More NP Completeness

Lecture 25

April 29, 2025

Part I

Wrap Up 3SAT

Last Time: 3SAT

Recall: last time, we wanted to prove that **3SAT** is **NP**-complete.
Need a function f such that if $\varphi \in$ **CNF-SAT** iff $f(\varphi) \in$ **3SAT**.

Last Time: 3SAT

Recall: last time, we wanted to prove that $\mathbf{3SAT}$ is $\mathbf{NP}$-complete.
Need a function f such that if $\varphi \in \mathbf{CNF\text{-}SAT}$ iff $f(\varphi) \in \mathbf{3SAT}$.

f converts each clause in φ into multiple size-three clauses:

- If $C = (\ell_1)$, include clauses $(\ell_1 \vee x_{C1} \vee x_{C2})$, $(\ell_1 \vee \overline{x_{C1}} \vee x_{C2})$, $(\ell_1 \vee x_{C1} \vee \overline{x_{C2}})$, and $(\ell_1 \vee \overline{x_{C1}} \vee \overline{x_{C2}})$.
- If $C = (\ell_1 \vee \ell_2)$, include clauses $(\ell_1 \vee \ell_2 \vee x_C)$ and $(\ell_1 \vee \ell_2 \vee \overline{x_C})$.
- If $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, include C.
- If $C = (\ell_1 \vee \ldots \vee \ell_k)$ (for $k \geq 4$), include clauses $(\ell_1 \vee \ell_2 \vee x_{C1})$, $(\overline{x_{C1}} \vee \ell_3 \vee x_{C2})$, $\ldots$, $(\overline{x_{C(k-3)}} \vee \ell_{k-1} \vee \ell_k)$.

Last Time: 3SAT

Recall: last time, we wanted to prove that $\mathbf{3SAT}$ is $\mathbf{NP}$-complete.
Need a function f such that if $\varphi \in \mathbf{CNF\text{-}SAT}$ iff $f(\varphi) \in \mathbf{3SAT}$.

f converts each clause in φ into multiple size-three clauses:

- If $C = (\ell_1)$, include clauses $(\ell_1 \vee x_{C1} \vee x_{C2})$, $(\ell_1 \vee \overline{x_{C1}} \vee x_{C2})$, $(\ell_1 \vee x_{C1} \vee \overline{x_{C2}})$, and $(\ell_1 \vee \overline{x_{C1}} \vee \overline{x_{C2}})$.
- If $C = (\ell_1 \vee \ell_2)$, include clauses $(\ell_1 \vee \ell_2 \vee x_C)$ and $(\ell_1 \vee \ell_2 \vee \overline{x_C})$.
- If $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, include C.
- If $C = (\ell_1 \vee \ldots \vee \ell_k)$ (for $k \geq 4$), include clauses $(\ell_1 \vee \ell_2 \vee x_{C1})$, $(\overline{x_{C1}} \vee \ell_3 \vee x_{C2})$, $\ldots$, $(\overline{x_{C(k-3)}} \vee \ell_{k-1} \vee \ell_k)$.

Need to show: φ is satisfiable iff $f(\varphi)$ is!

CNF-SAT to 3SAT: If

$\varphi \to f(\varphi)$:

$(\ell_1) \to (\ell_1 \vee x_{C1} \vee x_{C2}) \wedge (\ell_1 \vee \overline{x_{C1}} \vee x_{C2}) \wedge (\ell_1 \vee x_{C1} \vee \overline{x_{C2}}) \wedge (\ell_1 \vee \overline{x_{C1}} \vee \overline{x_{C2}})$

$(\ell_1 \vee \ell_2) \to (\ell_1 \vee \ell_2 \vee x_C) \wedge (\ell_1 \vee \ell_2 \vee \overline{x_C})$

$(\ell_1 \vee \ell_2 \vee \ell_3) \to (\ell_1 \vee \ell_2 \vee \ell_3)$

$(\ell_1 \vee \ldots \vee \ell_k) \to (\ell_1 \vee \ell_2 \vee x_{C1}) \wedge (\overline{x_{C1}} \vee \ell_3 \vee x_{C2}) \wedge \ldots \wedge (\overline{x_{C(k-3)}} \vee \ell_{k-1} \vee \ell_k)$

$x_{C1} = T$ $\quad x_{C2} = T \ldots x_{C(k-3)} = T$

Claim: If $f(\varphi)$ is satisfiable, so is φ.

Given a satisfying assignment to $f(\varphi)$, keep the values of all the "original" variables.

CNF-SAT to 3SAT: Only-If

$\varphi \to f(\varphi)$:

$(\ell_1) \to (\ell_1 \vee x_{C1} \vee x_{C2}) \wedge (\ell_1 \vee \overline{x_{C1}} \vee x_{C2}) \wedge (\ell_1 \vee x_{C1} \vee \overline{x_{C2}}) \wedge (\ell_1 \vee \overline{x_{C1}} \vee \overline{x_{C2}})$

$(\ell_1 \vee \ell_2) \to (\ell_1 \vee \ell_2 \vee x_C) \wedge (\ell_1 \vee \ell_2 \vee \overline{x_C})$

$(\ell_1 \vee \ell_2 \vee \ell_3) \to (\ell_1 \vee \ell_2 \vee \ell_3)$

$(\ell_1 \vee \ldots \vee \ell_k) \to (\ell_1 \vee \ell_2 \vee x_{C1}) \wedge (\overline{x_{C1}} \vee \ell_3 \vee x_{C2}) \wedge \ldots \wedge (\overline{x_{C(k-3)}} \vee \ell_{k-1} \vee \ell_k)$

$x_{C1} = T$ $x_{C2} = F$ ... $x_{C(k-3)} = F$

Claim: If φ is satisfiable, so is $f(\varphi)$.

Given a satisfying assignment to φ, construct an assignment to $f(\varphi)$ by keeping all "original" vars

① if # literals is ≤ 3 : assign the added vars arbitrarily

② if # literals is ≥ 4:

Part II

Independent Set

Independent Set

Recall: An independent set is $\boldsymbol{S} \subseteq \boldsymbol{V}$ such that no vertices in $\boldsymbol{S}$ have an edge between them. Let $\boldsymbol{IS} = \{(\boldsymbol{G}, \boldsymbol{k}) \mid \boldsymbol{G}$ has an IS of size $\boldsymbol{k}\}$.

Independent Set

Recall: An independent set is $S \subseteq V$ such that no vertices in S have an edge between them. Let $IS = \{(G, k) \mid G \text{ has an IS of size } k\}$.

Claim

IS is NP-complete.

Independent Set

Recall: An independent set is $S \subseteq V$ such that no vertices in S have an edge between them. Let $IS = \{(G, k) \mid G \text{ has an IS of size } k\}$.

Claim

IS is NP-complete.

IS is in NP: w is the description of an IS of size k.

Independent Set

Recall: An independent set is $S \subseteq V$ such that no vertices in S have an edge between them. Let $IS = \{(G, k) \mid G$ has an IS of size $k\}$.

Claim

IS *is* ***NP****-complete.*

IS is in ***NP***: w is the description of an IS of size k.

What problem should we reduce to ***IS*** in order to prove hardness?

SAT CNF-SAT 3SAT

3SAT to IS: Intuition

We have a 3SAT formula φ. We want to construct (G, k) such that φ is satisfiable iff G has an IS of size k.

3SAT to IS: Intuition

We have a 3SAT formula φ. We want to construct (G, k) such that φ is satisfiable iff G has an IS of size k.

Key observation: φ is satisfiable iff we can pick one literal from each clause to be true.

(We don't need to pick *every* true literal—just don't pick two that contradict!)

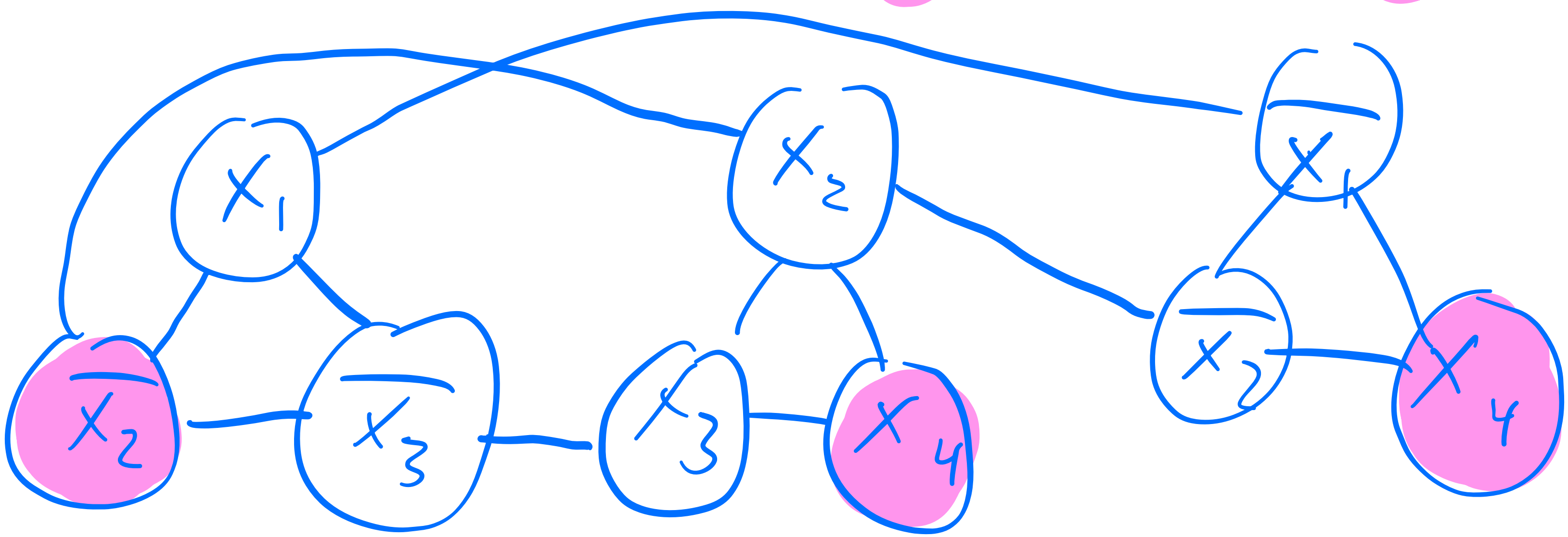
3SAT to IS: Intuition

We have a 3SAT formula φ. We want to construct (G, k) such that φ is satisfiable iff G has an IS of size k.

Key observation: φ is satisfiable iff we can pick one literal from each clause to be true.

(We don't need to pick *every* true literal—just don't pick two that contradict!)

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



3SAT to IS: Reduction

Let $f(\varphi) = (G, k)$ where:

3SAT to IS: Reduction

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.

3SAT to IS: Reduction

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.
- For each variable x_i in φ, add edges between each vertex labeled x_i and each labeled $\overline{x_i}$.

3SAT to IS: Reduction

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.
- For each variable x_i in φ, add edges between each vertex labeled x_i and each labeled $\overline{x_i}$.
- We set k as the number of clauses in φ.

3SAT to IS: Reduction

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.
- For each variable x_i in φ, add edges between each vertex labeled x_i and each labeled $\overline{x_i}$.
- We set k as the number of clauses in φ.

This reduction clearly runs in polynomial time. (In fact, quadratic.)
Just need to show $\varphi \in \mathbf{3SAT}$ iff $f(\varphi) \in \mathbf{IS}$.

3SAT to IS: Only-If

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.
- For each variable x_i in φ, add edges between each vertex labeled x_i and each labeled $\overline{x_i}$.
- We set k as the number of clauses in φ.

Claim: If φ is satisfiable, G has an IS of size k.

Given a satisfying assignment to φ, I know each clause has ≥ 1 true literal

For each clause, pick one literal, and include the corresponding vertex in S.

3SAT to IS: If

Let $f(\varphi) = (G, k)$ where:

- For each clause C in φ, we add three connected vertices to G, labeled with the literals of C.
- For each variable x_i in φ, add edges between each vertex labeled x_i and each labeled $\overline{x_i}$.
- We set k as the number of clauses in φ.

Claim: If G has an IS of size k, φ is satisfiable.

Given an IS $S \subseteq V$ of size k.

① S must pick exactly one vertex from each clause gadget.

② S does not contain vertices labeled x_i & $\overline{x_i}$

$\Rightarrow$ assign variables where $x_i = T$ if x_i is in S, $x_i = F$ if $\overline{x_i}$ is in S, else arbitrary

Related problems

Recall: Independent Set and Clique reduce to each other.

(G has an IS of size k iff $\overline{G}$ has a clique of size k.)

Related problems

Recall: Independent Set and Clique reduce to each other.

(G has an IS of size k iff $\overline{G}$ has a clique of size k.)

Claim

$CLIQUE = \{(G, k) \mid G$ has a clique of size $k\}$ is NP-complete.

Related problems

Recall: Independent Set and Clique reduce to each other.

(G has an IS of size k iff $\overline{G}$ has a clique of size k.)

Claim

$CLIQUE = \{(G, k) \mid G$ *has a clique of size* $k\}$ *is* NP*-complete.*

A ***vertex cover*** is $S \subseteq V$ such that every edge in G has at least one endpoint in S.

Related problems

Recall: Independent Set and Clique reduce to each other.

(G has an IS of size k iff $\overline{G}$ has a clique of size k.)

Claim

$CLIQUE = \{(G, k) \mid G$ has a clique of size $k\}$ is NP-complete.

A ***vertex cover*** is $S \subseteq V$ such that every edge in G has at least one endpoint in S.

Observation: S is a vertex cover iff $V - S$ is an independent set.

Related problems

Recall: Independent Set and Clique reduce to each other.

(G has an IS of size k iff $\overline{G}$ has a clique of size k.)

Claim

$CLIQUE = \{(G, k) \mid G$ has a clique of size $k\}$ is NP-complete.

A ***vertex cover*** is $S \subseteq V$ such that every edge in G has at least one endpoint in S.

Observation: S is a vertex cover iff $V - S$ is an independent set.

Claim

$VC = \{(G, k) \mid G$ has a vertex cover of size $k\}$ is NP-complete.

Part III

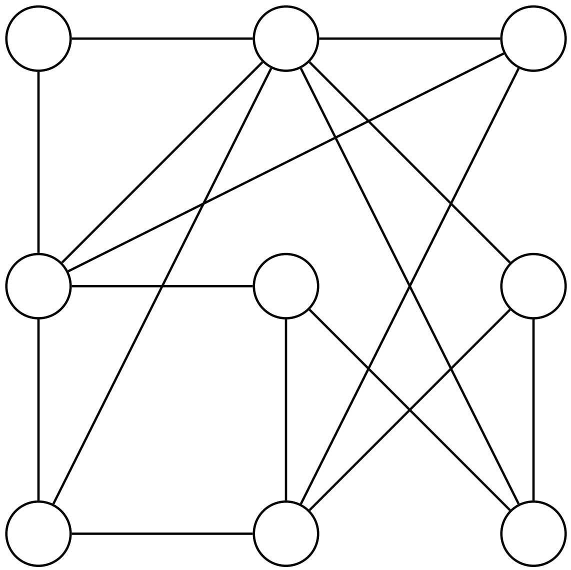
3-Coloring

Graph Coloring

For a graph G, a valid coloring is an assignment of “colors” to each vertex such that no edge has the same color on both ends.

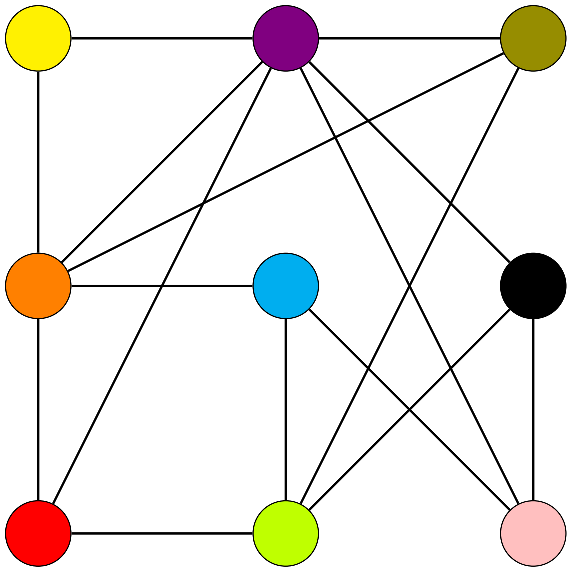
Graph Coloring

For a graph G, a valid coloring is an assignment of "colors" to each vertex such that no edge has the same color on both ends.



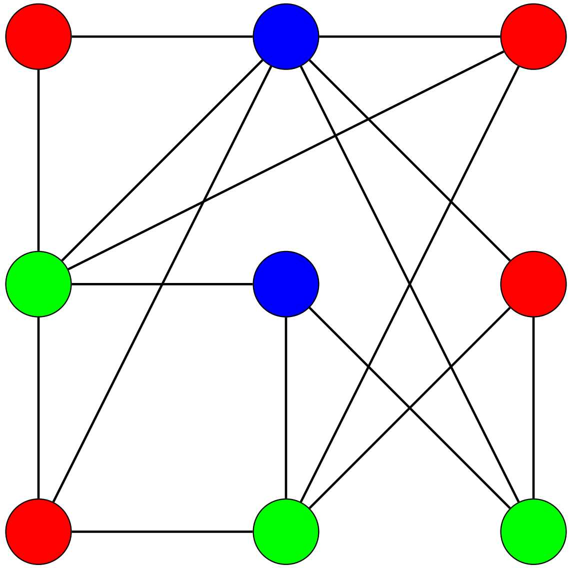
Graph Coloring

For a graph G, a valid coloring is an assignment of "colors" to each vertex such that no edge has the same color on both ends.



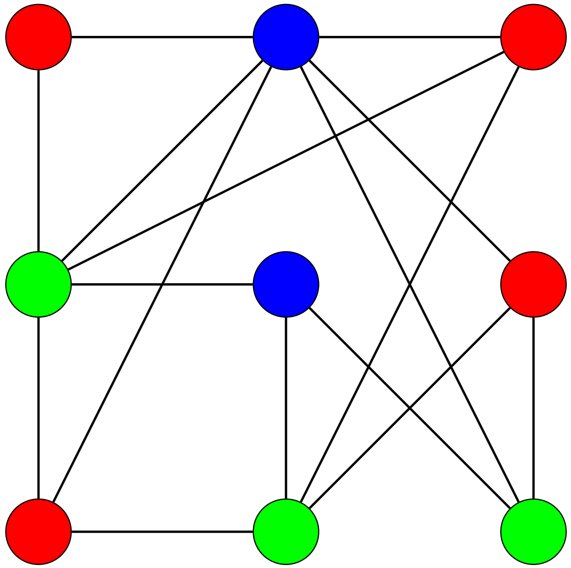
Graph Coloring

For a graph G, a valid coloring is an assignment of “colors” to each vertex such that no edge has the same color on both ends.



Graph Coloring

For a graph G, a valid coloring is an assignment of "colors" to each vertex such that no edge has the same color on both ends.

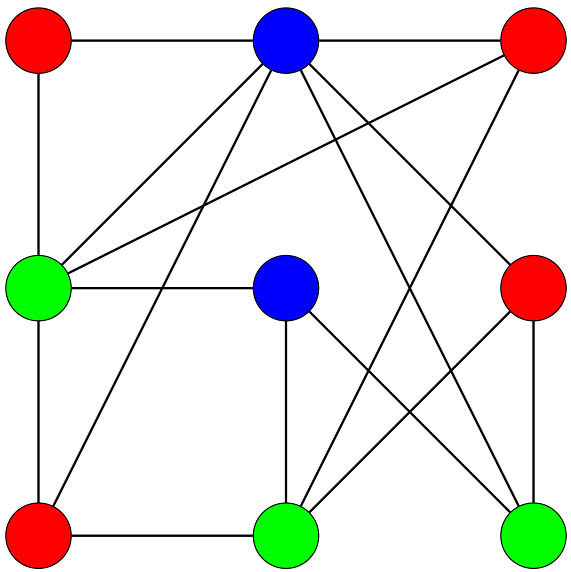


Key question: given a graph G, what is the fewest colors we can use?

(This is referred to as the "chromatic number" of G)

Graph Coloring

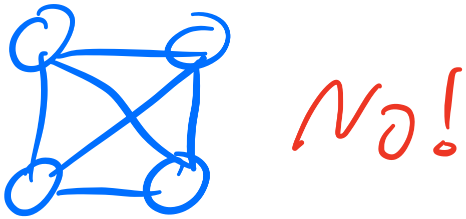
For a graph G, a valid coloring is an assignment of "colors" to each vertex such that no edge has the same color on both ends.



Key question: given a graph G, what is the fewest colors we can use?

(This is referred to as the "chromatic number" of G)

Our focus: Can G be 3-colored?



3COLOR

Claim

$\mathbf{3}\boldsymbol{COLOR} = \{\boldsymbol{G} \mid \boldsymbol{G}$ *has a valid* $\mathbf{3}$*-coloring*$\}$ *is* $\boldsymbol{NP}$*-complete.*

3COLOR

Claim

$\mathbf{3COLOR} = \{\mathbf{G} \mid \mathbf{G}$ has a valid $\mathbf{3}$-coloring$\}$ is $\mathbf{NP}$-complete.

$\mathbf{3COLOR}$ is in $\mathbf{NP}$: $\mathbf{w}$ is the description of a valid $\mathbf{3}$-coloring.

3COLOR

Claim

$\mathbf{3COLOR} = \{\mathbf{G} \mid \mathbf{G}$ has a valid $\mathbf{3}$-coloring$\}$ is $\mathbf{NP}$-complete.

3COLOR is in ***NP***: ***w*** is the description of a valid **3**-coloring.

What problem should we reduce to ***3COLOR*** in order to prove hardness?

3SAT

IS (or Clique/VC)

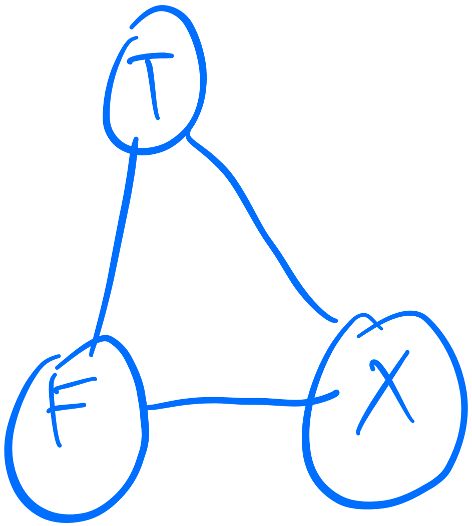
3SAT to 3COLOR: Intuition

We have a **3*SAT*** formula φ. We want to construct a graph G such that φ is satisfiable iff G is 3-colorable.

3SAT to 3COLOR: Intuition

We have a $\mathbf{3}\boldsymbol{SAT}$ formula φ. We want to construct a graph $\boldsymbol{G}$ such that φ is satisfiable iff $\boldsymbol{G}$ is $\mathbf{3}$-colorable.

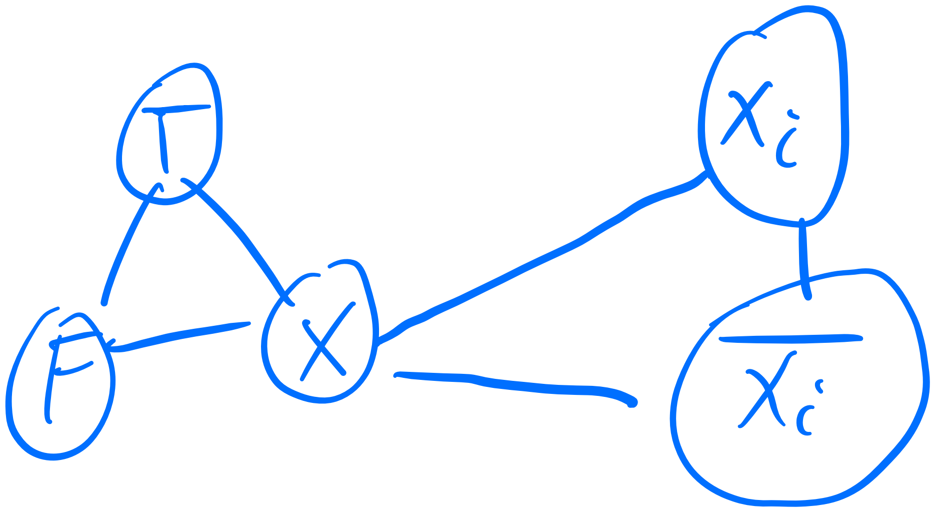
Step 1: associate "colors" with True / False values.



3SAT to 3COLOR: Intuition

We have a **3*SAT*** formula φ. We want to construct a graph G such that φ is satisfiable iff G is **3**-colorable.

Step 1: associate "colors" with True / False values.
Step 2: ensure each variable is assigned to True or to False.



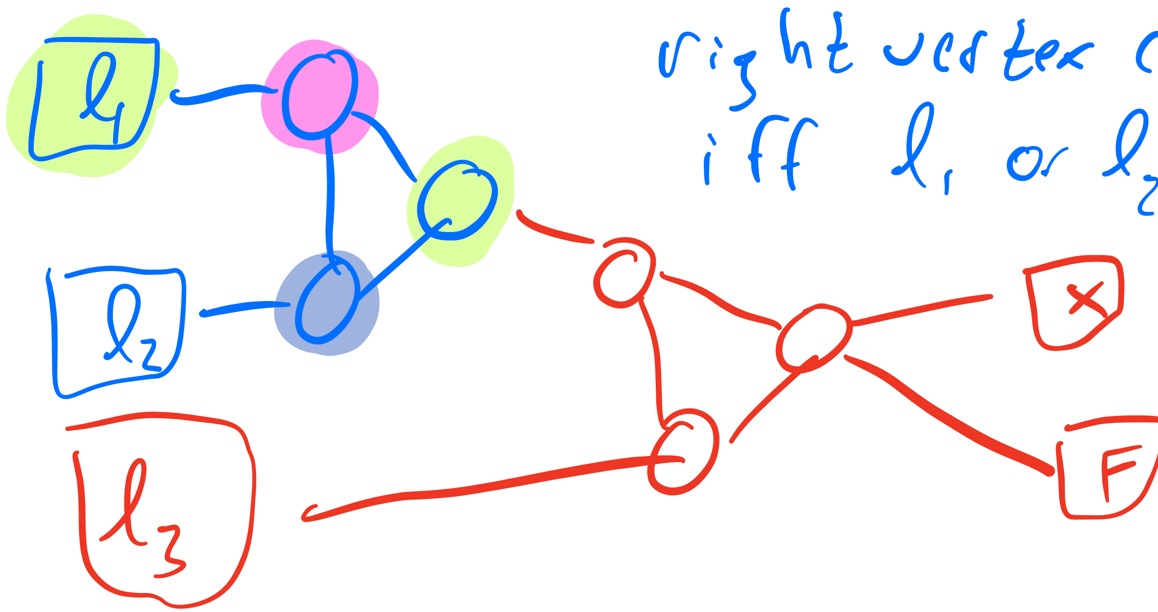
3SAT to 3COLOR: Intuition

We have a **3SAT** formula φ. We want to construct a graph G such that φ is satisfiable iff G is **3**-colorable.

Step 1: associate "colors" with True / False values.
Step 2: ensure each variable is assigned to True or to False.
Step 3: ensure each clause is satisfied.

Simple case: $(l_1 \vee l_2)$ $(l_1 \vee l_2 \vee l_3)$

right vertex can be colored "true" iff l_1 or l_2 is colored "true"



3SAT to 3COLOR: Reduction

Let $f(\varphi) = G$, where:

3SAT to 3COLOR: Reduction

Let $f(\varphi) = G$, where:

- We add vertices T, F, and X to G, all connected.

3SAT to 3COLOR: Reduction

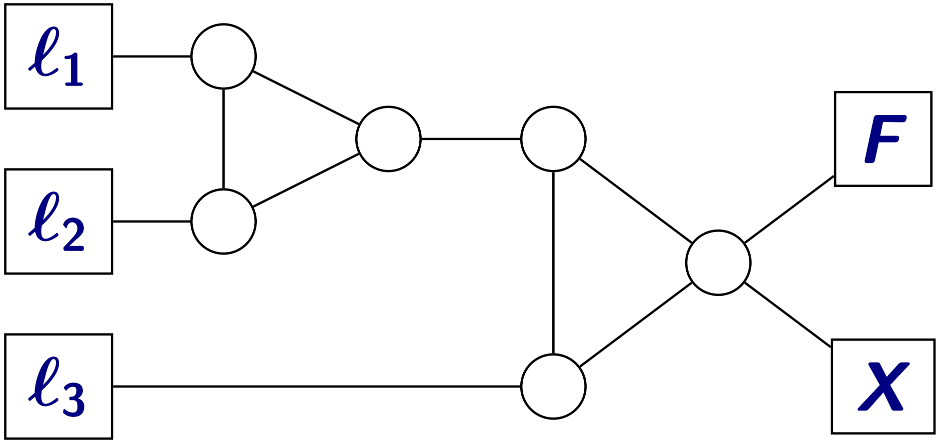
Let $f(\varphi) = G$, where:

- We add vertices T, F, and X to G, all connected.
- For each variable x_i in φ, we add vertices x_i and $\overline{x_i}$, connected to each other and to X.

3SAT to 3COLOR: Reduction

Let $f(\varphi) = G$, where:

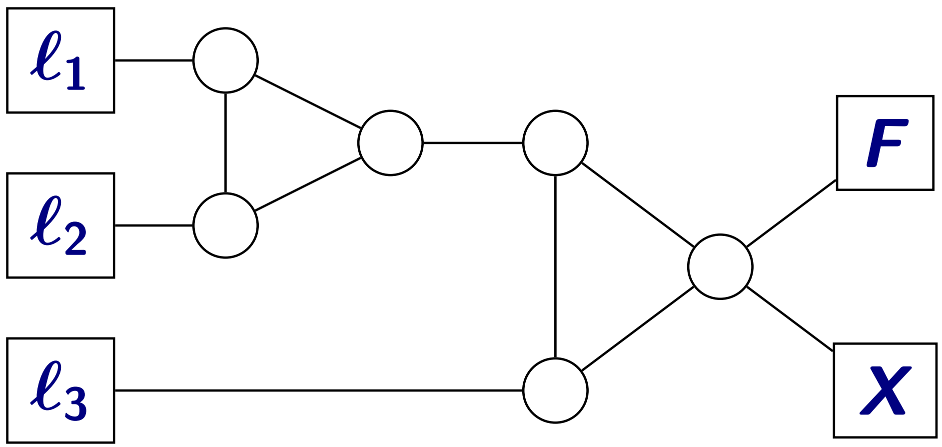
- We add vertices T, F, and X to G, all connected.
- For each variable x_i in φ, we add vertices x_i and $\overline{x_i}$, connected to each other and to X.
- For each clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, we add the following "gadget" to G: (Note: square vertices already exist in G.)



3SAT to 3COLOR: Reduction

Let $f(\varphi) = G$, where:

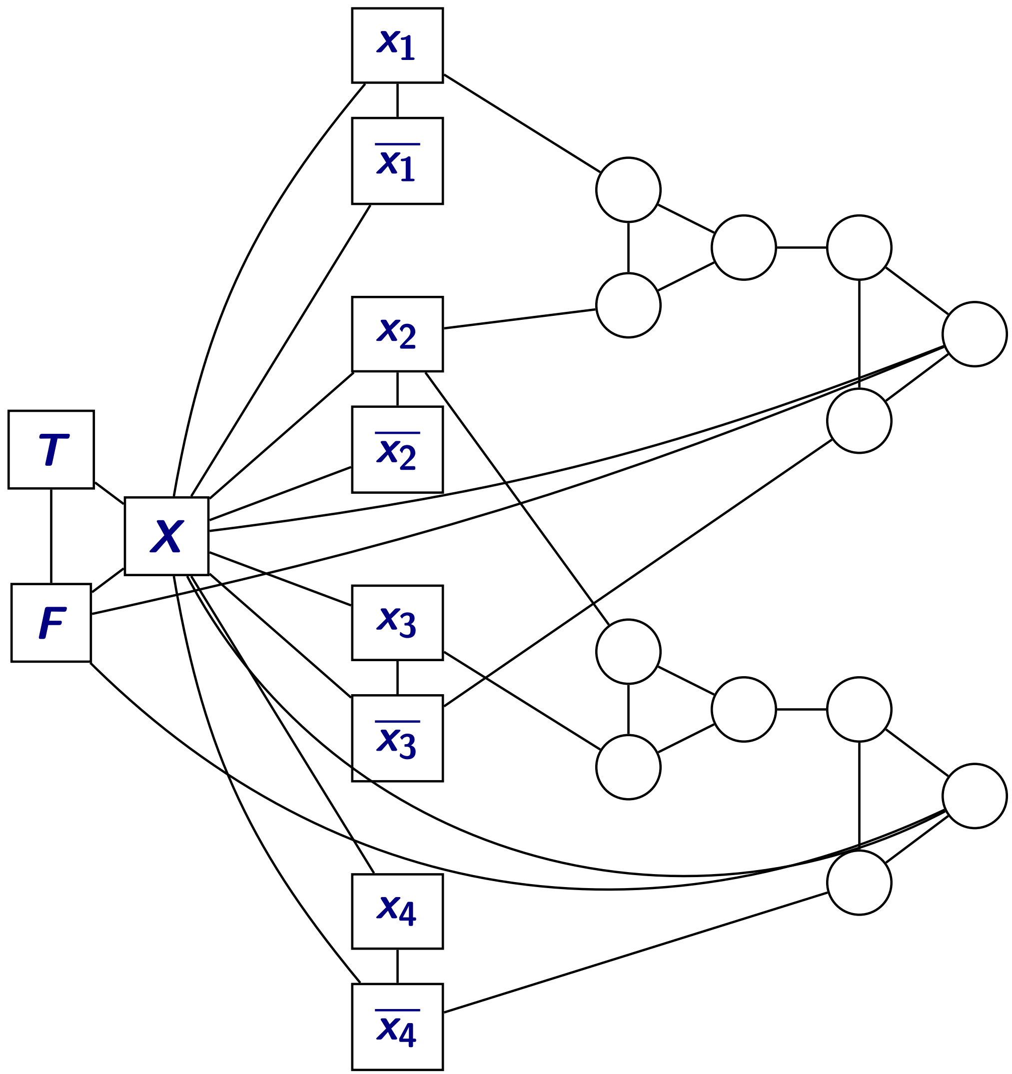
- We add vertices T, F, and X to G, all connected.
- For each variable x_i in φ, we add vertices x_i and $\overline{x_i}$, connected to each other and to X.
- For each clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, we add the following "gadget" to G: (Note: square vertices already exist in G.)



This reduction clearly runs in polynomial time. (In fact, linear.)
Just need to show $\varphi \in 3SAT$ iff $G \in 3COLOR$.

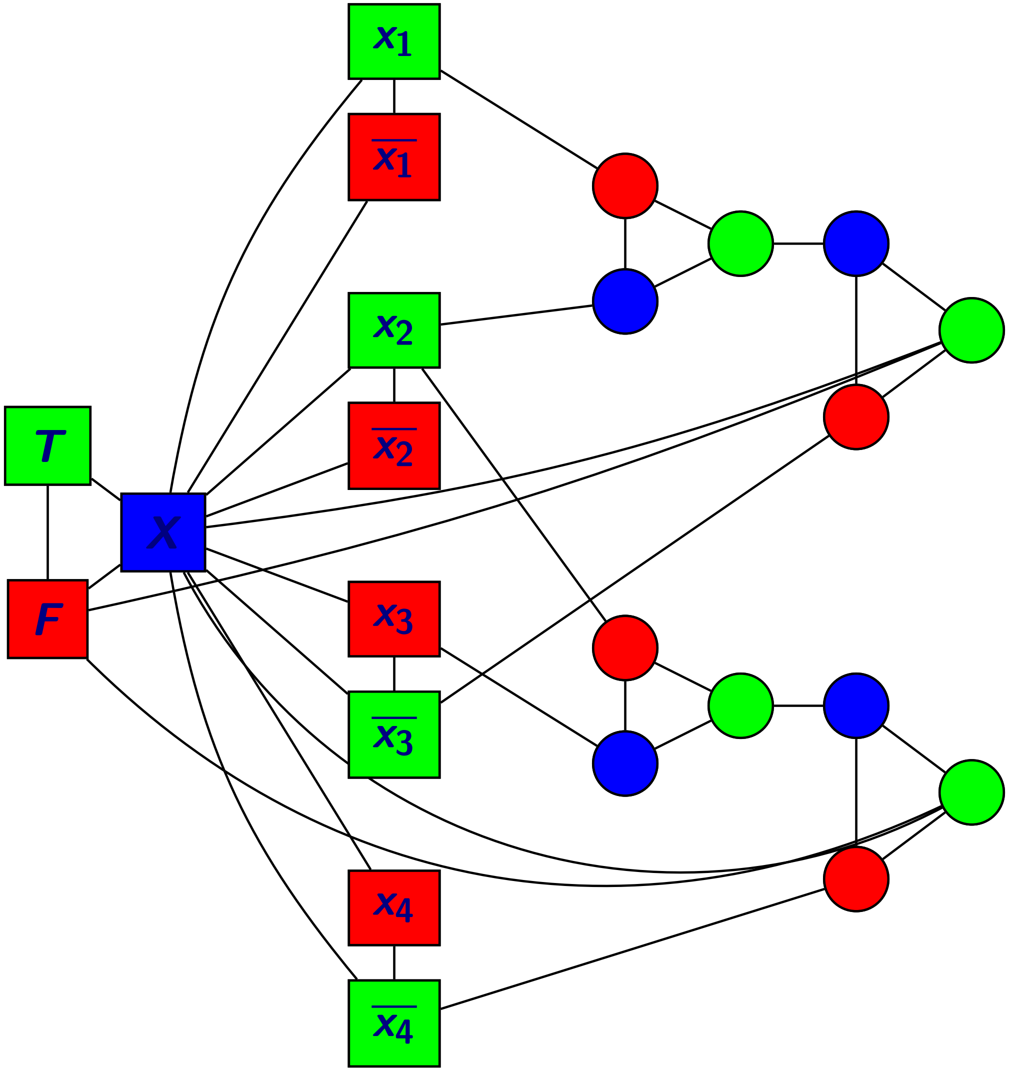
3SAT to 3COLOR: Picture

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



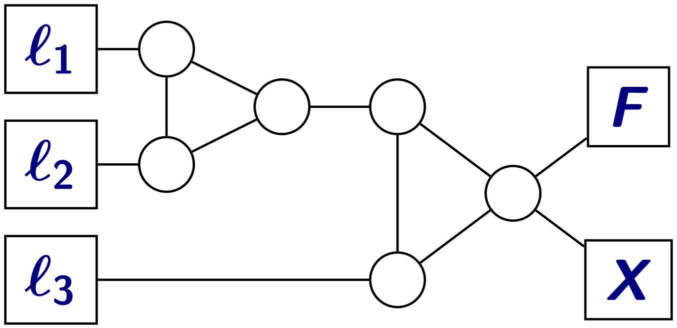
3SAT to 3COLOR: Picture

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



3SAT to 3COLOR: Only-If

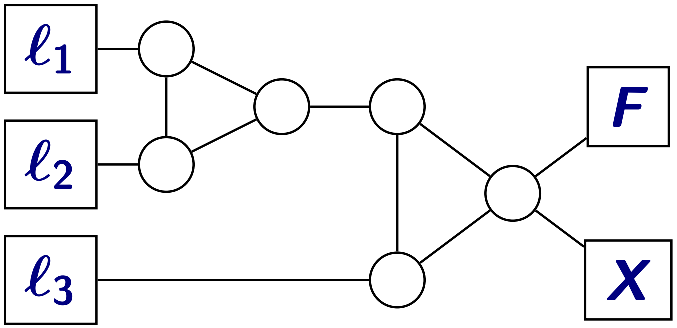
Let $f(\varphi) = G$, where for each clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, we include:



Claim: if φ is satisfiable, G is 3-colorable.

3SAT to 3COLOR: If

Let $f(\varphi) = G$, where for each clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$, we include:



Claim: if G is 3-colorable, φ is satisfiable.

Takeaway Points

Known ***NP***-complete languages

- SAT (from Cook-Levin)
- CNF-SAT (from Cook-Levin)
- 3SAT (from CNF-SAT)
- Independent Set (from 3SAT)
- Clique (from Independent Set)
- Vertex Cover (from Independent Set)
- 3-coloring (from 3SAT)