3-satisfiability reduces to vertex cover

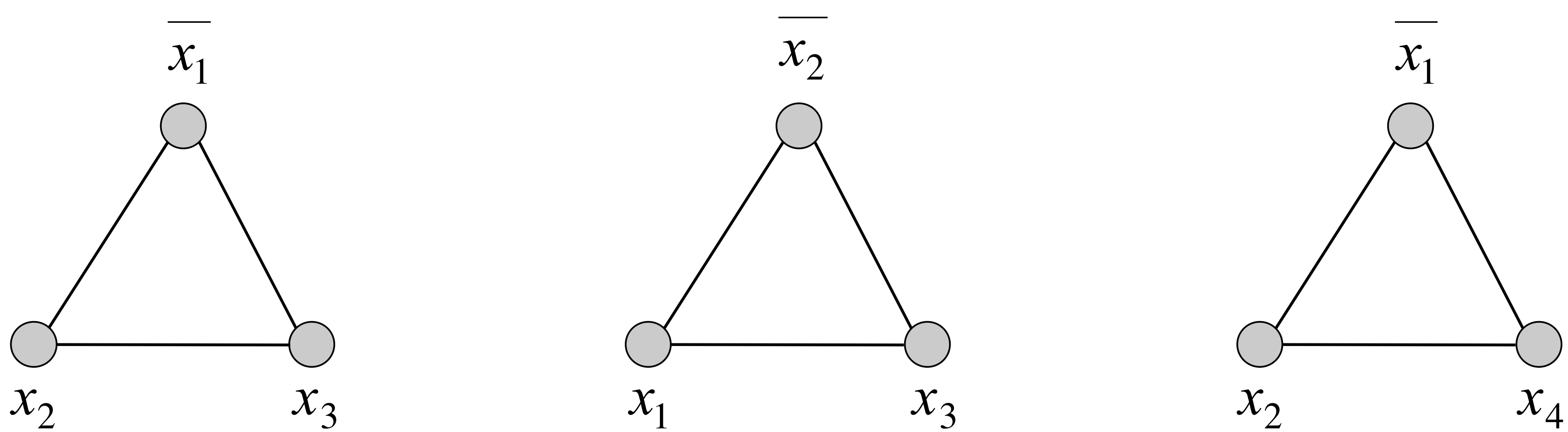
Theorem. 3-SAT $\leq_P$ VERTEX-COVER.

Pf. Given an instance Φ of 3-SAT, we construct an instance (G, k) of VERTEX-COVER that has a vertex cover of size $2k$ iff Φ is satisfiable.

Construction.

- G contains 3 nodes for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.

G



k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

3-satisfiability reduces to vertex cover

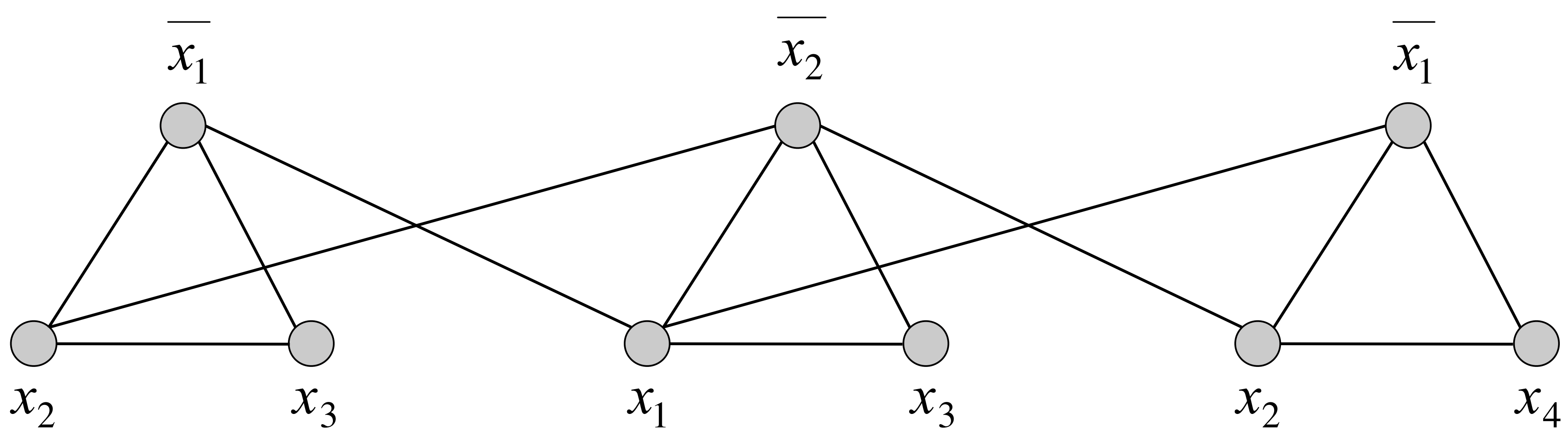
Theorem. 3-SAT $\leq_P$ VERTEX-COVER.

Pf. Given an instance Φ of 3-SAT, we construct an instance (G, k) of VERTEX-COVER that has a vertex cover of size $2k$ iff Φ is satisfiable.

Construction.

- G contains 3 nodes for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.
- Connect literal to each of its negations.

G



k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

3-satisfiability reduces to independent set

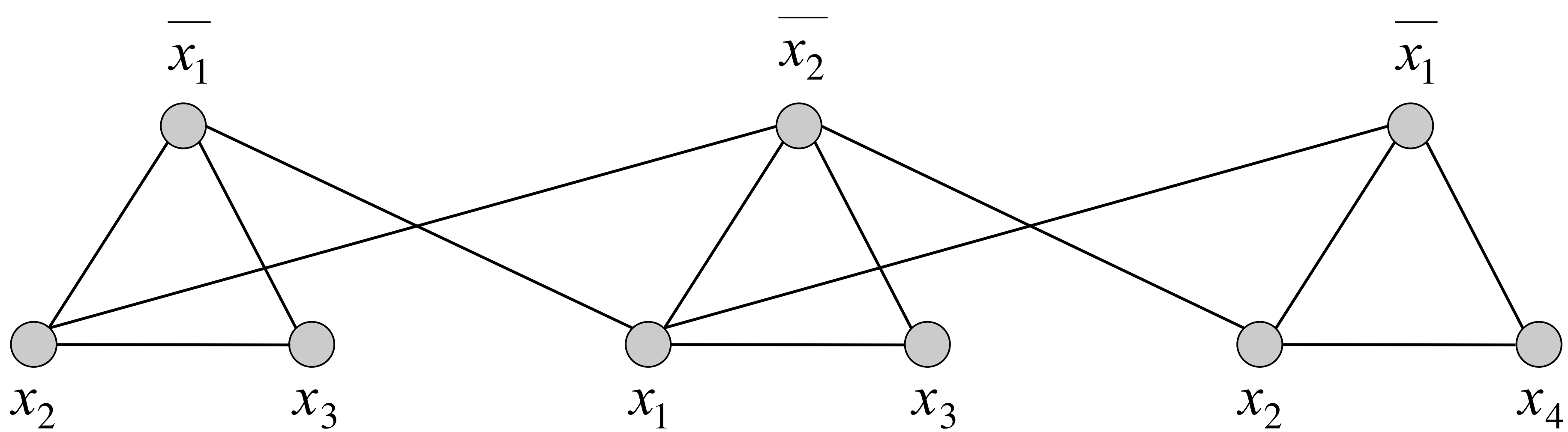
Lemma. G contains vertex cover of size $2k$ iff Φ is satisfiable.

Pf. $\Rightarrow$ Let S be a vertex cover of size $2k$.

- S must contain exactly two nodes in each triangle.
- Set the excluded literal to *true* (and remaining variables consistently).
- Truth assignment is consistent and all clauses are satisfied.

Pf $\Leftarrow$ Given satisfying assignment, select one true literal from each triangle, and exclude that one. This is a vertex cover of size $2k$. ▪

G



k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

Directed hamilton cycle reduces to hamilton cycle

DIR-HAM-CYCLE: Given a digraph $G = (V, E)$, does there exist a simple directed cycle Γ that contains every node in V ?

3-satisfiability reduces to directed hamilton cycle

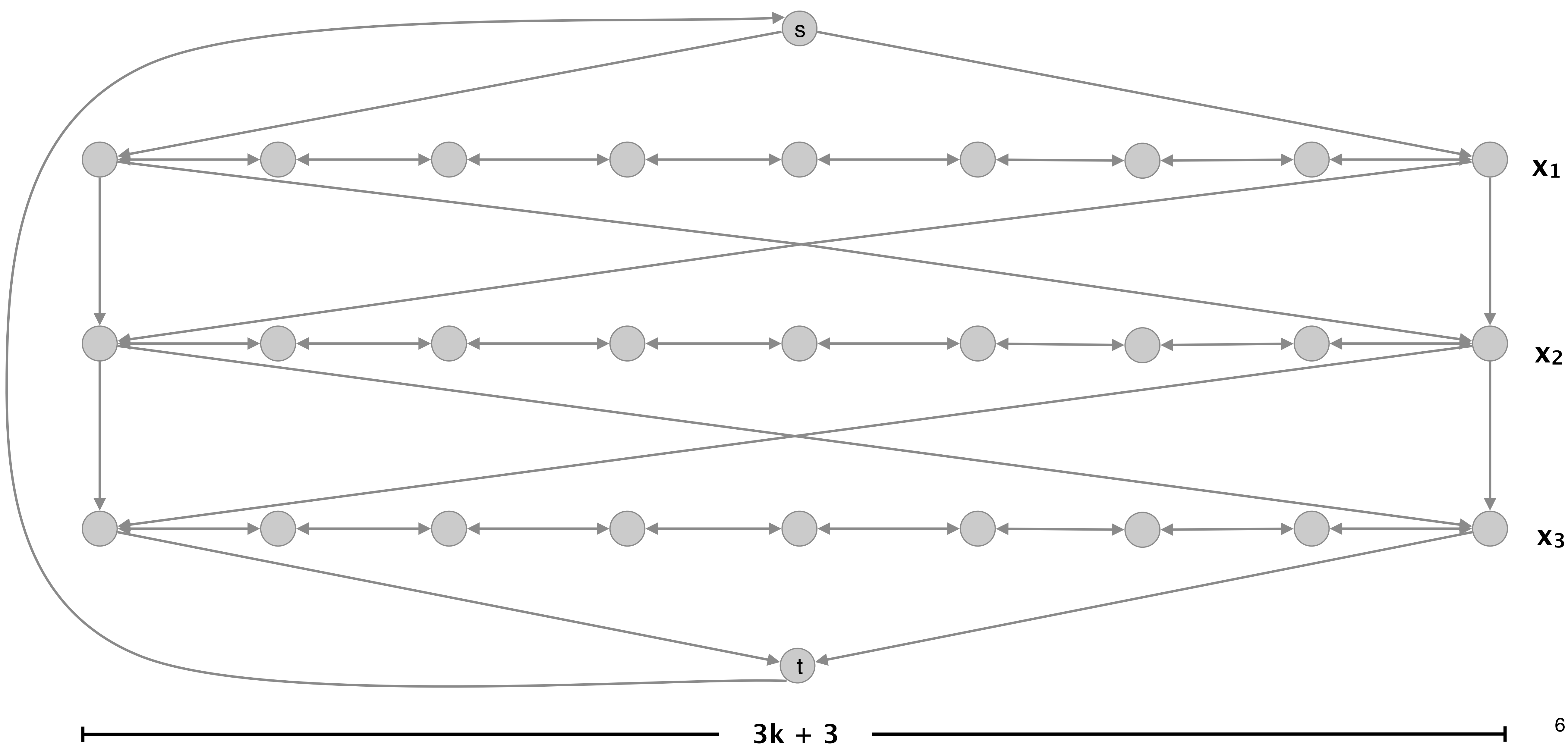
Theorem. 3-SAT $\le_P$ DIR-HAM-CYCLE.

Pf. Given an instance Φ of 3-SAT, we construct an instance of DIR-HAM-CYCLE that has a Hamilton cycle iff Φ is satisfiable.

Construction. First, create graph that has 2^n Hamilton cycles which correspond in a natural way to 2^n possible truth assignments.

3-satisfiability reduces to directed hamilton cycle

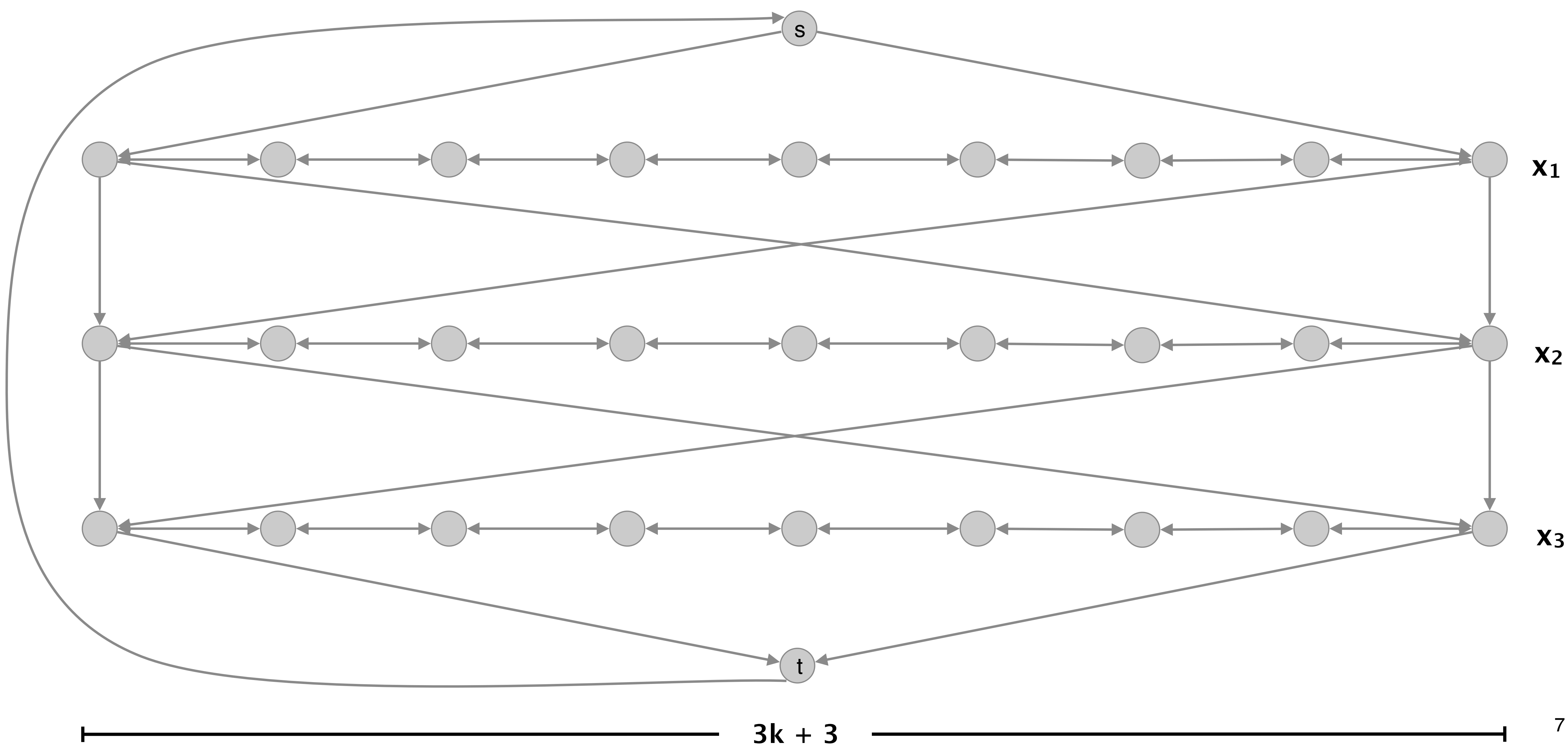
Construction. Given 3-SAT instance Φ with n variables x_i and k clauses



3-satisfiability reduces to directed hamilton cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

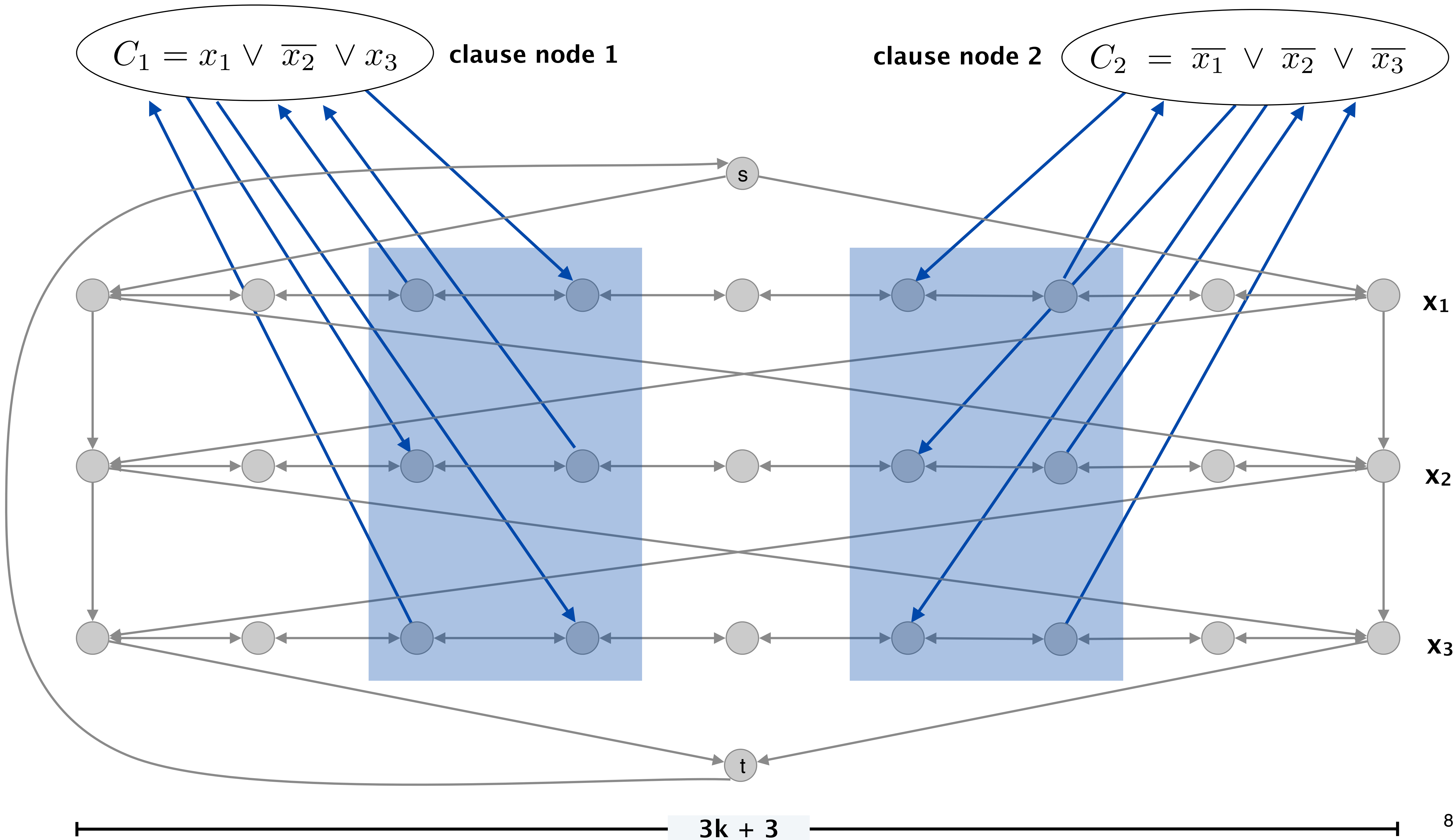
- Intuition: traverse path i from left to right $\Leftrightarrow$ set variable $x_i = true$.



3-satisfiability reduces to directed hamilton cycle

Construction. Given 3-SAT instance Φ with n variables x_i and k clauses.

- For each clause, add a node and 6 edges.



3-satisfiability reduces to directed hamilton cycle

Lemma. Φ is satisfiable iff G has a Hamilton cycle.

Pf. $\Rightarrow$

- Suppose 3-SAT instance has satisfying assignment x^*.
- Then, define Hamilton cycle in G as follows:
  - if $x^*_i = true$, traverse row i from left to right
  - if $x^*_i = false$, traverse row i from right to left
  - for each clause C_j , there will be at least one row i in which we are going in "correct" direction to splice clause node C_j into cycle (and we splice in C_j exactly once)

3-satisfiability reduces to directed hamilton cycle

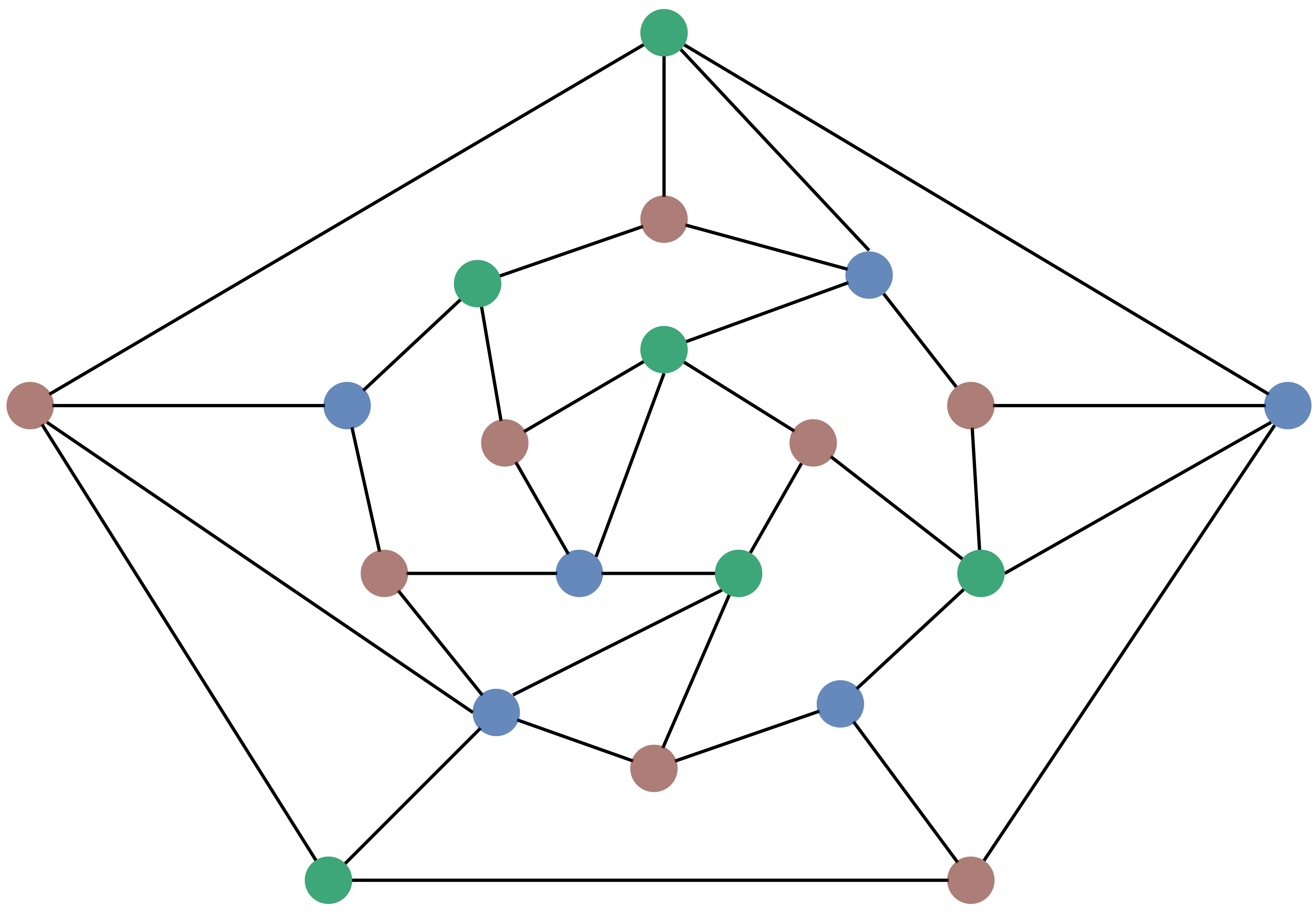
Lemma. Φ is satisfiable iff G has a Hamilton cycle.

Pf. $\Leftarrow$

- Suppose G has a Hamilton cycle Γ.
- If Γ enters clause node C_j , it must depart on mate edge.
  - nodes immediately before and after C_j are connected by an edge $e \in E$
  - removing C_j from cycle, and replacing it with edge e yields Hamilton cycle on $G - \{ C_j \}$
- Continuing in this way, we are left with a Hamilton cycle Γ' in $G - \{ C_1 , C_2 , \ldots, C_k \}$.
- Set $x^*_i = true$ iff Γ' traverses row i left to right.
- Since Γ visits each clause node C_j , at least one of the paths is traversed in "correct" direction, and each clause is satisfied. ▪

3-colorability

3-COLOR. Given an undirected graph G, can the nodes be colored red, green, and blue so that no adjacent nodes have the same color?



yes instance

3-satisfiability reduces to 3-colorability

Theorem. 3-SAT $\le_P$ 3-COLOR.

Pf. Given 3-SAT instance Φ, we construct an instance of 3-COLOR that is 3-colorable iff Φ is satisfiable.

3-satisfiability reduces to 3-colorability

Construction.

(i) Create a graph G with a node for each literal.

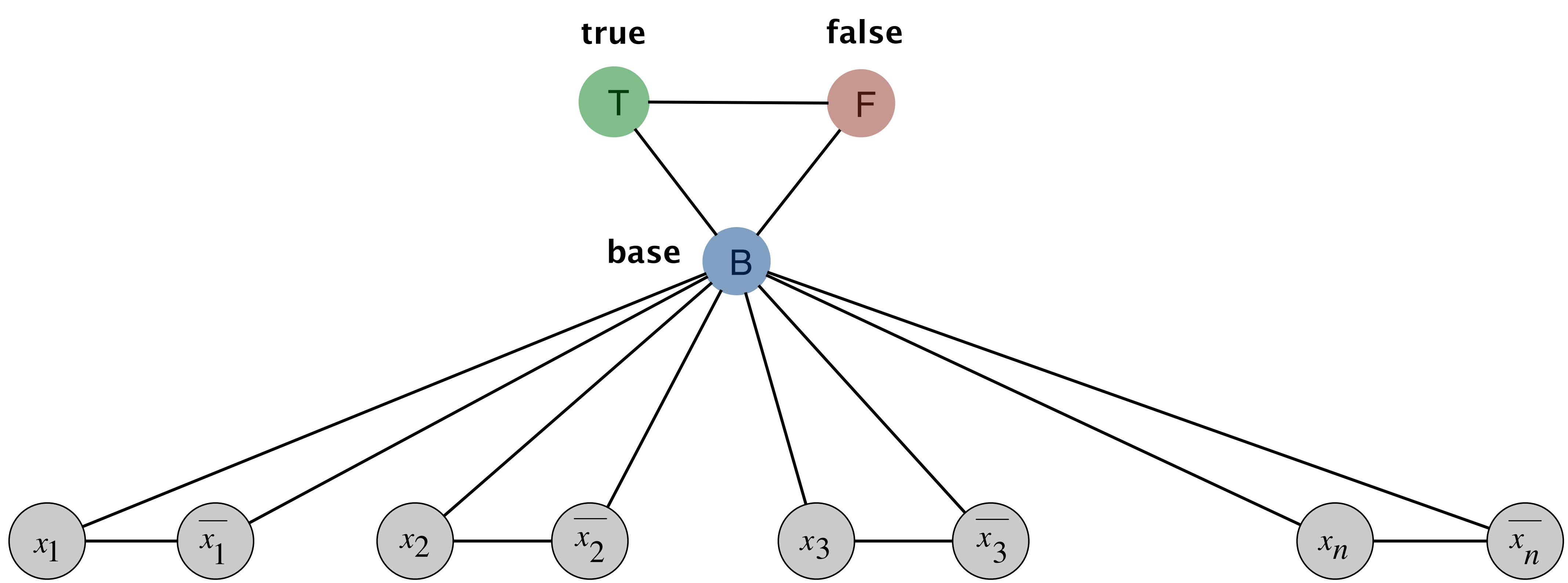
(ii) Connect each literal to its negation.

(iii) Create 3 new nodes T, F, and B; connect them in a triangle.

(iv) Connect each literal to B.

(v) For each clause C_j, add a gadget of 6 nodes and 13 edges.

to be described later

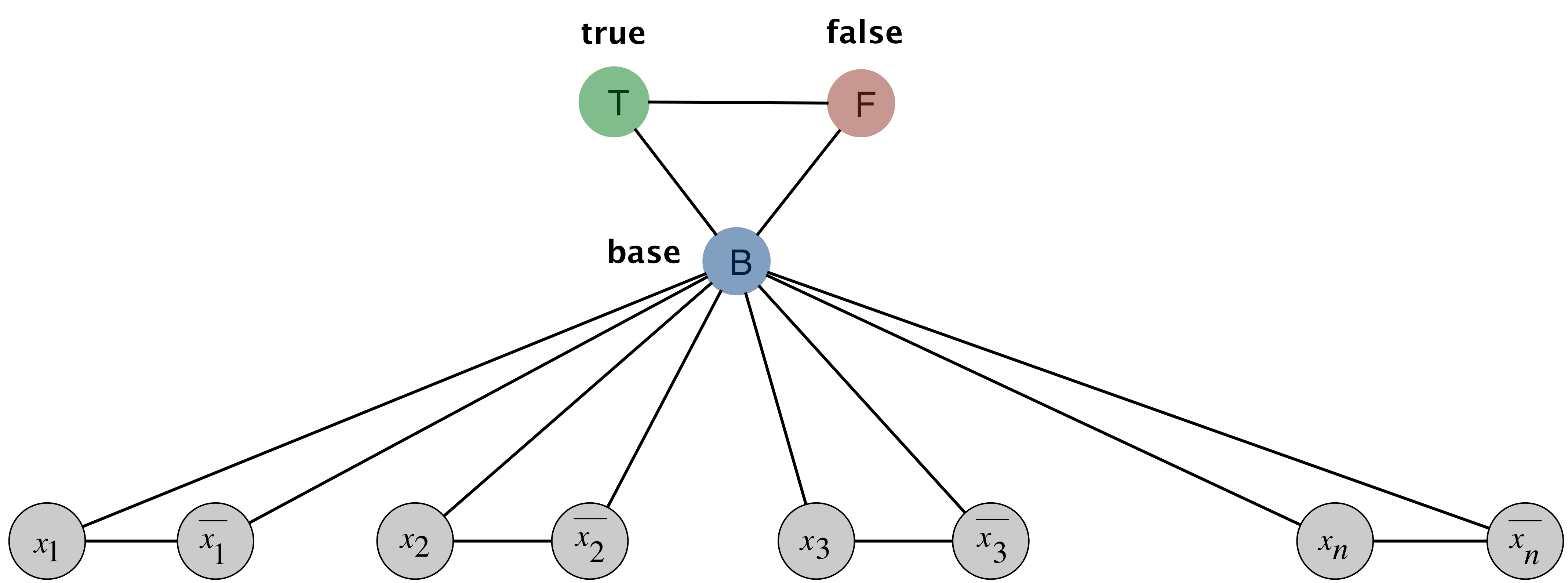


3-satisfiability reduces to 3-colorability

Lemma. Graph G is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph G is 3-colorable.

- Consider assignment that sets all T literals to true.
- (iv) ensures each literal is T or F.
- (ii) ensures a literal and its negation are opposites.

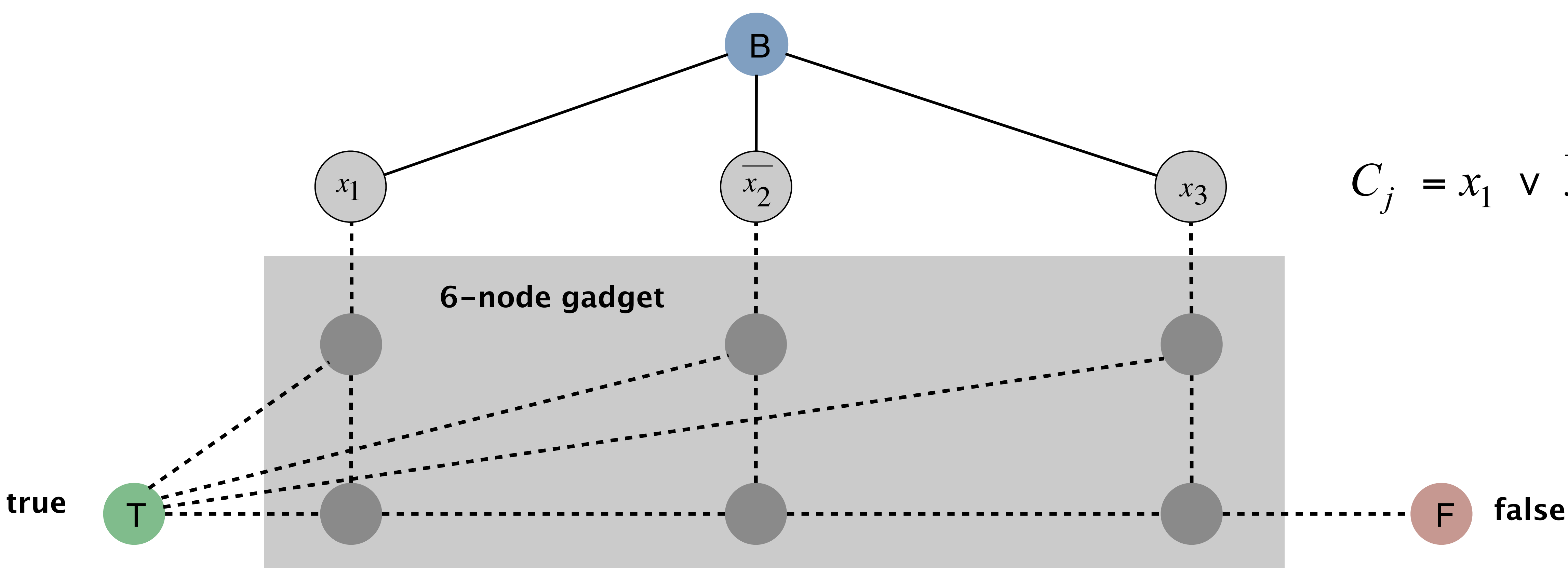


3-satisfiability reduces to 3-colorability

Lemma. Graph G is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph G is 3-colorable.

- Consider assignment that sets all T literals to true.
- (iv) ensures each literal is T or F.
- (ii) ensures a literal and its negation are opposites.
- (v) ensures at least one literal in each clause is T.



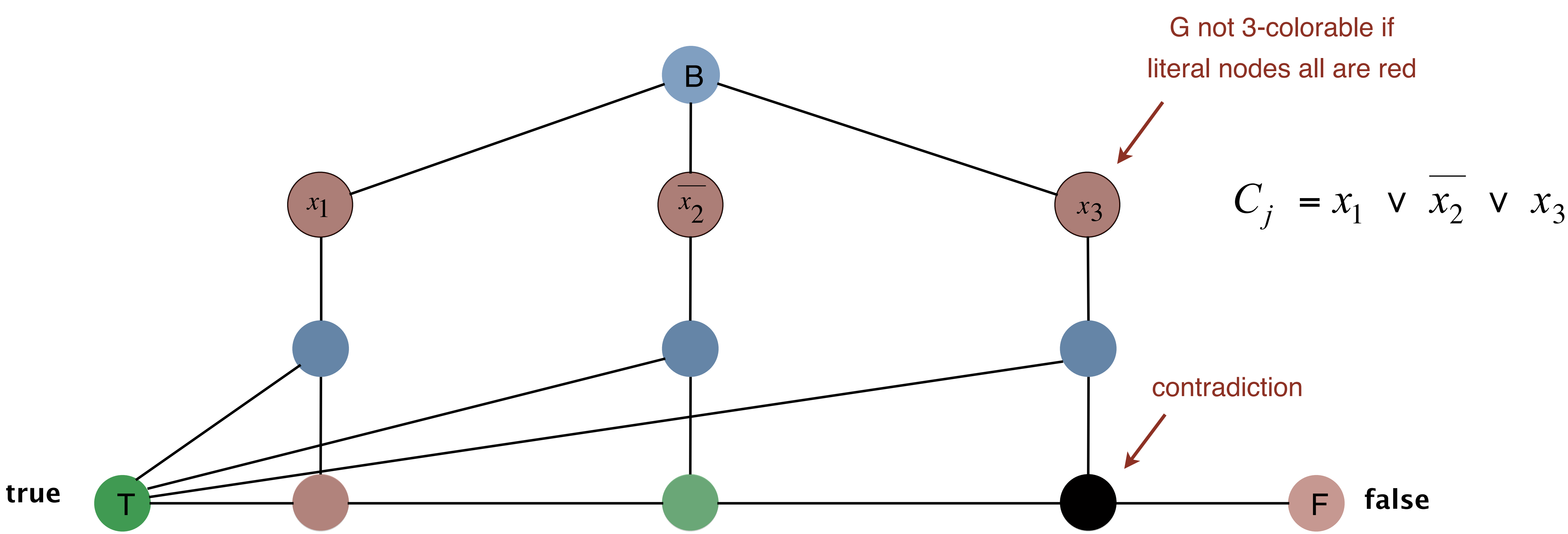
$$C_j = x_1 \lor \overline{x_2} \lor x_3$$

3-satisfiability reduces to 3-colorability

Lemma. Graph G is 3-colorable iff Φ is satisfiable.

Pf. $\Rightarrow$ Suppose graph G is 3-colorable.

- Consider assignment that sets all T literals to true.
- (iv) ensures each literal is T or F.
- (ii) ensures a literal and its negation are opposites.
- (v) ensures at least one literal in each clause is T.



3-satisfiability reduces to 3-colorability

Lemma. Graph G is 3-colorable iff Φ is satisfiable.

Pf. $\Leftarrow$ Suppose 3-SAT instance Φ is satisfiable.

- Color all true literals T.
- Color node below green node F, and node below that B.
- Color remaining middle row nodes B.
- Color remaining bottom nodes T or F as forced. ▪