NP-complete Languages

Polynomial time reducible

Definition:

Language A
is polynomial time reducible to
language B

if there is a polynomial computable function f such that:

$$w \in A \Leftrightarrow f(w) \in B$$

Theorem:

Suppose that A is polynomial reducible to B.

If $B \in P$ then $A \in P$.

Theorem: 3CNF-SAT is polynomial time reducible to CLIQUE

Proof: give a polynomial time reduction of one problem to the other

Transform formula to graph

Example of a polynomial-time reduction:

We will reduce the

3CNF-satisfiability problem

to the

CLIQUE problem

3CNF formula:

literal variable or its complement

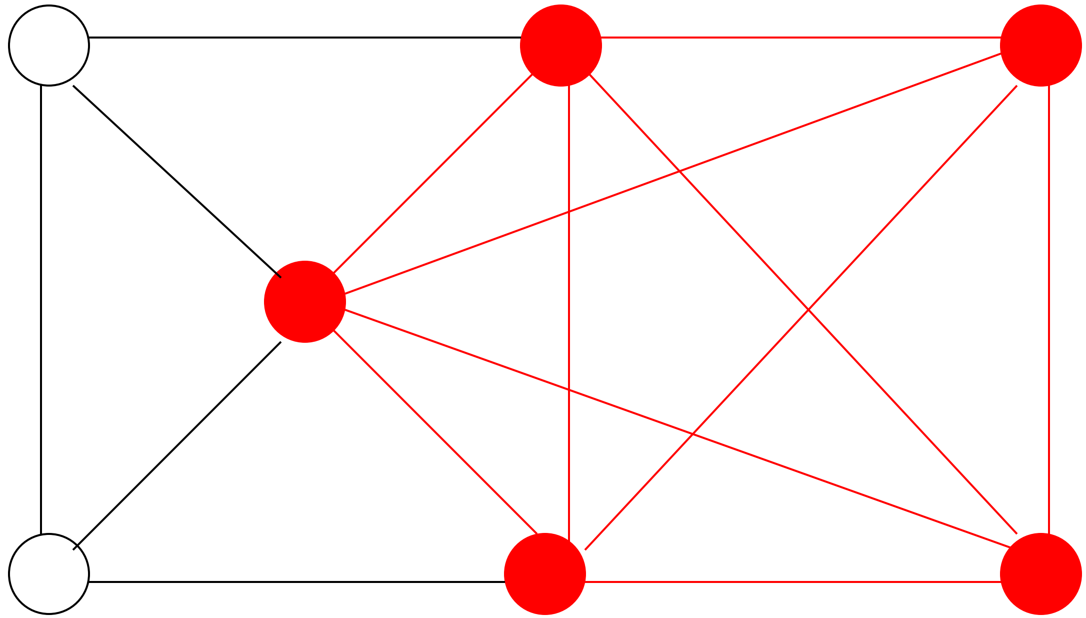
$$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge \underbrace{(x_3 \vee \overline{x_5} \vee x_6)}_{\text{clause}} \wedge (x_3 \vee \overline{x_6} \vee x_4) \wedge (x_4 \vee x_5 \vee x_6)$$

Each clause has three literals

Language:

3CNF-SAT $= \{ w : w$ is a satisfiable 3CNF formula$\}$

A 5-clique in graph G



Language:

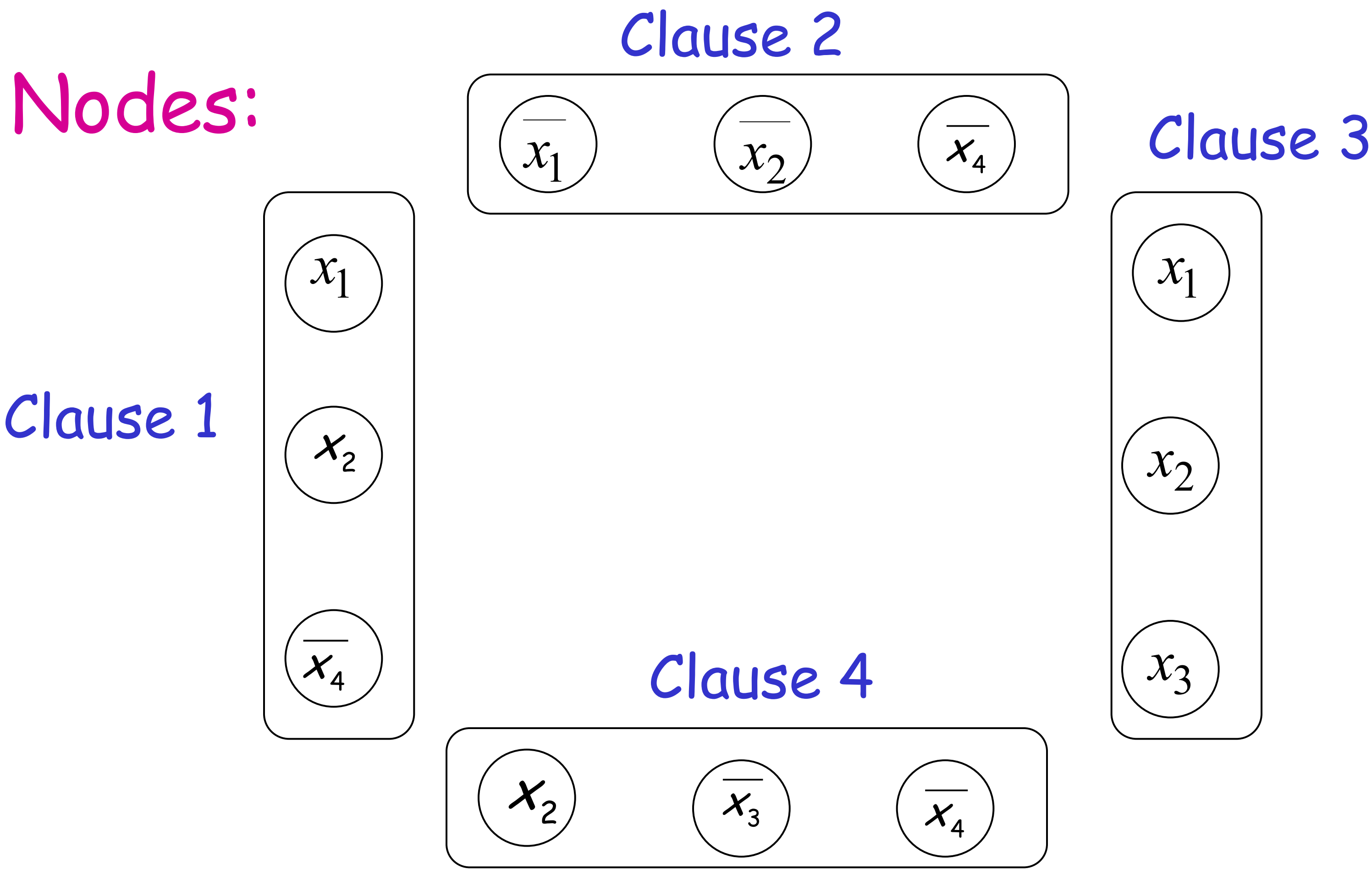
CLIQUE = $\{\langle G, k \rangle$: graph G contains a k-clique$\}$

Transform formula to graph.

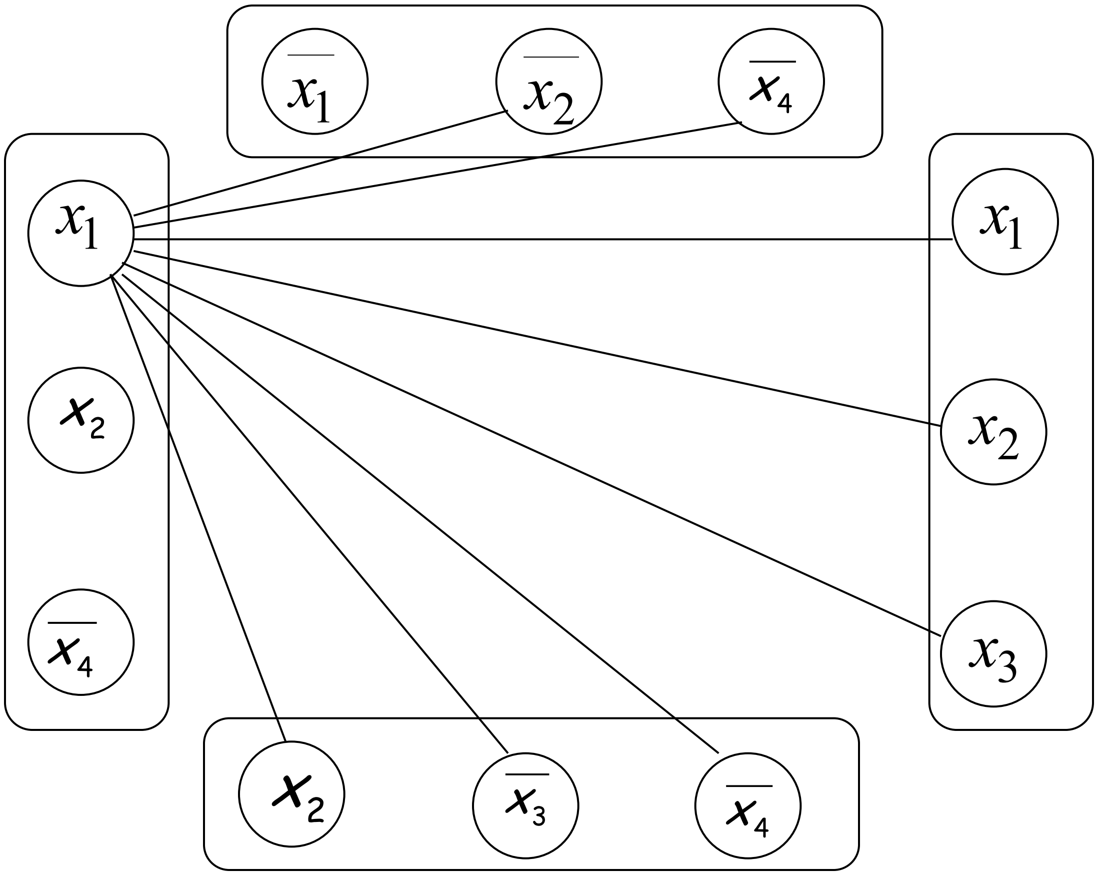
Example:

$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$$

Create Nodes:

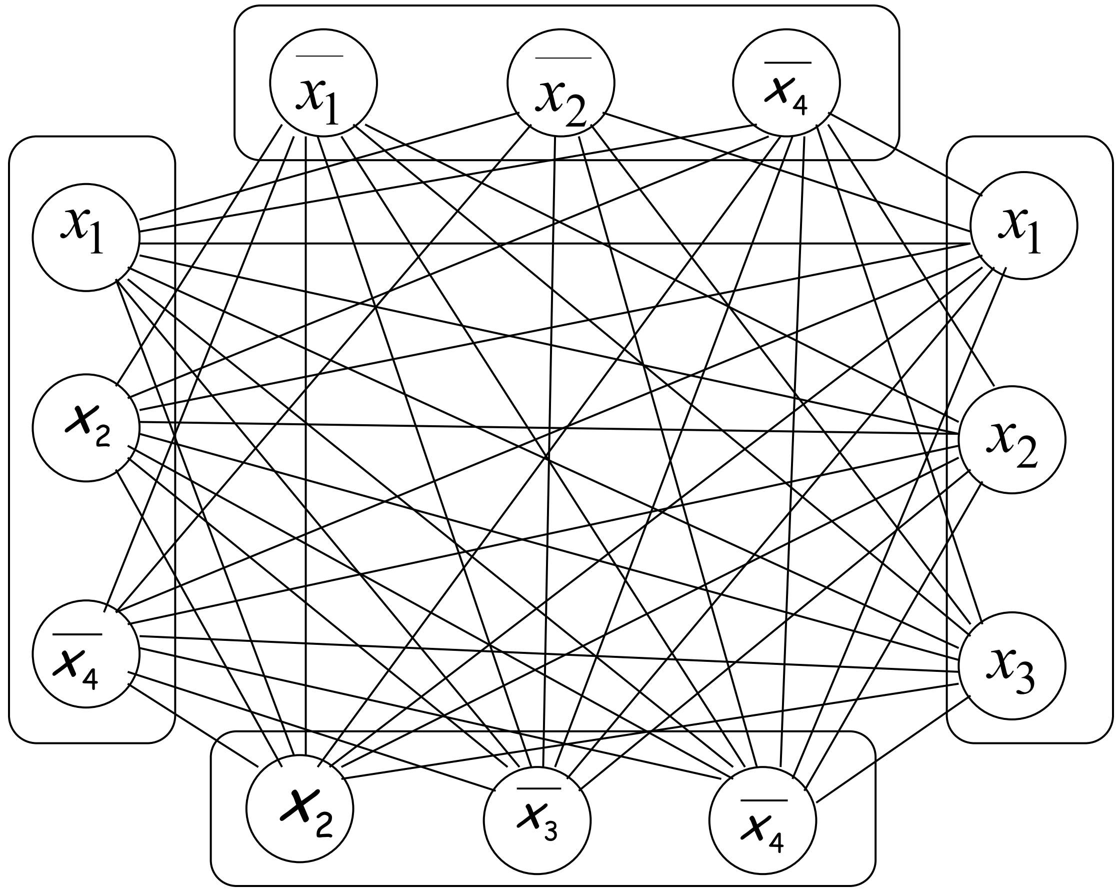


$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$$



Add link from a literal ξ to a literal in every other clause, except the complement $\overline{\xi}$

$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4})$$



Resulting Graph

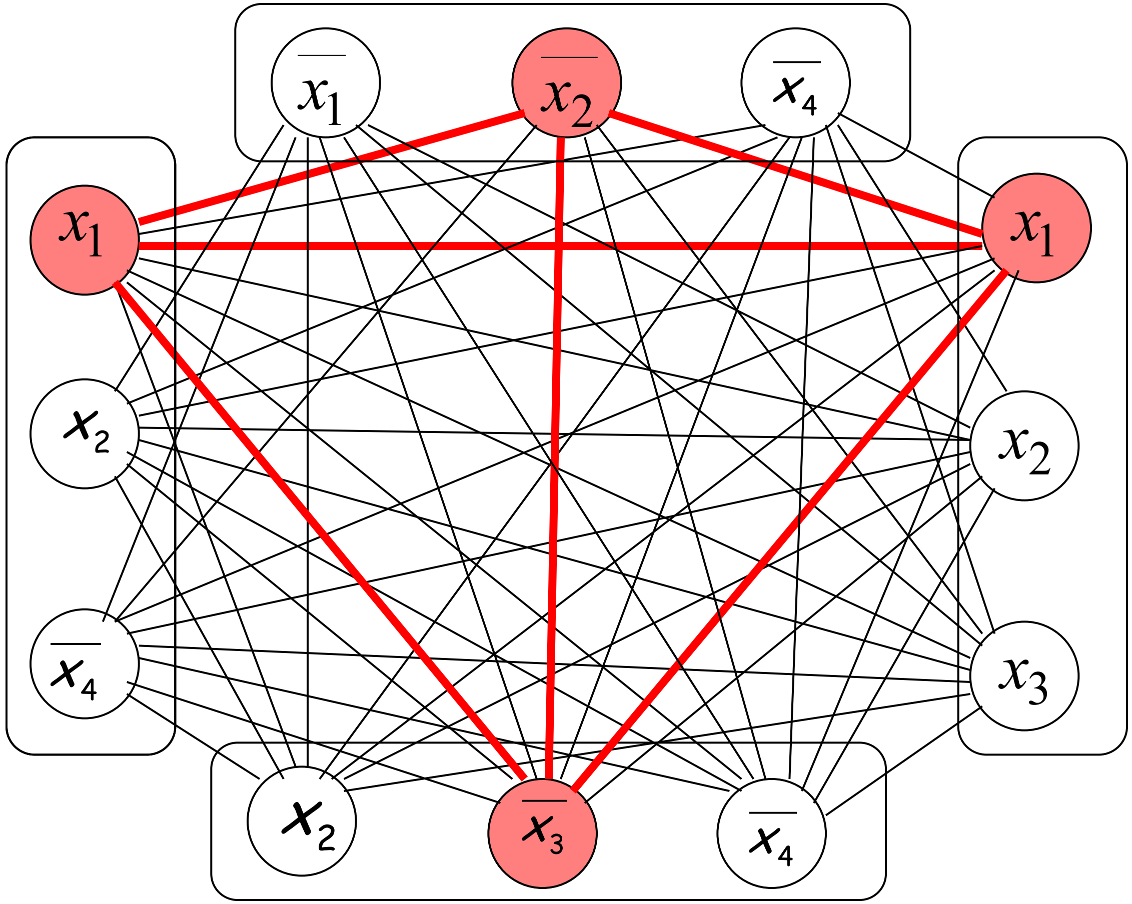
$$(x_1 \vee x_2 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee \overline{x_4}) = 1$$

$x_1 = 1$

$x_2 = 0$

$x_3 = 0$

$x_4 = 1$



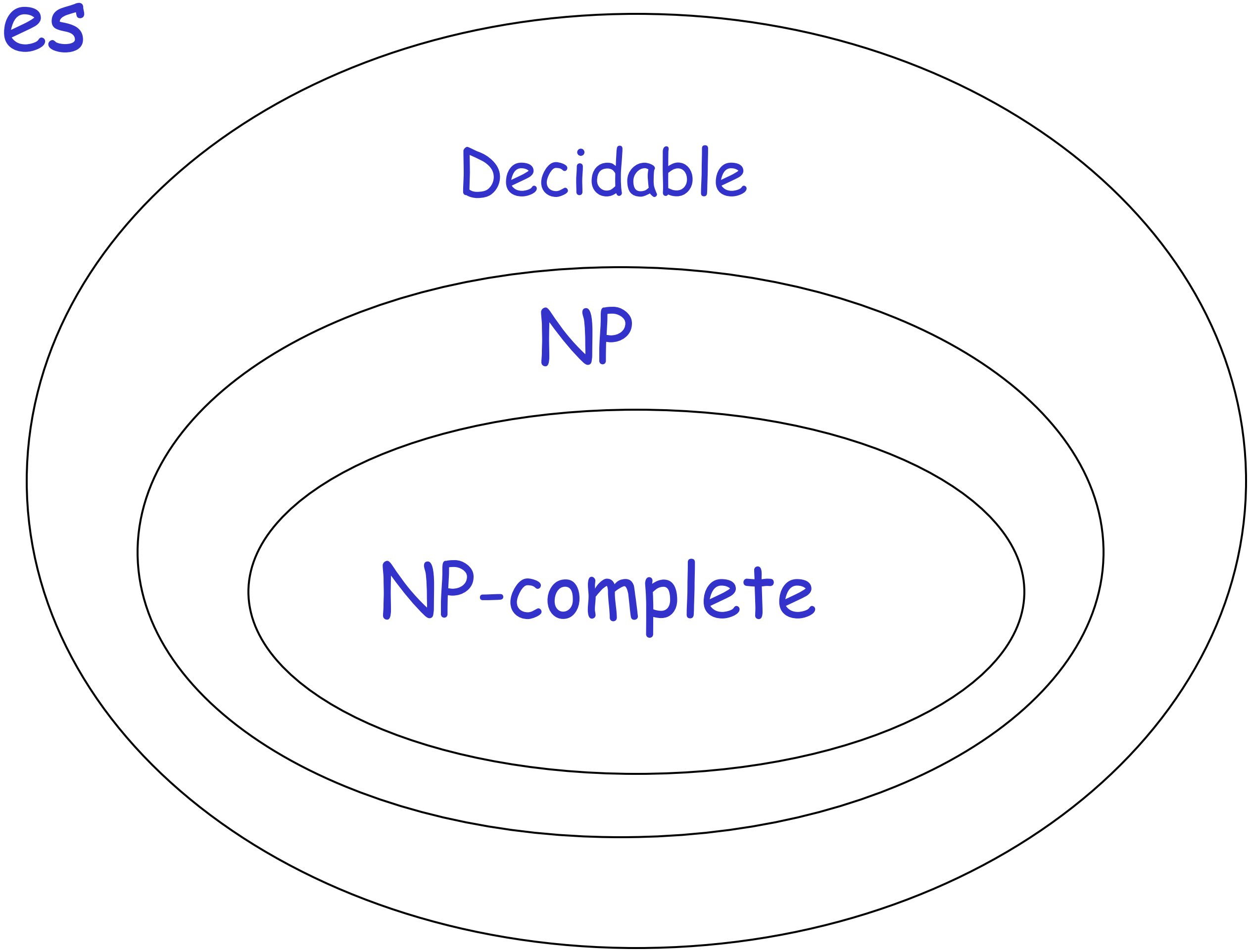
The formula is satisfied if and only if the Graph has a 4-clique

End of Proof

NP-complete Languages

NP-complete Languages

We define the class of NP-complete languages



A language L is NP-complete if:

- L is in NP, and
- Any language in NP is reduced to L in polynomial time

An NP-complete Language

Cook-Levin Theorem:

Language SAT (satisfiability problem)
is NP-complete

Theorem:

If: a. Language A is NP-complete

b. Language B is in NP

c. A is polynomial time reducible to B

Then: B is NP-complete

CLIQUE is NP-complete

Proof:

a. 3CNF-SAT is NP-complete

b. CLIQUE is in NP

c. 3CNF-SAT is polynomial reducible to CLIQUE
(shown earlier)

Apply previous theorem with

A=3CNF-SAT and B=CLIQUE

Theorem: HAMILTONIAN-PATH is NP-complete

Proof:

1. HAMILTONIAN-PATH is in NP

2. We will reduce in polynomial time 3CNF-SAT to HAMILTONIAN-PATH (NP-complete)

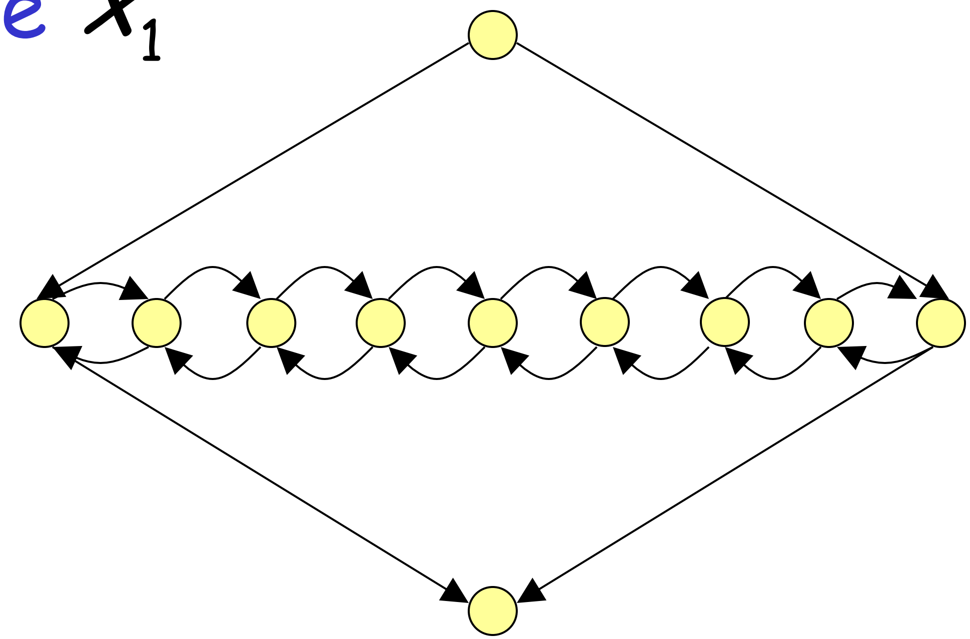
Consider an arbitrary CNF formula

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

Whatever we will do applies to 3CNF formulas as well

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

Gadget for variable x_1



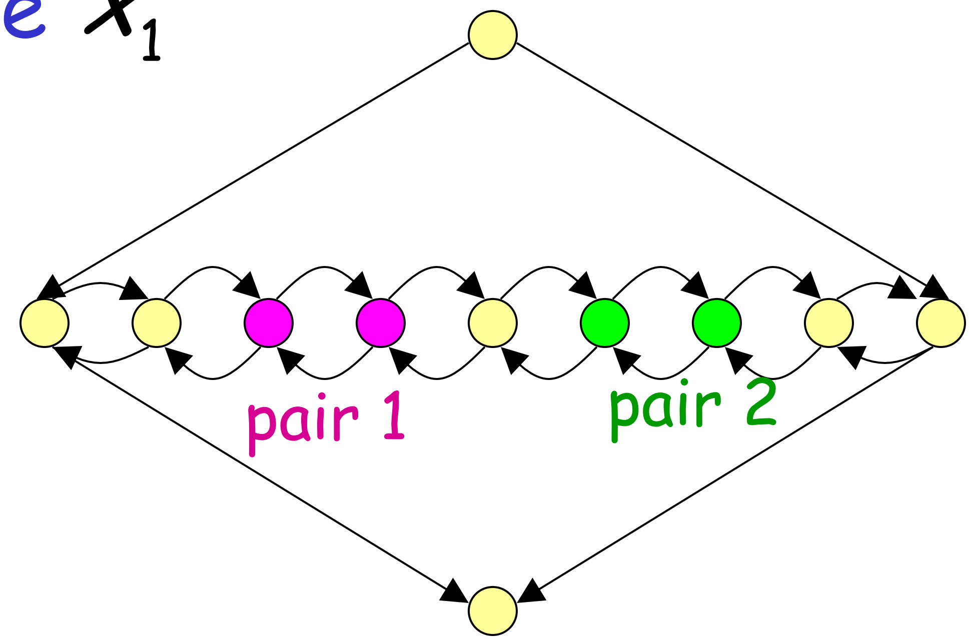
Number of nodes in row: $3l+3$

for l clauses

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

Gadget for variable x_1

clause node $(x_1 \vee x_2)$

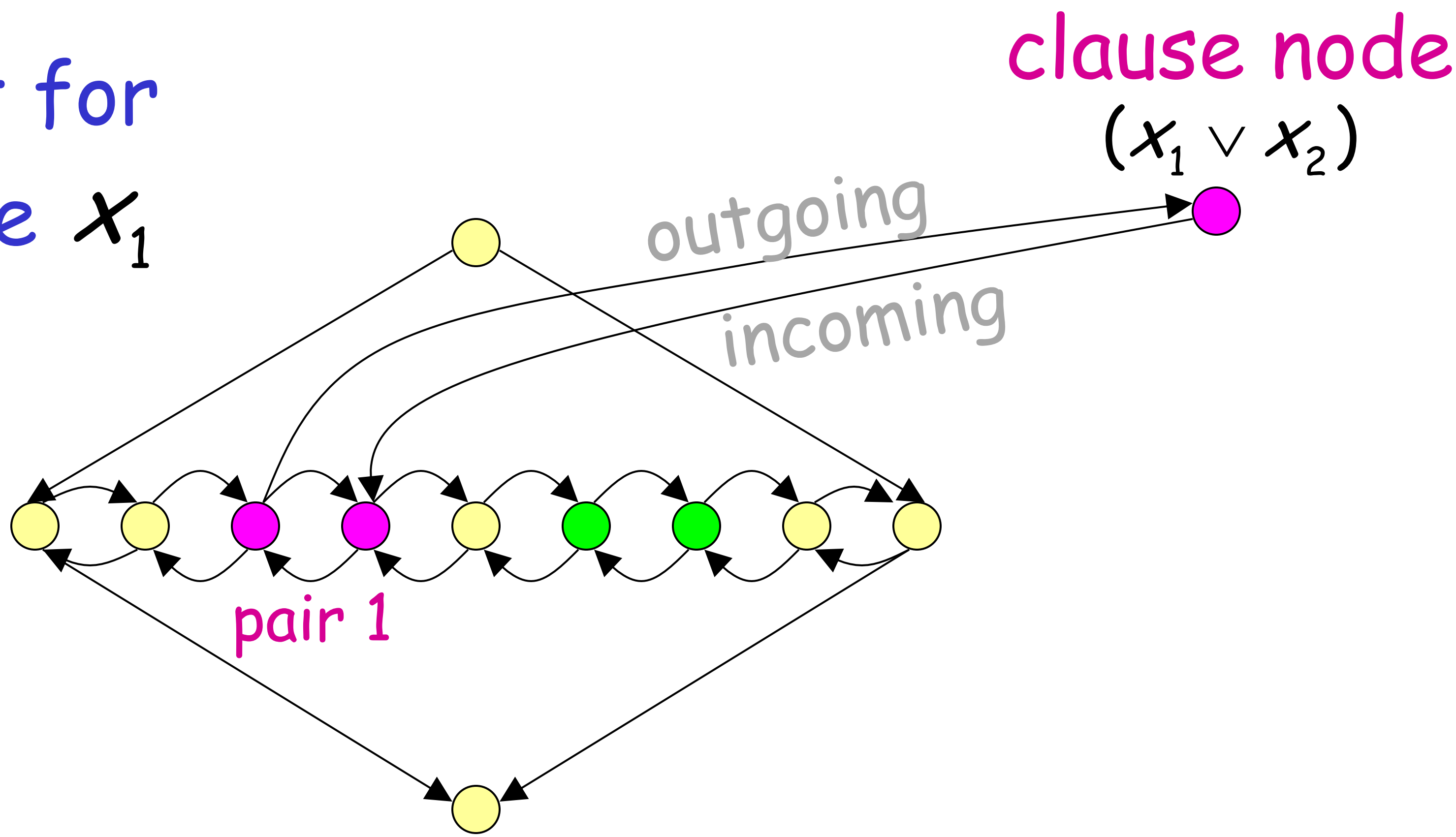


clause node $(\overline{x_1} \vee x_2)$

A pair of nodes in gadget for each clause node;
Pairs separated by a node

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

Gadget for variable x_1



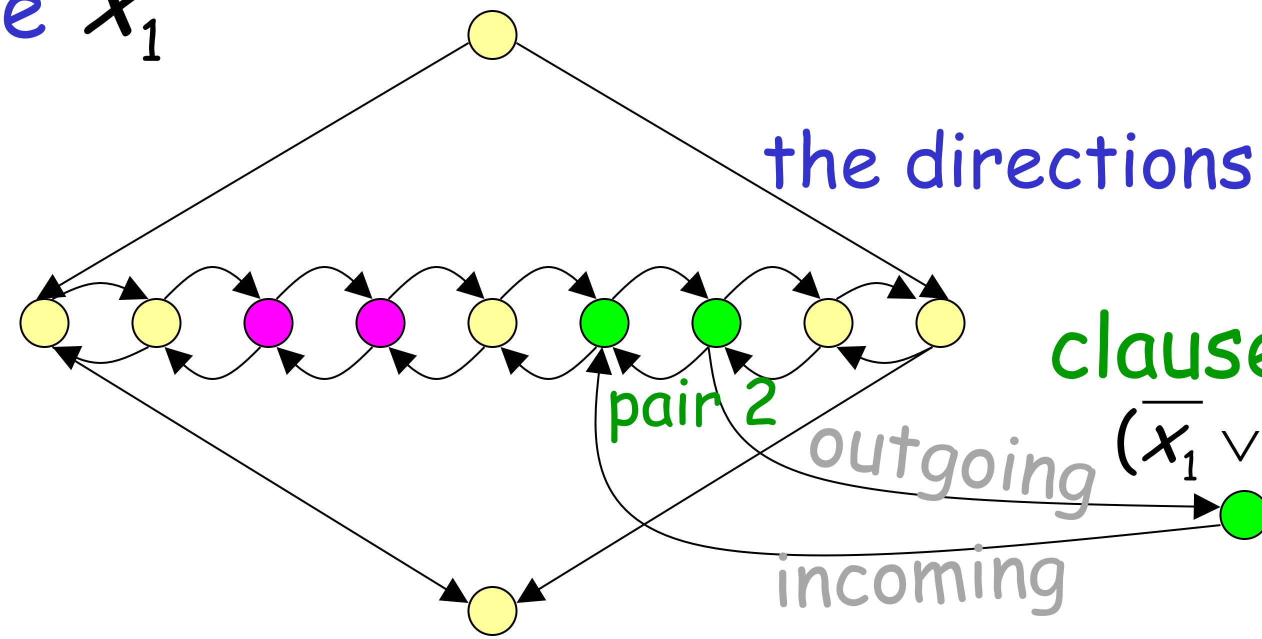
If variable appears as is in clause:

first edge from gadget outgoing (left to right)

second edge from gadget incoming

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

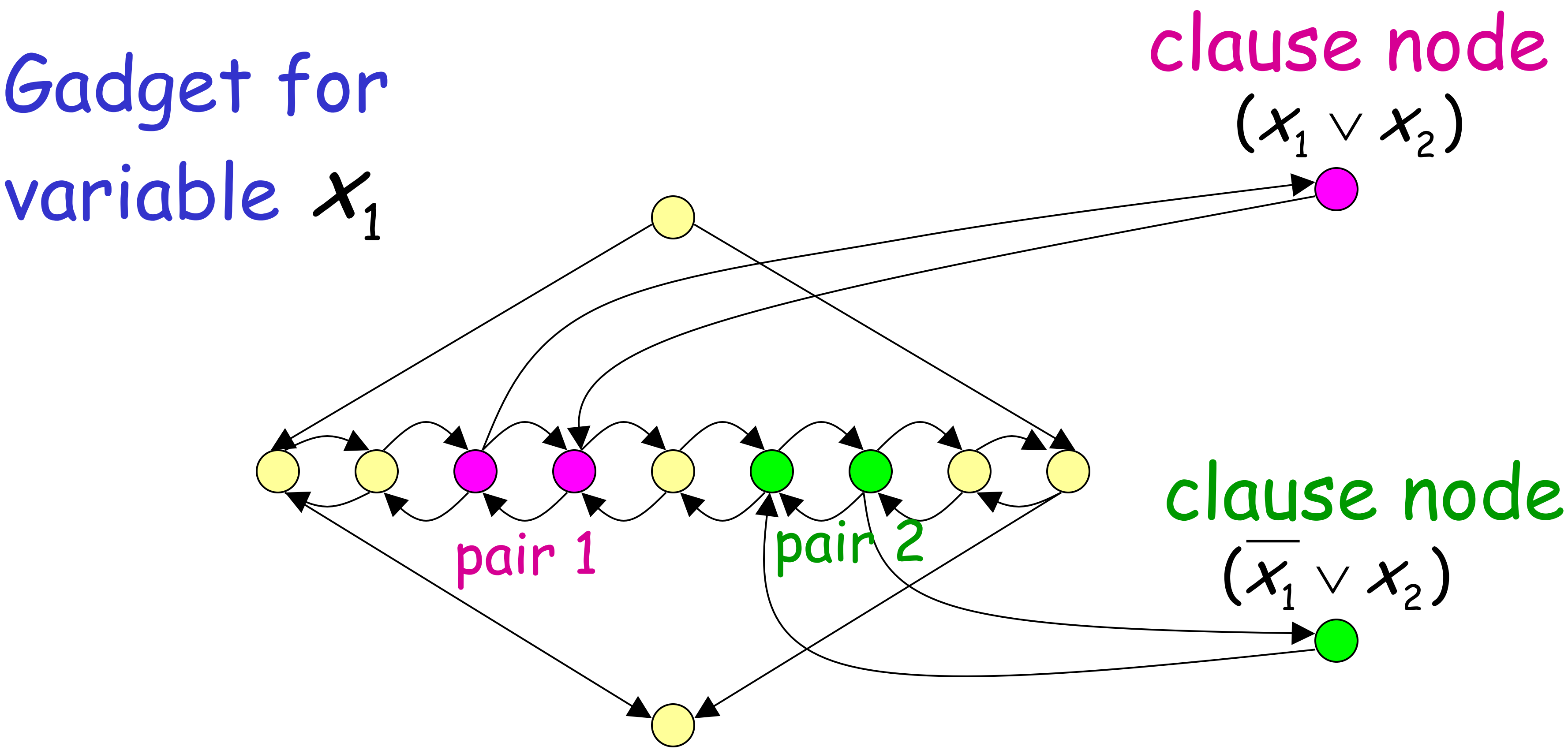
Gadget for variable x_1



If variable appears inverted in clause:

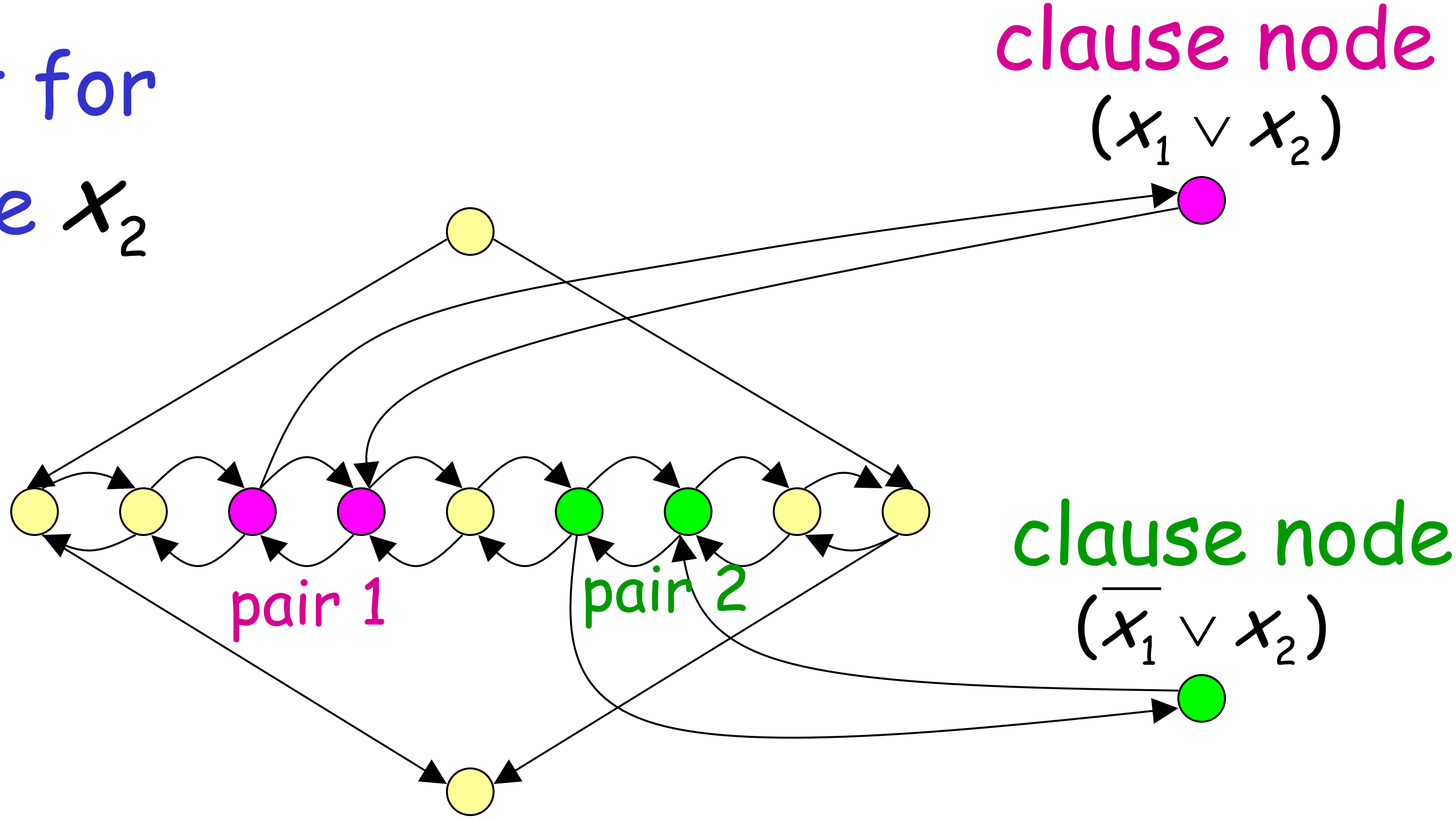
- first edge from gadget incoming (left to right)
- second edge from gadget outgoing

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$



$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

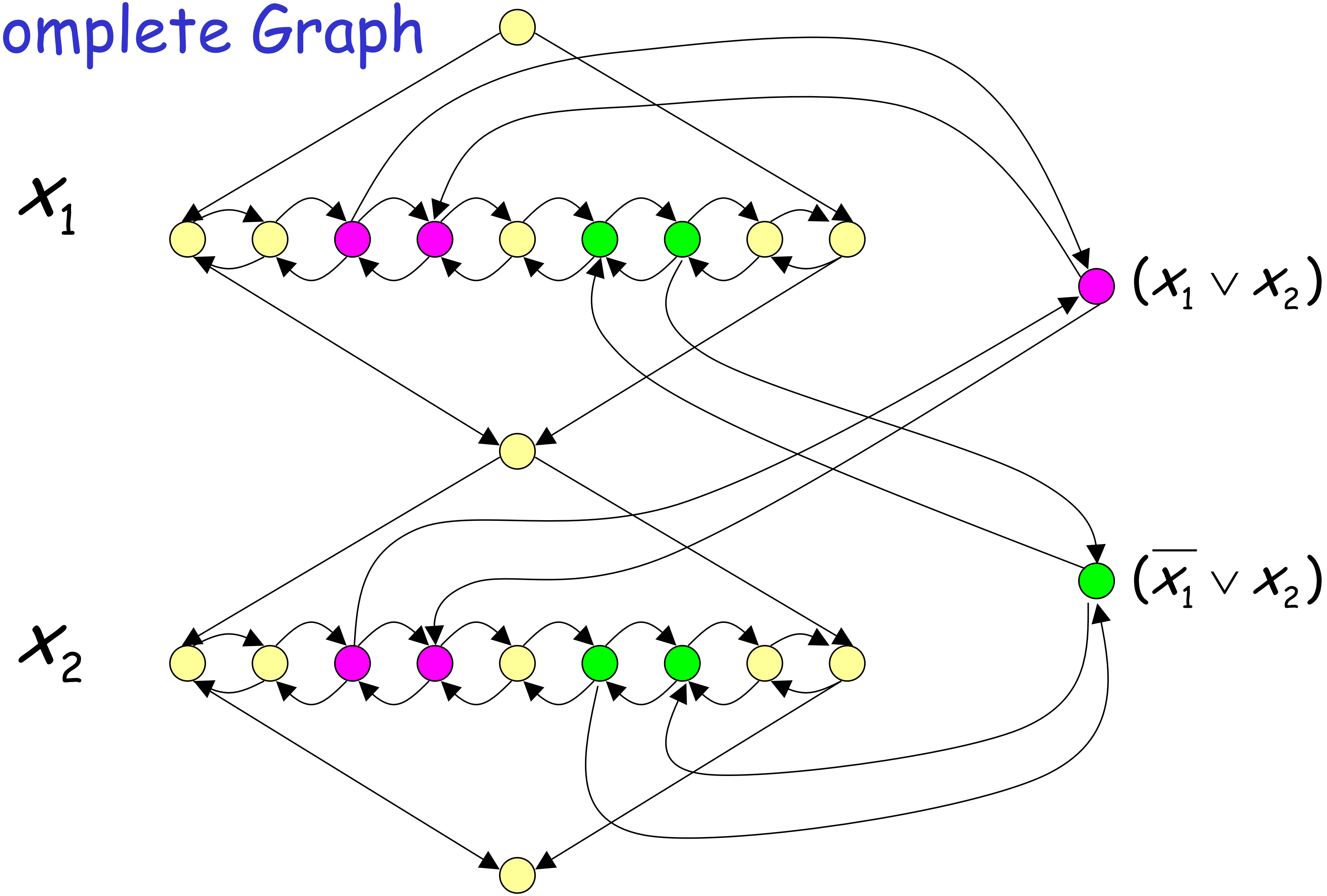
Gadget for variable x_2



Arrow directions to clause nodes are same here

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2)$$

Complete Graph

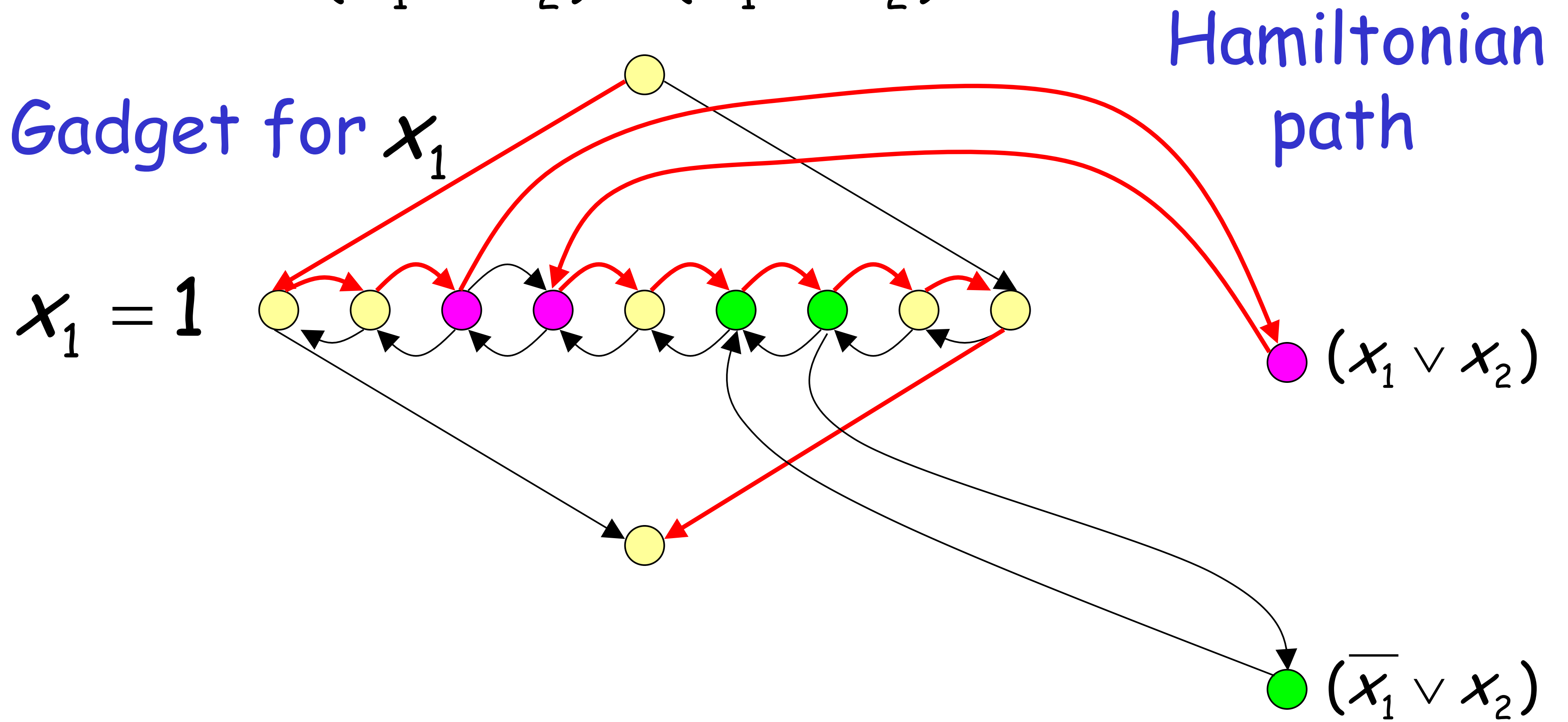


$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$

Satisfying assignment: $x_1 = 1$ $x_2 = 1$

Satisfying assignment: $x_1 = 0$ $x_2 = 1$

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$



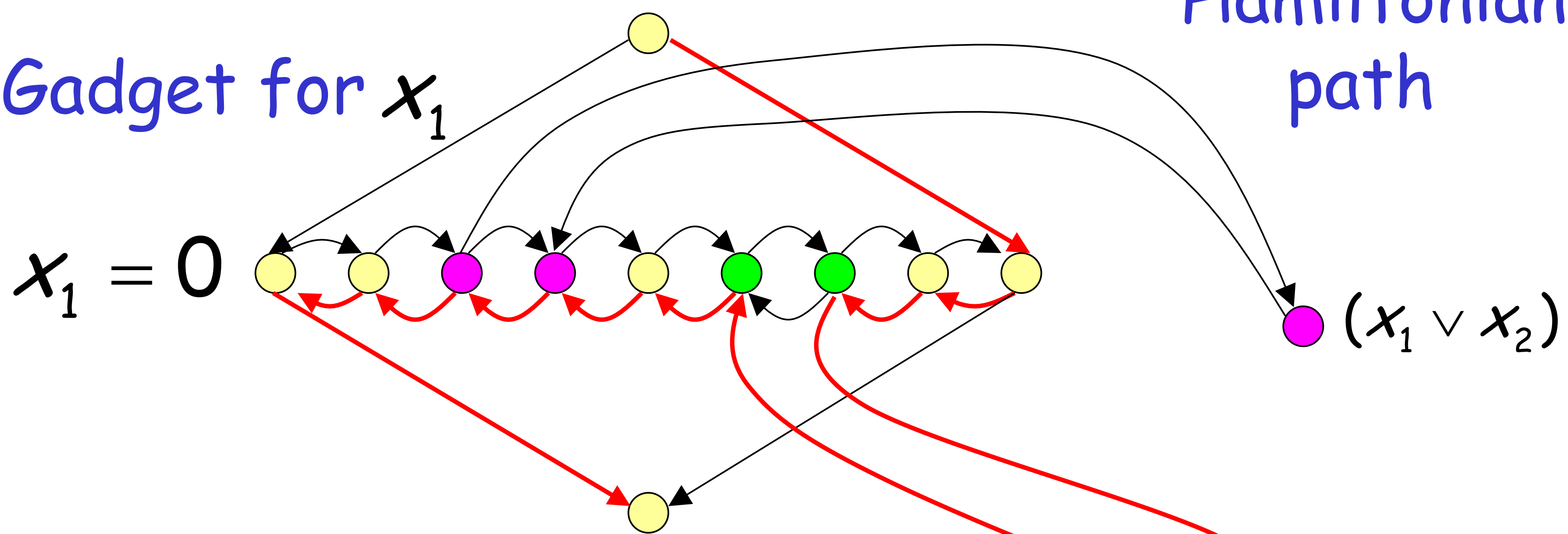
If variable is set to 1 traverse its gadget from left to right

Visit clauses satisfied from the variable assignment

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$

Hamiltonian path

Gadget for x_1



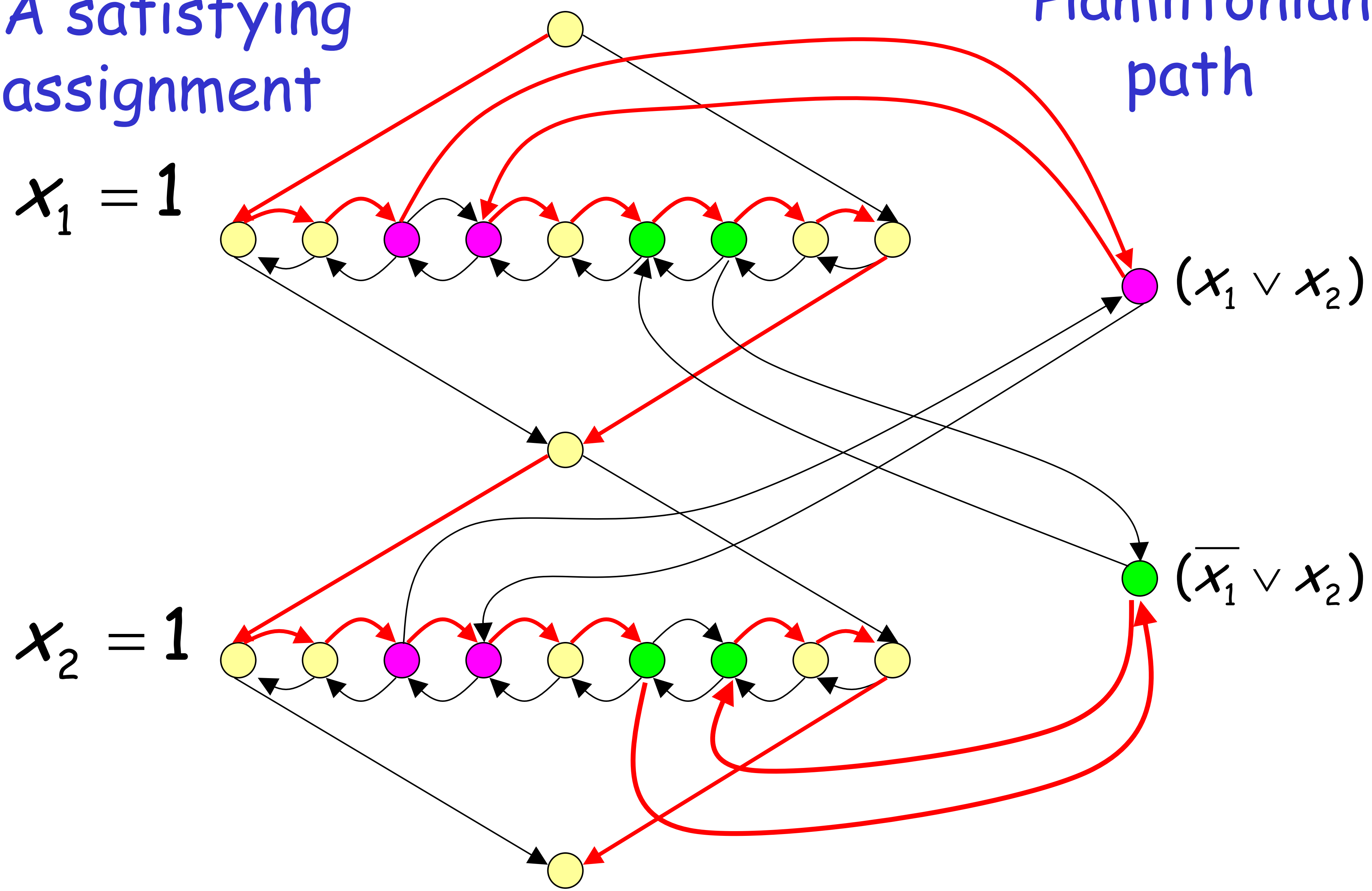
If variable is set to 0 traverse its gadget from right to left

Visit clauses satisfied from the variable assignment

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$

A satisfying assignment

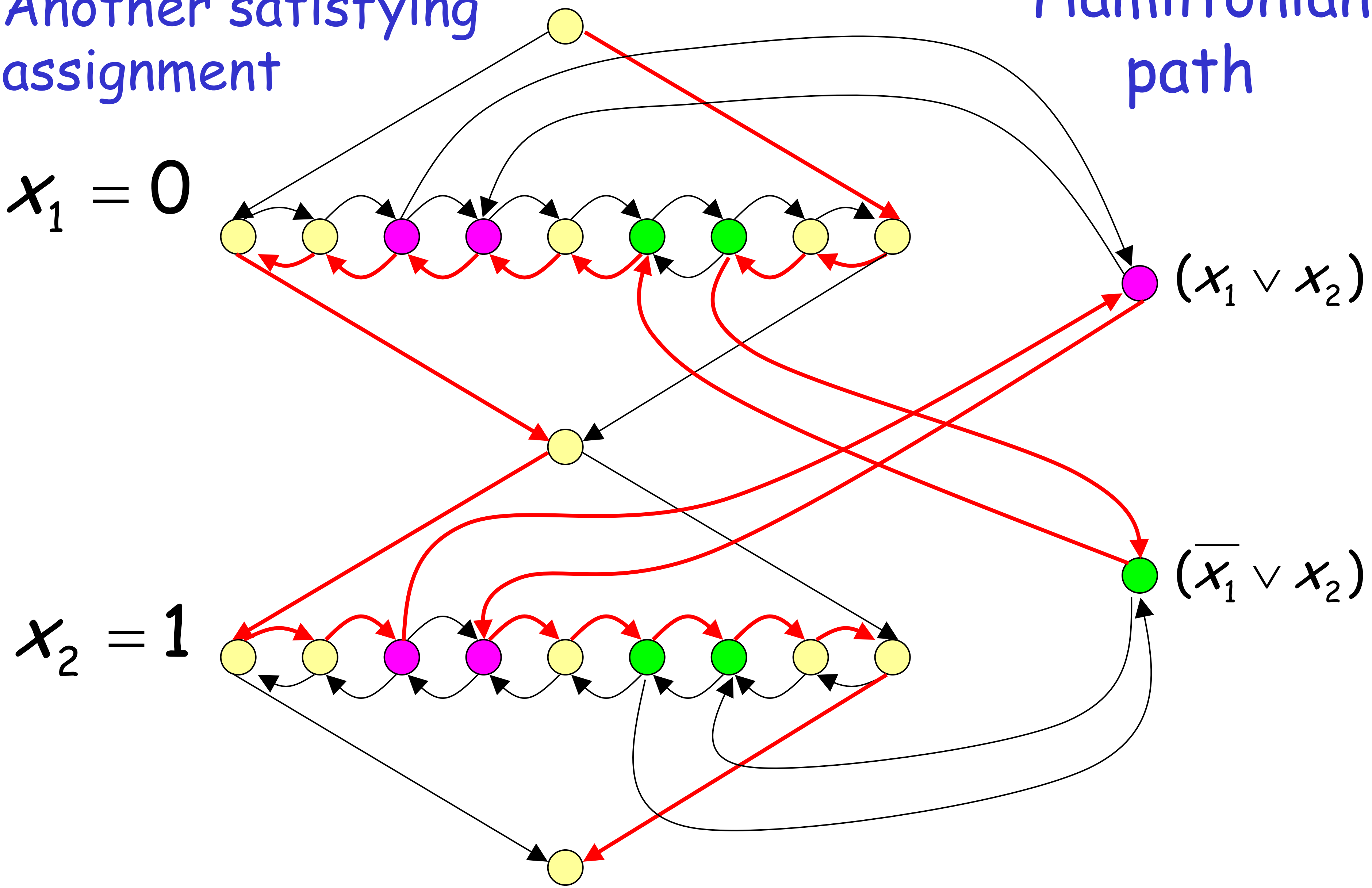
Hamiltonian path



$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$

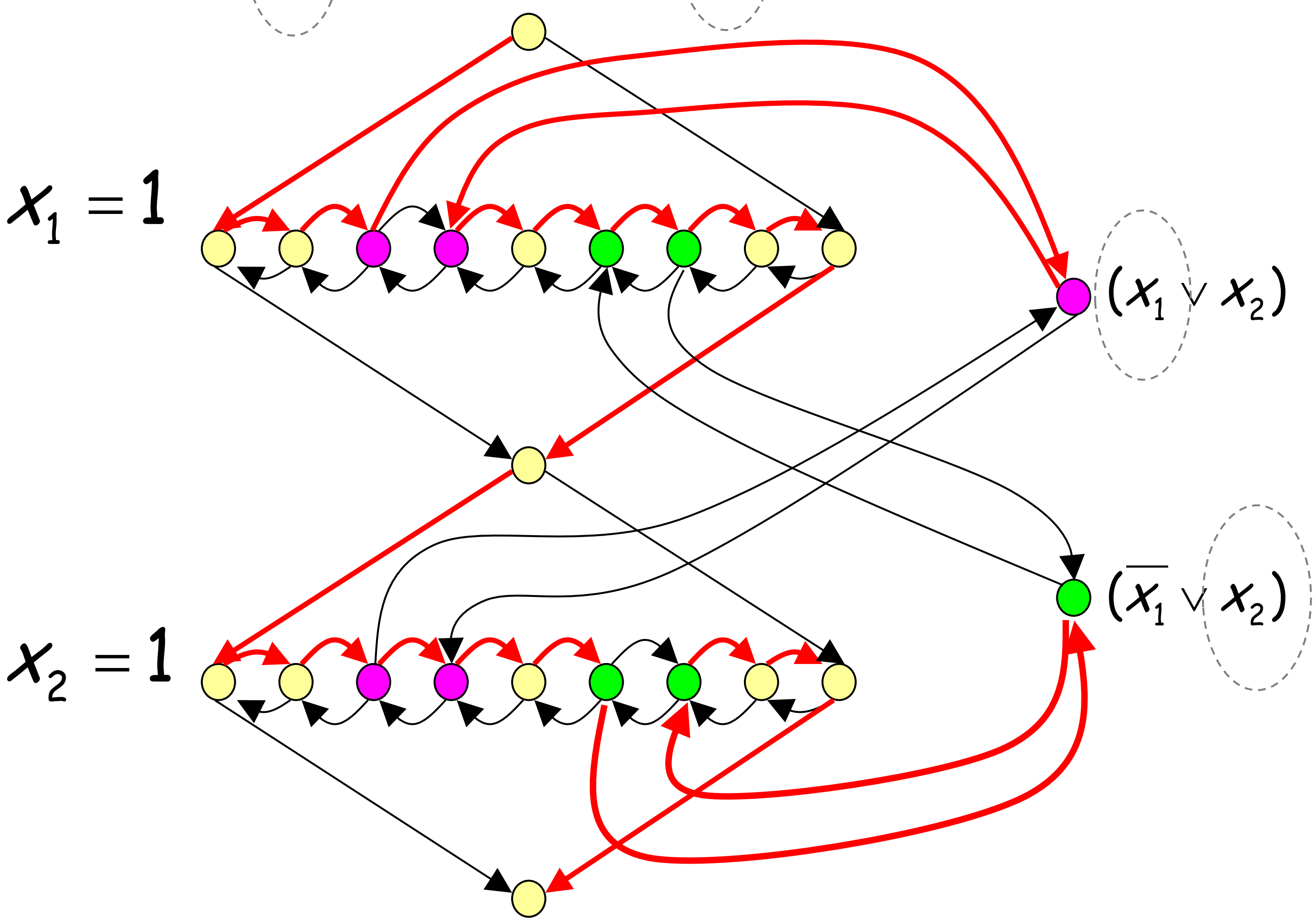
Another satisfying assignment

Hamiltonian path



For each clause pick one variable that satisfies it and visit respective node once from that variable gadget

$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) = 1$$

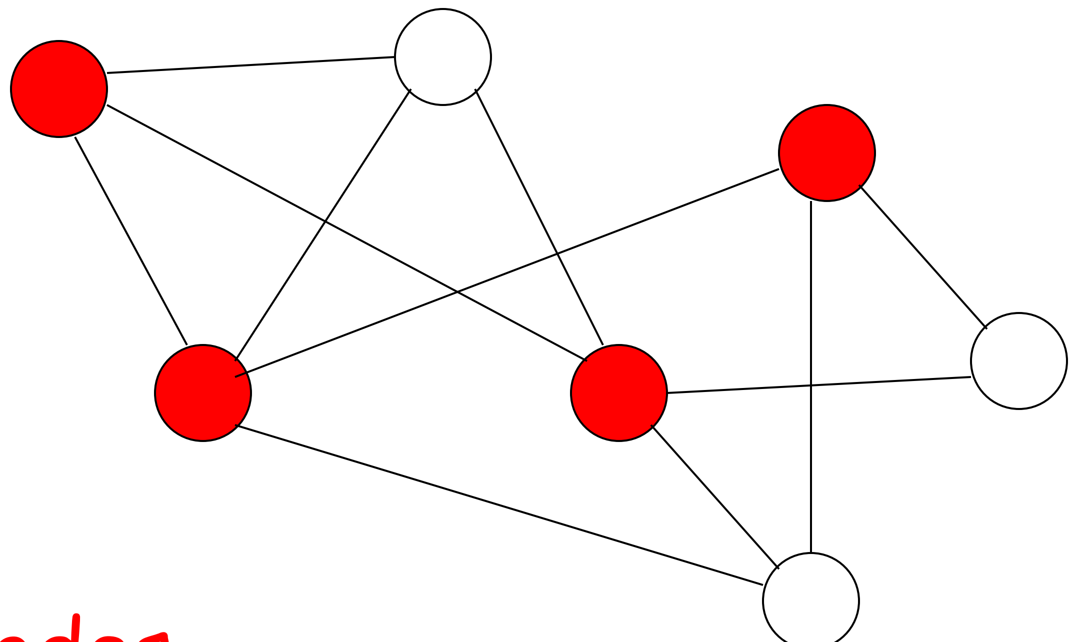


Another example:
Vertex Cover

Vertex Cover

Vertex cover of a graph
is a subset of nodes S such that every edge
in the graph touches one node in S

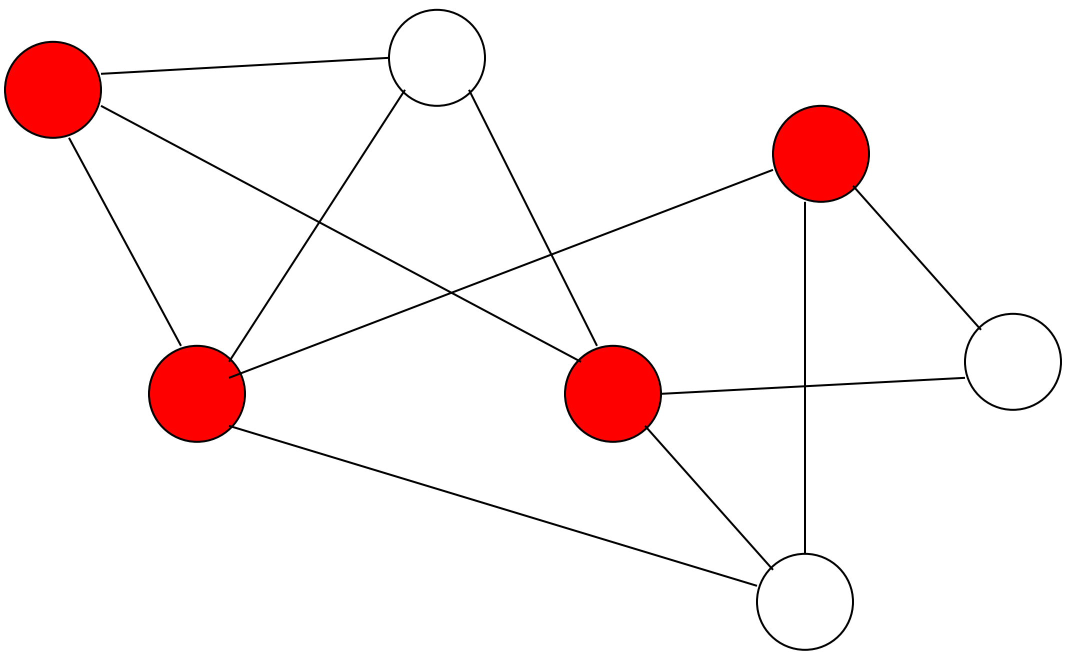
Example:



S = red nodes

Size of vertex-cover
is the number of nodes in the cover

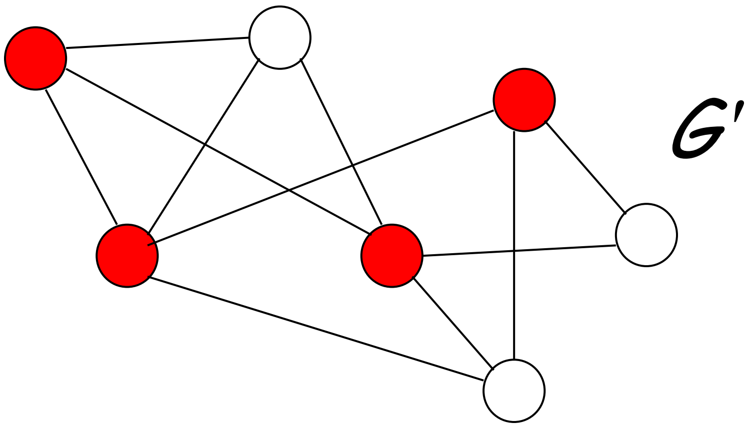
Example: $|S|=4$



Corresponding language:

$$\text{VERTEX-COVER} = \{\langle G, k \rangle : \text{graph } G \text{ contains a vertex cover of size } k\}$$

Example:



$$\langle G', 4 \rangle \in \text{VERTEX - COVER}$$

VERTEX-COVER is NP-complete

Proof:

1. VERTEX-COVER is in NP

2. We will reduce in polynomial time 3CNF-SAT to VERTEX-COVER (NP-complete)

Let φ be a 3CNF formula
with m variables
and l clauses

Example:

$$\varphi = \underbrace{(x_1 \vee x_2 \vee x_3)}_{\text{Clause 1}} \wedge \underbrace{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_4})}_{\text{Clause 2}} \wedge \underbrace{(\overline{x_1} \vee x_3 \vee x_4)}_{\text{Clause 3}}$$

$m = 4$

$l = 3$

Formula φ can be converted to a graph G such that:

φ is satisfied

if and only if

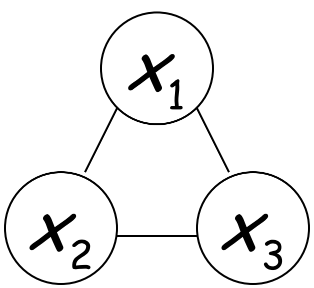
G Contains a vertex cover of size $k = m + 2l$

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$$

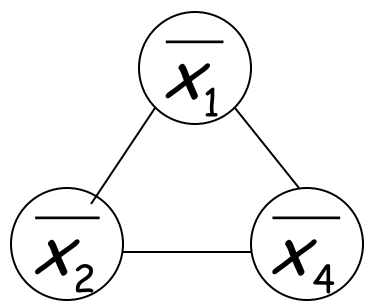
Clause 1 Clause 2 Clause 3

Variable Gadgets $2m$ nodes

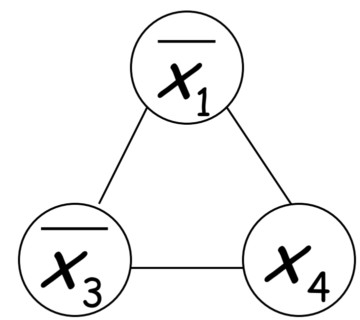
Clause Gadgets $3l$ nodes



Clause 1



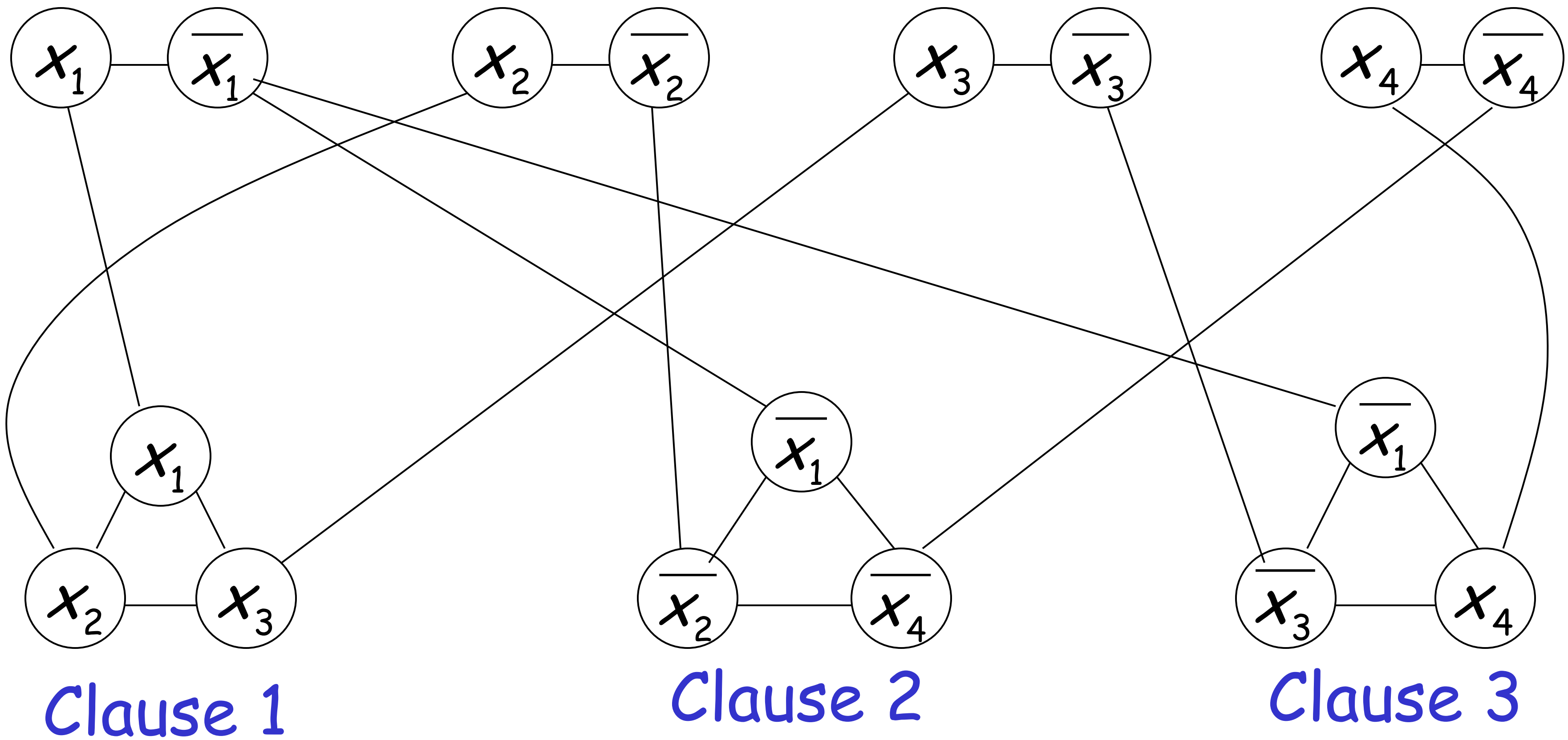
Clause 2



Clause 3

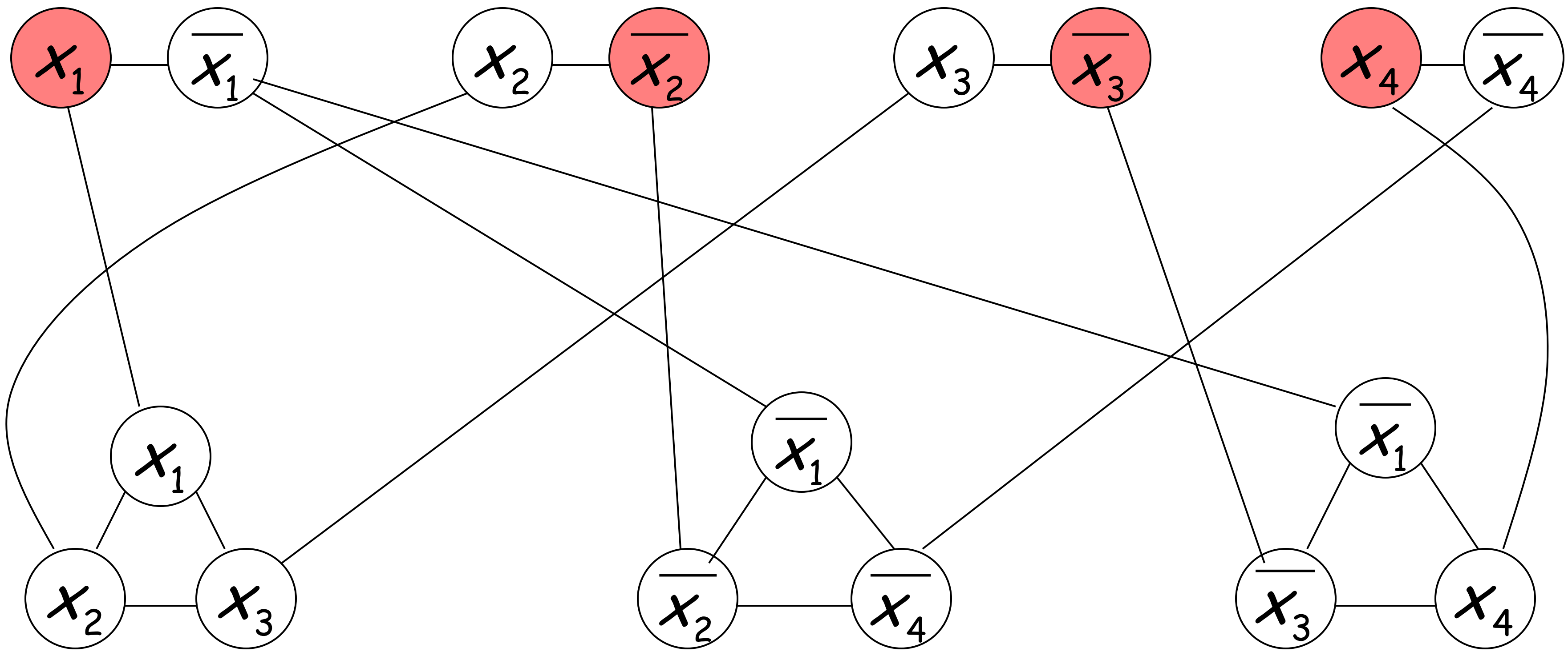
$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$$

Clause 1 Clause 2 Clause 3



$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$$

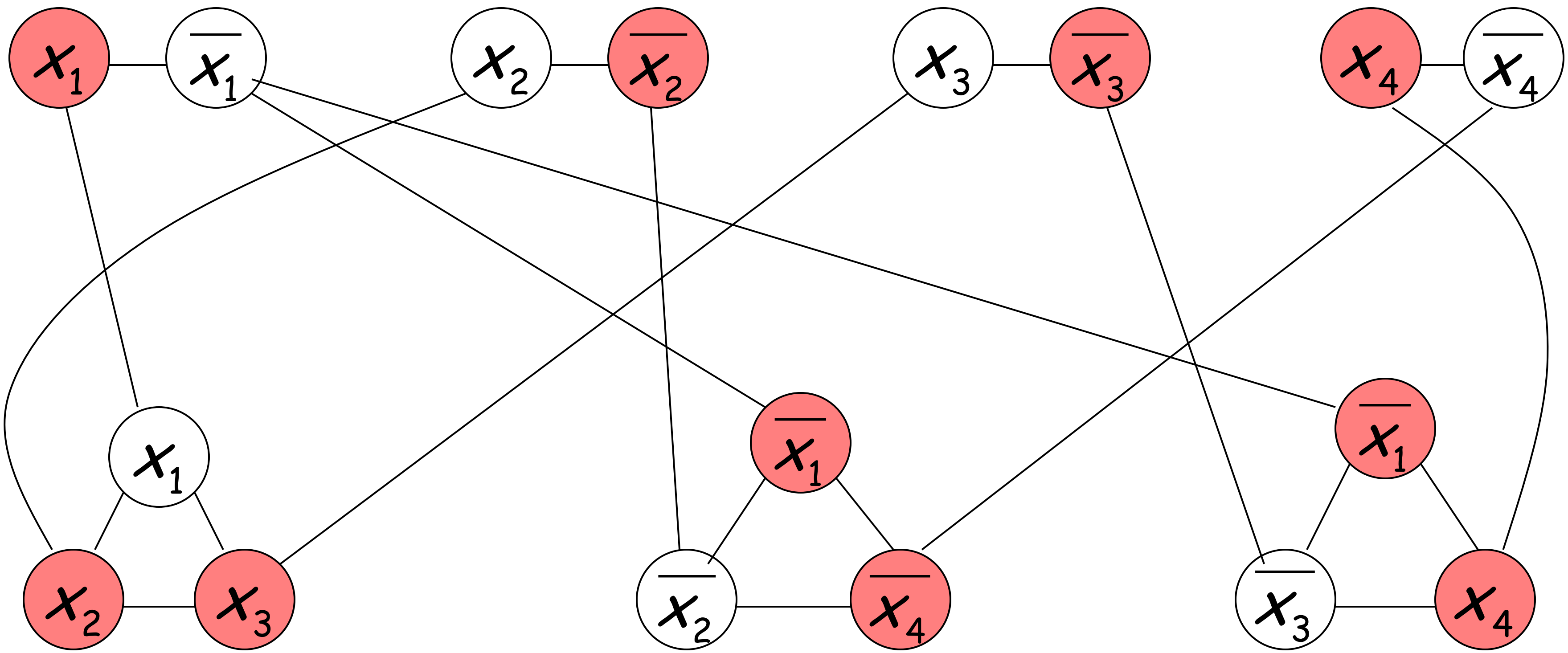
$x_1 = 1 \qquad x_2 = 0 \qquad x_3 = 0 \qquad x_4 = 1$



Put every satisfying literal in the cover

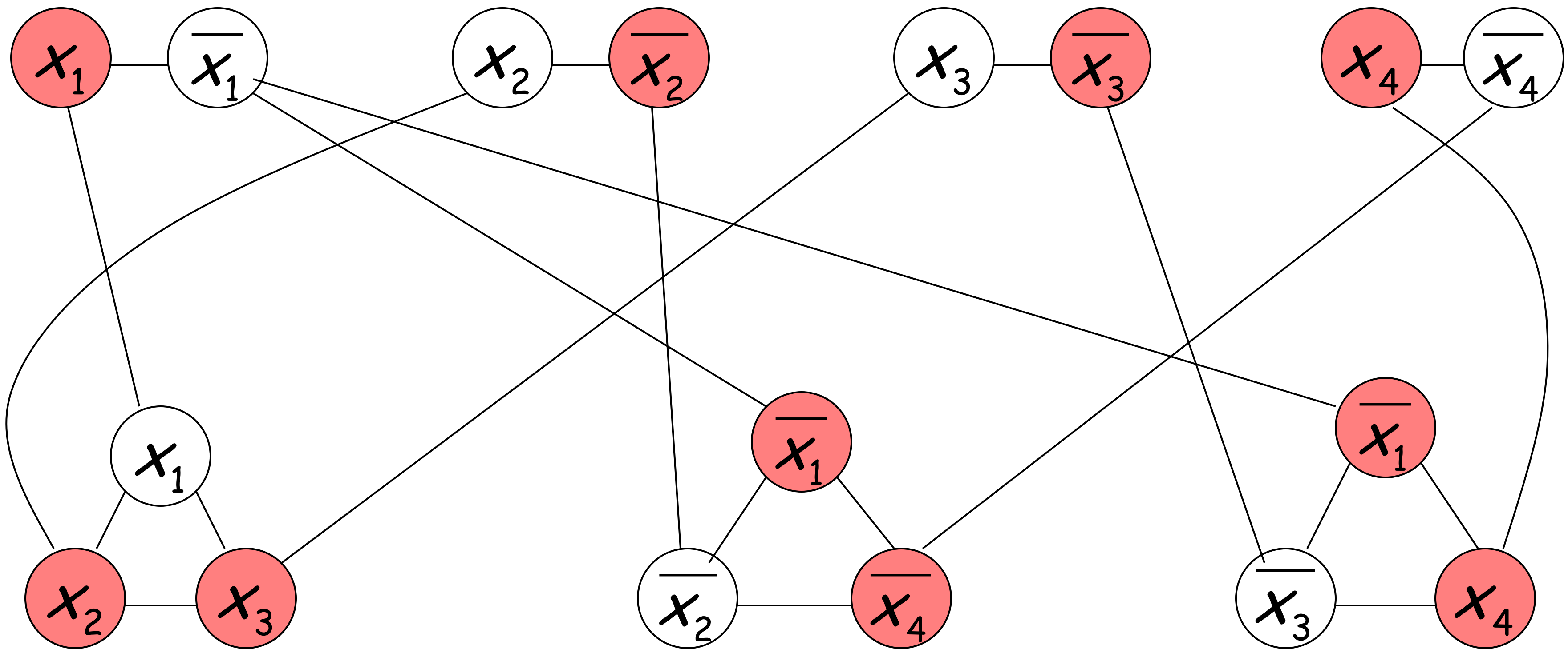
$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4)$$

$x_1 = 1 \qquad x_2 = 0 \qquad x_3 = 0 \qquad x_4 = 1$



Select one satisfying literal in each clause gadget and include the remaining literals in the cover

This is a vertex cover since every edge is adjacent to a chosen node



The proof can be generalized for arbitrary φ

The graph G is constructed in polynomial time with respect to the size of φ

Therefore:

we have reduced in polynomial time

3CNF-SAT to VERTEX-COVER

End of proof