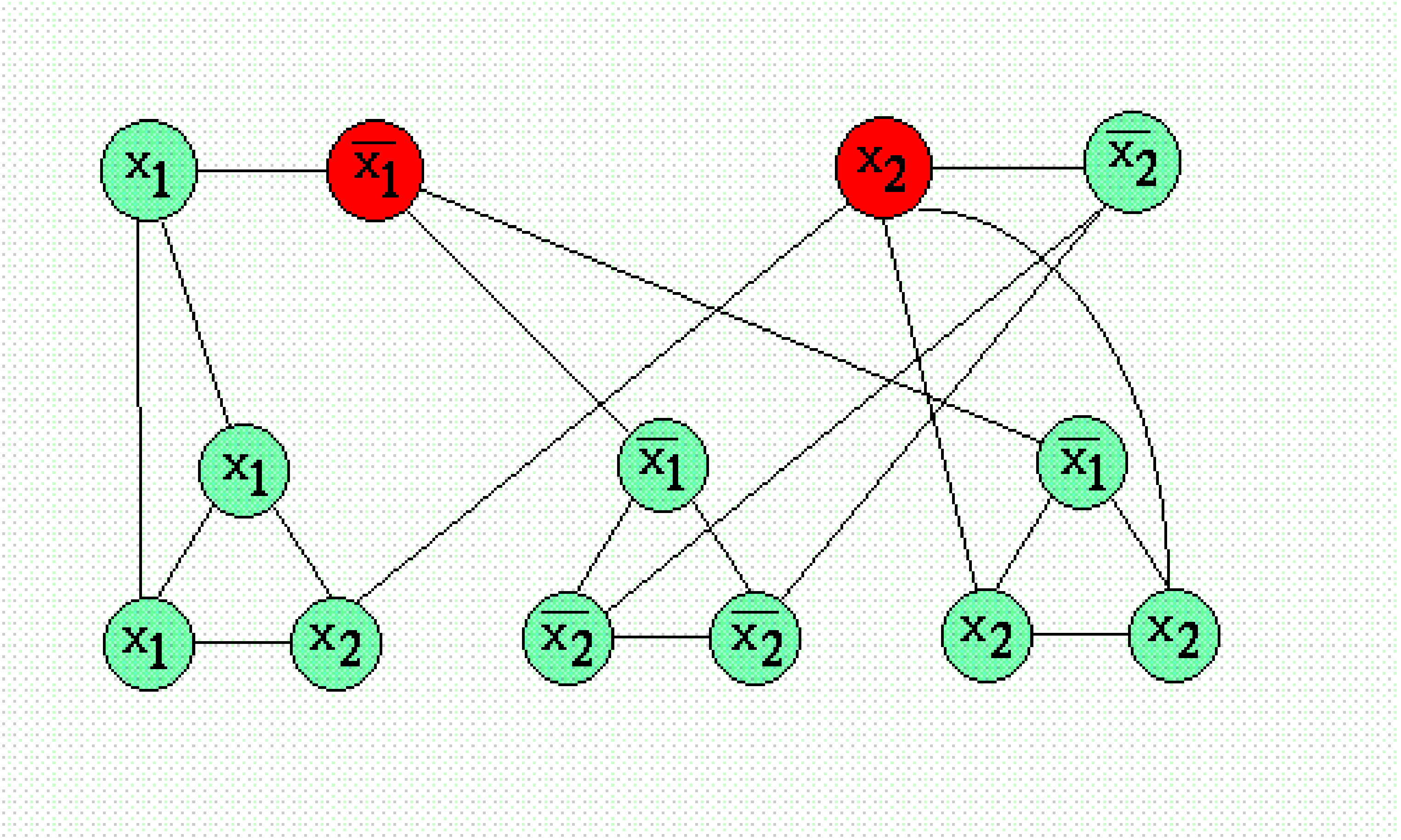


Number of variables: n

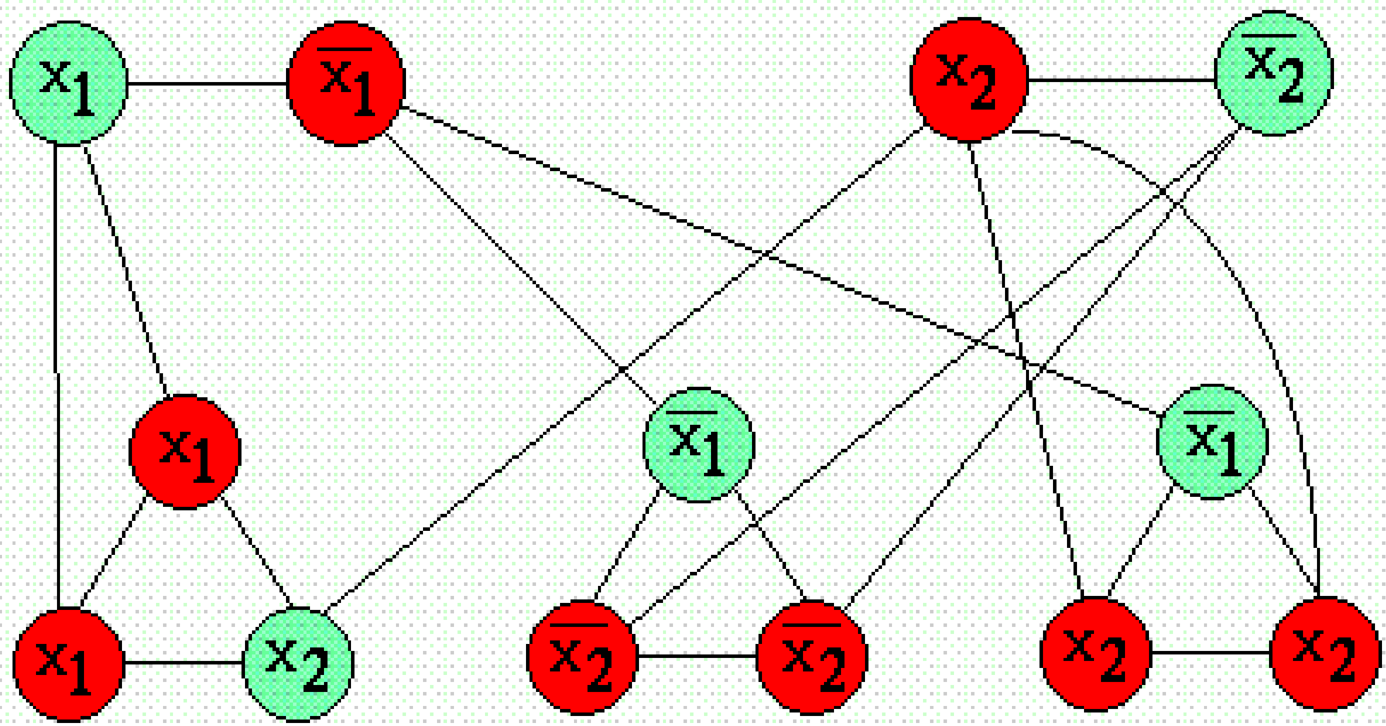
Number of clauses: m

When is there a vertex cover of order n + 2m?

Put vertices corresponding to true variables in the vertex cover.



Satisfying assignment: Each clause has at least one true variable. Put two other vertices into the vertex cover:



So each truth assignment corresponds to a vertex cover of order n + 2m.

Any vertex cover of order n + 2m corresponds to a satisfying assignment because we can only select at most one of x and ¬x (these are the true variables). The true variables must satisfy each clause since at most 2 vertices can be selected from each clause gadget.