Algorithms - Spring '25

NP-Hardness (cont):
3SAT + graphs

Recap

- HW due today
- Reading due Wed.

Dfn: NP-Hard

X is NP-Hard
$\Longleftrightarrow$
If X could be solved in polynomial time, then P = NP.

So if <u>any</u> NP-Hard problem could be solved in polynomial time, then all of NP could be.

Note: Not <u>at all</u> obvious any such problem exists!

Cook-Levine Thm:

Circuit SAT is NP-Hard.

Proof (sketch):

Suppose I have an algorithm to solve CIRCUIT-SAT in poly time.

Take any problem in NP, A.

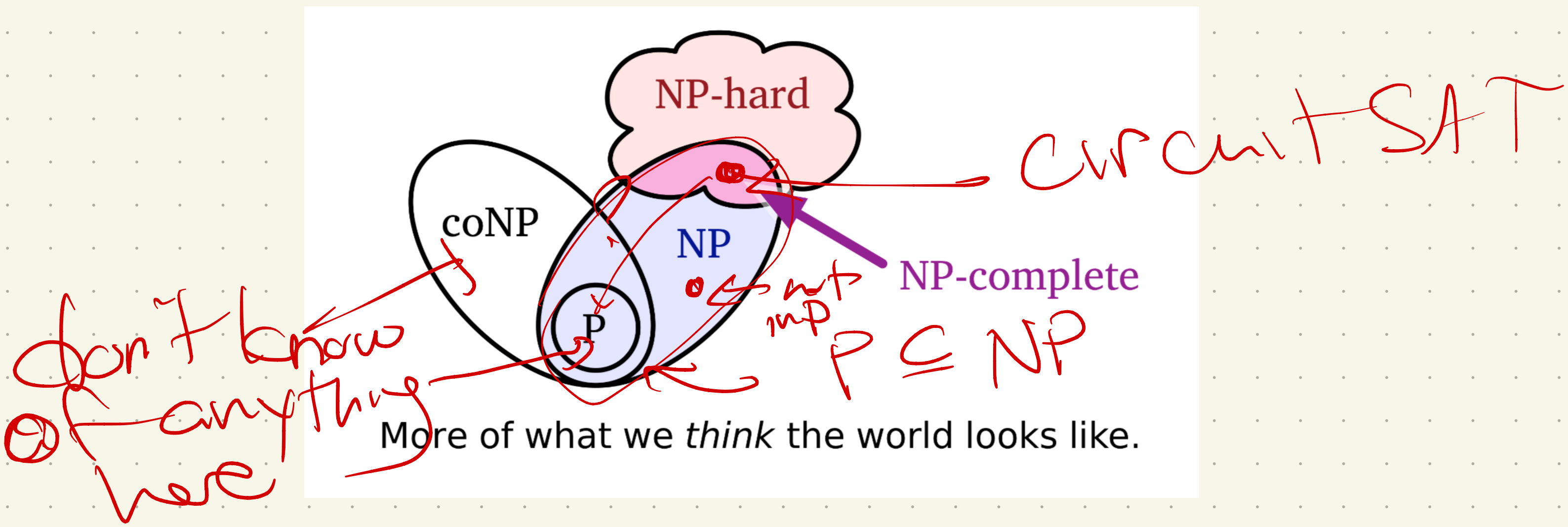
Reduce A to CIRCUIT-SAT in polynomial time: build circuit.

Therefore, I have a poly time alg for A.

A $\Rightarrow$ (convert) circuit ☒

So, there is at least one problem that is NP-Hard, & in NP, but which we don't think is in P:

is P=NP?



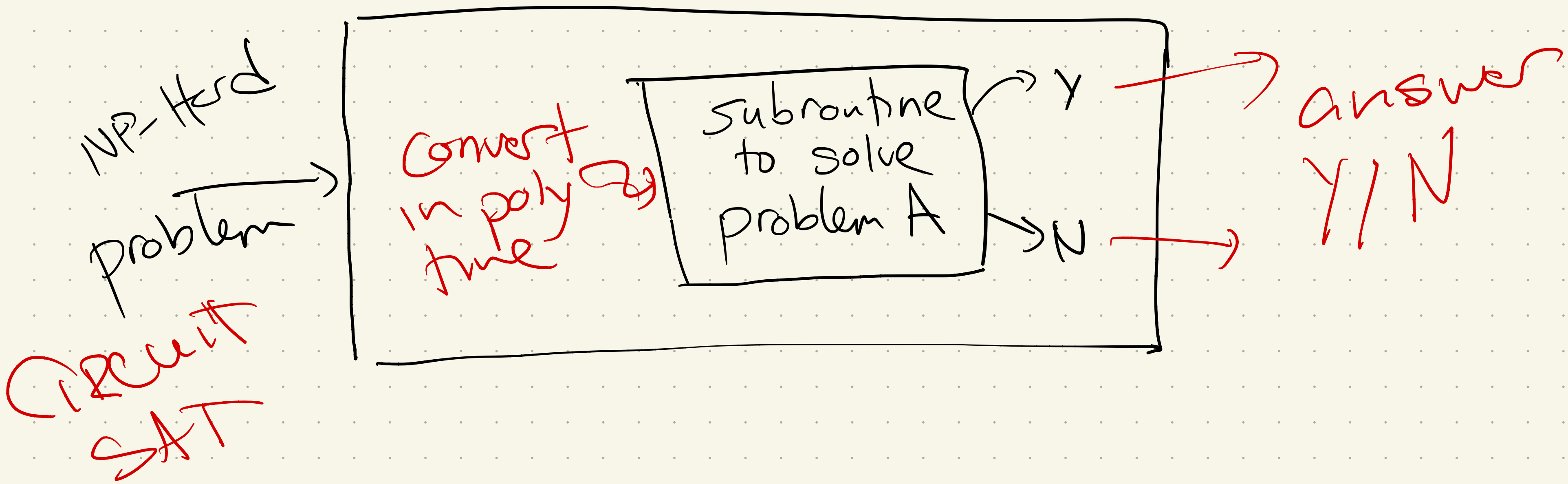
More of what we *think* the world looks like.

NP-Complete: NP-Hard & in NP

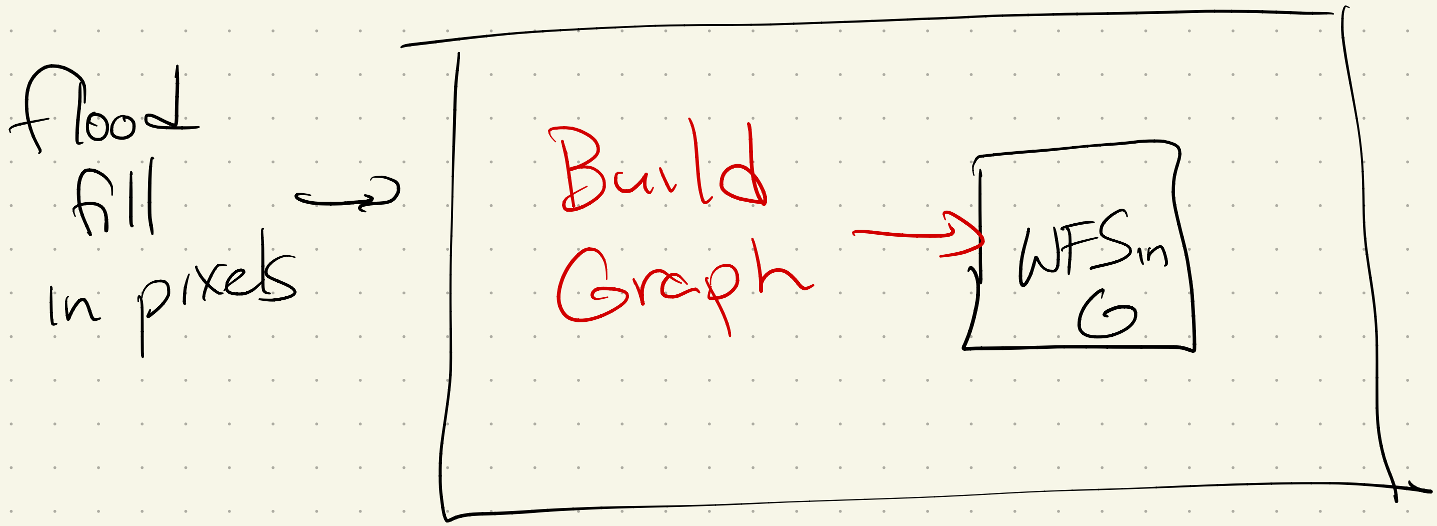
To prove NP-Hardness of A:

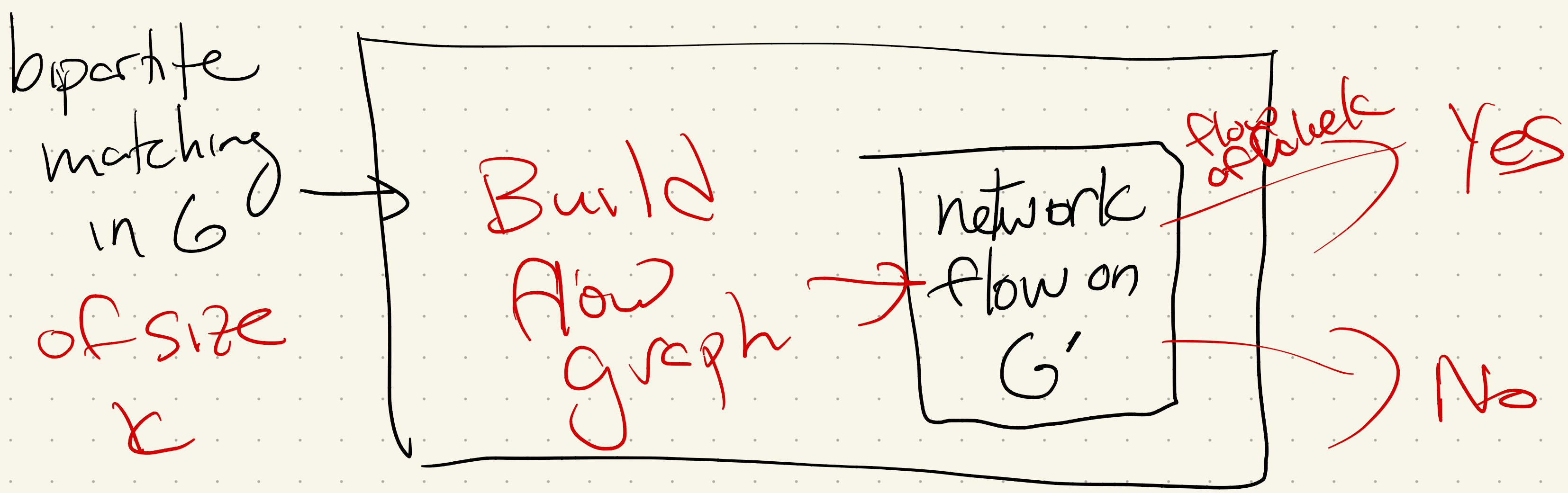
Reduce a known NP-Hard problem to A.

(Alternative is to show any problem in NP can be turned into A, like Cook.)



We've seen reductions!
But used them to solve problems!





This will feel odd, though:

To prove a new problem is hard, we'll show how we could solve a known hard problem using new problem as a subroutine.

<u>Why</u>? Just like halting problem!

Well, if a poly time algorithm existed, than you'd also be able to solve the hard problem!

(Therefore, "can't" be any such alg)

Other NP-Hard Problems:

SAT: Given a boolean formula, is there a way to assign inputs so result is 1?

Ex: $(a \vee b \vee c \vee \bar{d}) \Leftrightarrow ((b \wedge \bar{c}) \vee \overline{(\bar{a} \Rightarrow d)} \vee (c \neq a \wedge b)),$ ←

n variables, m clauses

First: in NP?

I claim answer is yes,

Certificate: assignment (T/F) to each variable

↳ size n

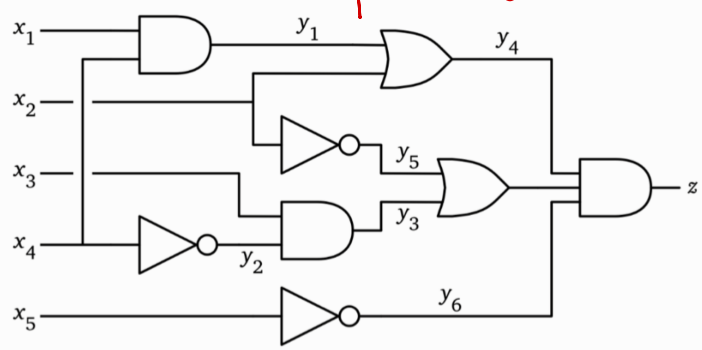
Can check in $O(n \cdot m)$ → truth tables!

Thm: SAT is NP-Hard.

pf: Reduce CIRCUIT SAT to SAT:

n inputs, m gates

Input: CIRCUIT



$$(y_1 = x_1 \wedge x_4) \wedge (y_2 = \overline{x_4}) \wedge (y_3 = x_3 \wedge y_2) \wedge (y_4 = y_1 \vee x_2) \wedge$$
$$(y_5 = \overline{x_2}) \wedge (y_6 = \overline{x_5}) \wedge (y_7 = y_3 \vee y_5) \wedge (z = y_4 \wedge y_7 \wedge y_6) \wedge z$$

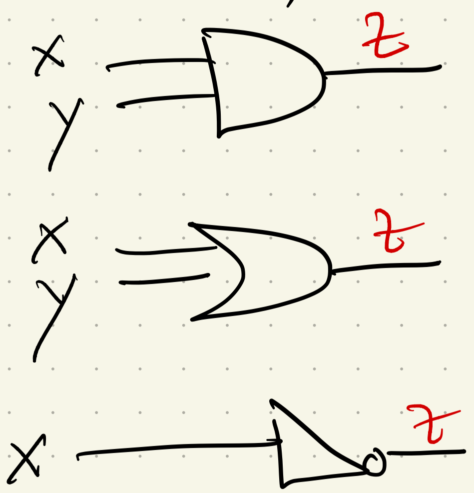
A boolean circuit with gate variables added, and an equivalent boolean formula.

Convert in poly time to clauses:

→ build a formula that is equivalent

More carefully:

1) For any gate, can transform:



$z = x \wedge y$

$z = x \vee y$

$z = \neg x$

2) "And" these together, + want final output true:

$m+1$ clauses

n variables

$O(mn)$ time to build SAT formula

Is this poly-size?

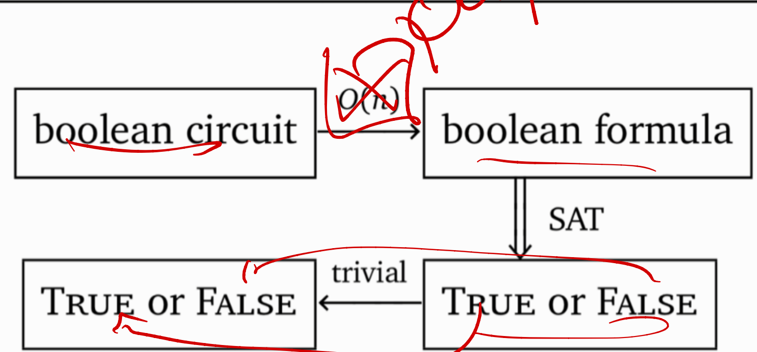
Given n inputs + m gates:

Variables: n

Clauses: $m+1$

time spent $\leq m \cdot n$

So our reduction:



$$T_{CSAT}(n) \leq O(n) + T_{SAT}(O(n)) \implies T_{SAT}(n) \geq T_{CSAT}(\Omega(n)) - O(n)$$

3SAT: 3CNF formulas:

3 variables OR-ed in each clause
"and" the clauses together

Thm: 3SAT is NP-Hard

pf: Reduce circuitSAT to 3SAT:

Need to show any circuit can be transformed
to 3CNF form

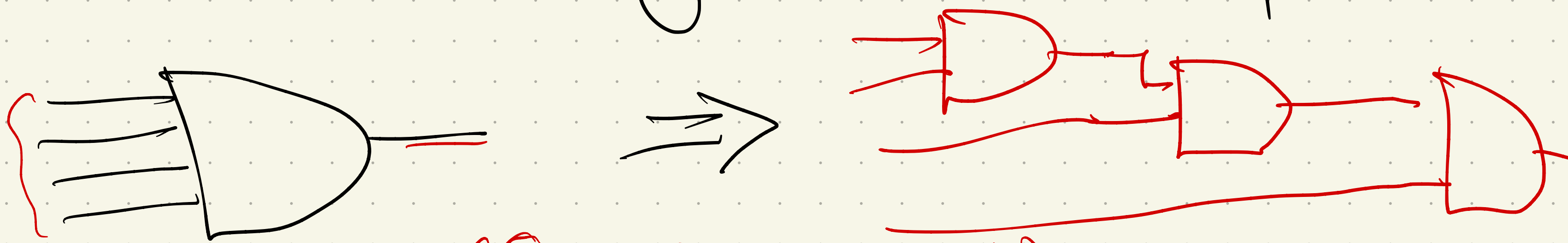
(so last reduction fails)

Instead ⟶

Given a circuit!

$z = x \wedge y$

① Rewrite so each gate has ≤2 inputs:



$a \wedge b \wedge c \wedge d = (((a \wedge b) \wedge c) \wedge d)$

② Write formula, like SAT. Only 3 types!

$$y = a \vee b$$
$$y = a \wedge b$$
$$y = \overline{a}$$

③ Now, change to CNF:
go back to truth tables

$$\begin{aligned} a = b \wedge c &\longmapsto (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b) \wedge (\bar{a} \vee c) \\ a = b \vee c &\longmapsto (\bar{a} \vee b \vee c) \wedge (a \vee \bar{b}) \wedge (a \vee \bar{c}) \\ a = \bar{b} &\longmapsto (a \vee b) \wedge (\bar{a} \vee \bar{b}) \end{aligned}$$

4

$a = b \wedge c$

a	b	c
T	T	T
T	T	F
T	F	F

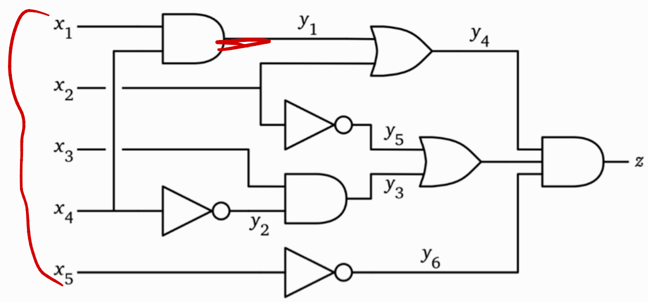
④ Now, need 3 per clause!

$$\begin{aligned} a &\longmapsto (a \vee x \vee y) \wedge (a \vee \bar{x} \vee y) \wedge (a \vee x \vee \bar{y}) \wedge (a \vee \bar{x} \vee \bar{y}) \\ a \vee b &\longmapsto (a \vee b \vee x) \wedge (a \vee b \vee \bar{x}) \end{aligned}$$

a	x	y
T		
T		

T

Note: Bigger!



$$(y_1 = x_1 \wedge x_4) \wedge (y_2 = \overline{x_4}) \wedge (y_3 = x_3 \wedge y_2) \wedge (y_4 = y_1 \vee x_2) \wedge$$
$$(y_5 = \overline{x_2}) \wedge (y_6 = \overline{x_5}) \wedge (y_7 = y_3 \vee y_5) \wedge (z = y_4 \wedge y_7 \wedge y_6) \wedge z$$

A boolean circuit with gate variables added, and an equivalent boolean formula.

How much bigger? (need polynomial)

⇓

$$(y_1 \vee \overline{x_1} \vee \overline{x_4}) \wedge (\overline{y_1} \vee x_1 \vee z_1) \wedge (\overline{y_1} \vee x_1 \vee \overline{z_1}) \wedge (\overline{y_1} \vee x_4 \vee z_2) \wedge (\overline{y_1} \vee x_4 \vee \overline{z_2})$$
$$\wedge (y_2 \vee x_4 \vee z_3) \wedge (y_2 \vee x_4 \vee \overline{z_3}) \wedge (\overline{y_2} \vee \overline{x_4} \vee z_4) \wedge (\overline{y_2} \vee \overline{x_4} \vee \overline{z_4})$$
$$\wedge (y_3 \vee \overline{x_3} \vee \overline{y_2}) \wedge (\overline{y_3} \vee x_3 \vee z_5) \wedge (\overline{y_3} \vee x_3 \vee \overline{z_5}) \wedge (\overline{y_3} \vee y_2 \vee z_6) \wedge (\overline{y_3} \vee y_2 \vee \overline{z_6})$$
$$\wedge (\overline{y_4} \vee y_1 \vee x_2) \wedge (y_4 \vee \overline{x_2} \vee z_7) \wedge (y_4 \vee \overline{x_2} \vee \overline{z_7}) \wedge (y_4 \vee \overline{y_1} \vee z_8) \wedge (y_4 \vee \overline{y_1} \vee \overline{z_8})$$
$$\wedge (y_5 \vee x_2 \vee z_9) \wedge (y_5 \vee x_2 \vee \overline{z_9}) \wedge (\overline{y_5} \vee \overline{x_2} \vee z_{10}) \wedge (\overline{y_5} \vee \overline{x_2} \vee \overline{z_{10}})$$
$$\wedge (y_6 \vee x_5 \vee z_{11}) \wedge (y_6 \vee x_5 \vee \overline{z_{11}}) \wedge (\overline{y_6} \vee \overline{x_5} \vee z_{12}) \wedge (\overline{y_6} \vee \overline{x_5} \vee \overline{z_{12}})$$
$$\wedge (\overline{y_7} \vee y_3 \vee y_5) \wedge (y_7 \vee \overline{y_3} \vee z_{13}) \wedge (y_7 \vee \overline{y_3} \vee \overline{z_{13}}) \wedge (y_7 \vee \overline{y_5} \vee z_{14}) \wedge (y_7 \vee \overline{y_5} \vee \overline{z_{14}})$$
$$\wedge (y_8 \vee \overline{y_4} \vee \overline{y_7}) \wedge (\overline{y_8} \vee y_4 \vee z_{15}) \wedge (\overline{y_8} \vee y_4 \vee \overline{z_{15}}) \wedge (\overline{y_8} \vee y_7 \vee z_{16}) \wedge (\overline{y_8} \vee y_7 \vee \overline{z_{16}})$$
$$\wedge (y_9 \vee \overline{y_8} \vee \overline{y_6}) \wedge (\overline{y_9} \vee y_8 \vee z_{17}) \wedge (\overline{y_9} \vee y_8 \vee \overline{z_{17}}) \wedge (\overline{y_9} \vee y_6 \vee z_{18}) \wedge (\overline{y_9} \vee y_6 \vee \overline{z_{18}})$$
$$\wedge (y_9 \vee z_{19} \vee z_{20}) \wedge (y_9 \vee \overline{z_{19}} \vee z_{20}) \wedge (y_9 \vee z_{19} \vee \overline{z_{20}}) \wedge (y_9 \vee \overline{z_{19}} \vee \overline{z_{20}})$$

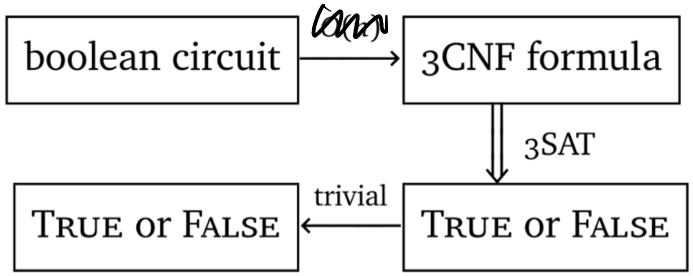
Each gate ⇓ ≤ 12 clauses

$m \hookrightarrow 12m$ clauses

3SAT variables: n circuit variables $\rightarrow n + m + 2m$

So:

size? $O(mn)$



$$T_{CSAT}(n) \le O(n) + T_{3SAT}(O(n)) \implies T_{3SAT}(n) \ge T_{CSAT}(\Omega(n)) - O(n)$$

poly

So: if could solve 3CNF, could solve CIRCUITSAT in poly time.

Historical note:

Why booleen functions?

(Think like a computer engineer for a moment...)

Next:

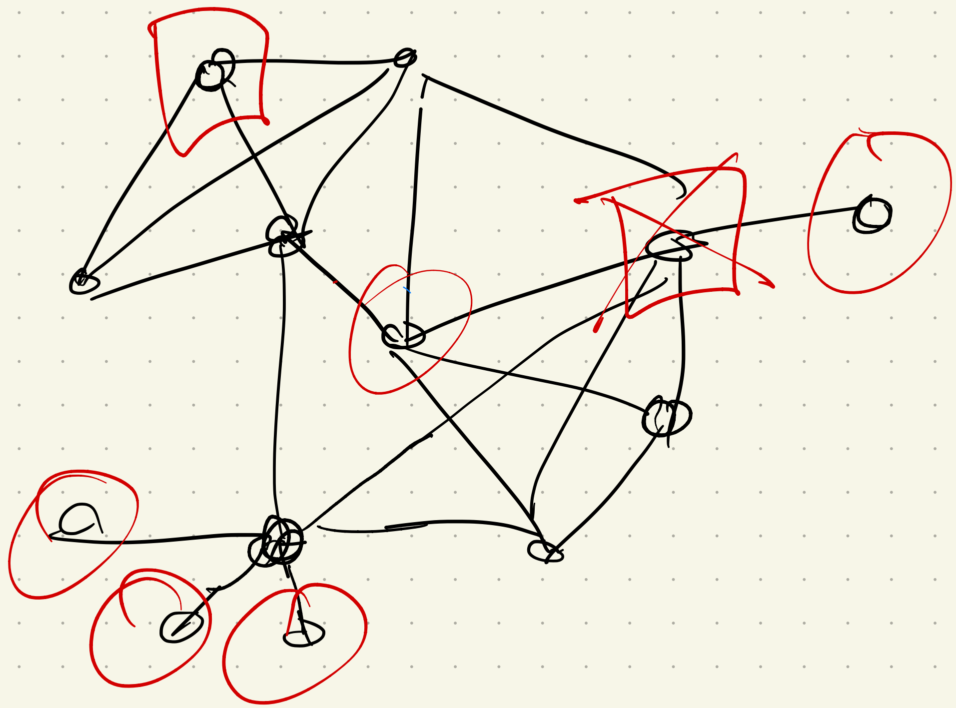
Can we do this with any useful problems?

(Logic is all well + good...)

Maybe → graphs?

Independent Set:

A set of vertices in a graph with no edges between them:

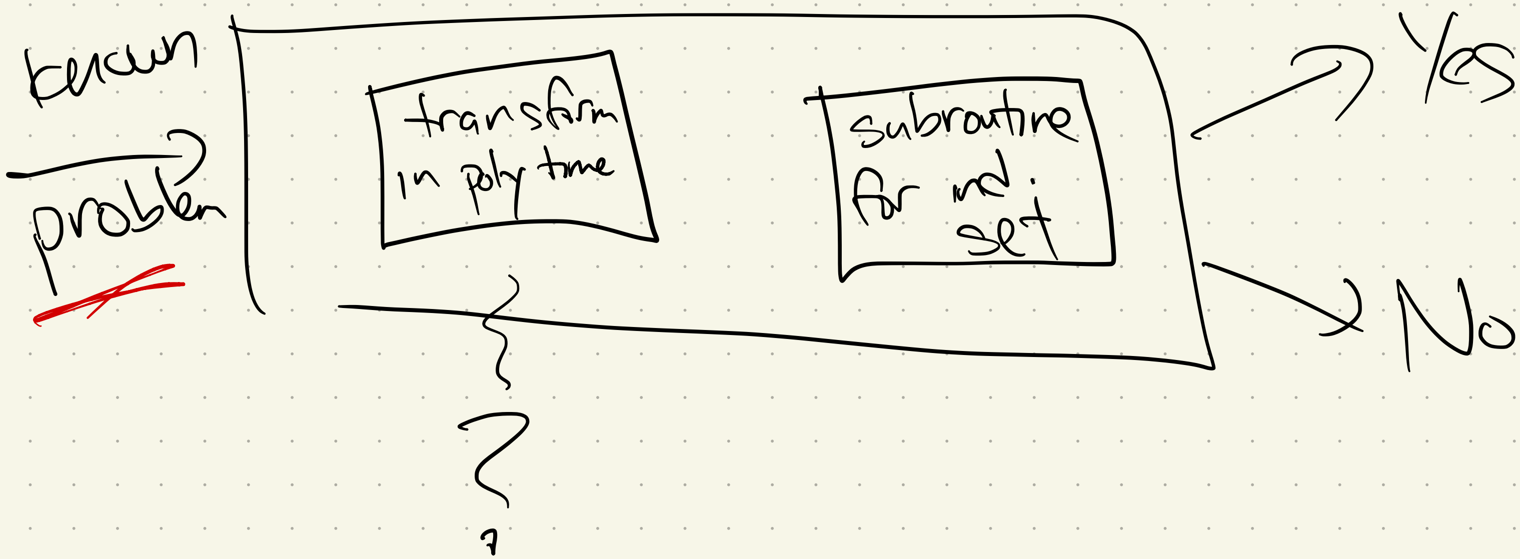


Decision version:

Given G & $k \in \mathbb{Z}$ does G have indep set of size $\geq k$?

Challenge: No booleans!

But reduction needs to take known NP-Hard problem + build a graph!



We'll use 3SAT

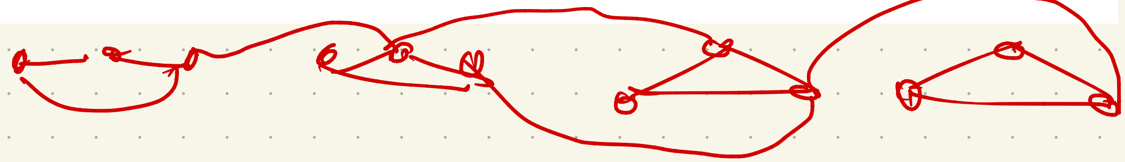
(but stop and marvel a bit first...)

Reduction:

Input is 3CNF boolean formula

$$(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$$

V:



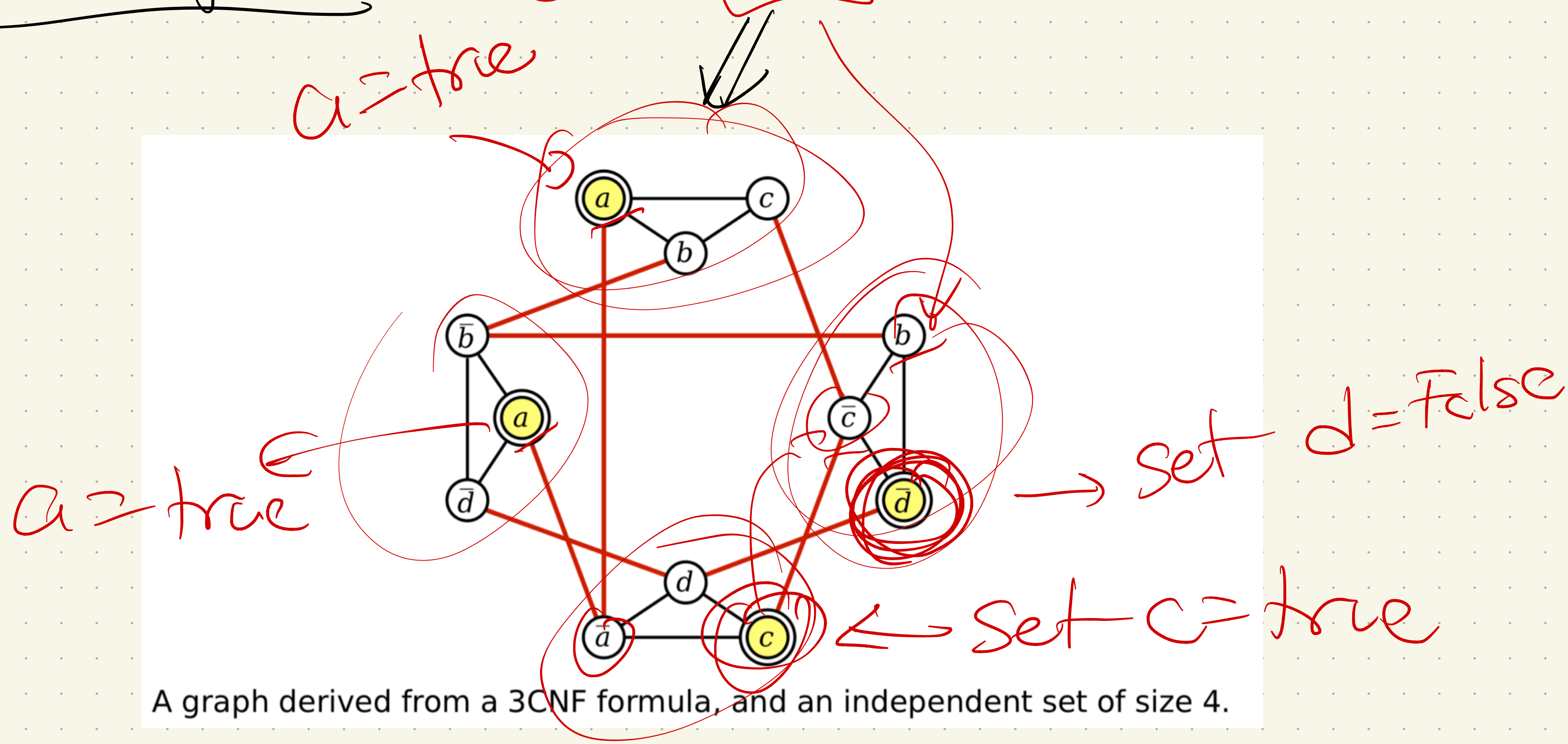
① Make a vertex for each literal in each clause

② Connect two vertices if:

- they are in same clause ✓
- they are a variable + its <u>inverse</u>

Example :

$(a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b} \lor \bar{d})$

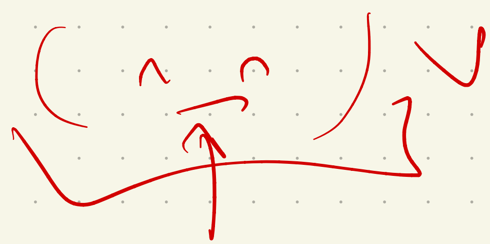


A graph derived from a 3CNF formula, and an independent set of size 4.

Claim:

formula is satisfiable $\Longleftrightarrow$ G has independent set of size $\geq m$

Spps formula is satisfiable: at least one variable per clause is true



Pick one
↳ add corresponding vertex to IS; one per clause

Can't have edge b/t clause variables since such an edge would mean both x & $\bar{x}$ are true.

$\Rightarrow$ m vertices, no edges

↳ IS so G, m is true

$\Leftarrow$: take IS in G of size $\geq m$

Can have at most one vertex per $\triangle$ (b/c pigeonhole)

so $\Rightarrow$ exactly one vertex in IS

Set that variable = T.

(set negations = F)

Rest of variables → either T/F.

One variable per clause
is now true

& no variable & its negation
are both true

⟹ formula is satisfied

Next: Graph Coloring

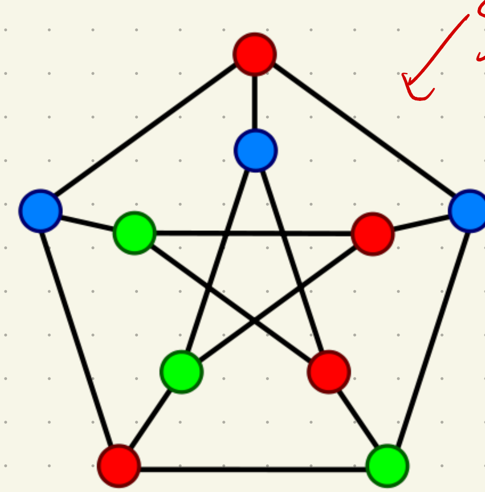
A **k-coloring** of a graph G is a map:

$$c : V \to \{1, \ldots, k\}$$

that assigns one of k "colors" to each vertex so that every edge has 2 different colors at its endpoints

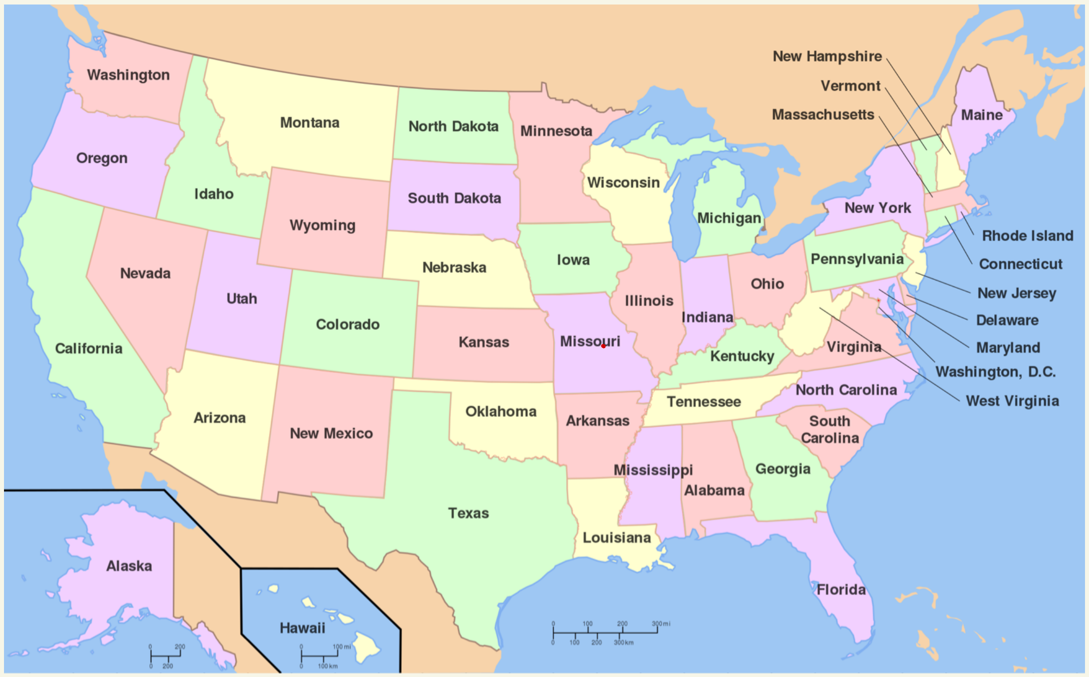
3-colorable

Goal: Use few colors



$k = V$ easy!

Aside: this is famous!
Ever heard of map coloring?



Famous theorem: 4 color theorem

Thm: 3-colorability is NP-Complete.

(Decision version: Given G & k, output yes/no)

In NP:

certificate: color for each vertex

To check:

loop over every edge & verify endpoints have different colors

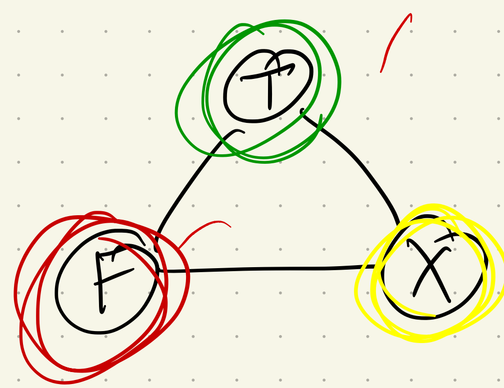
NP-Hard:

Reduction from 3SAT.

Given formula for 3SAT Φ, we'll make a graph G_Φ.

Φ will be satisfiable $\iff G_\Phi$ can be 3-colored.

Key notion: Build "gadgets"!

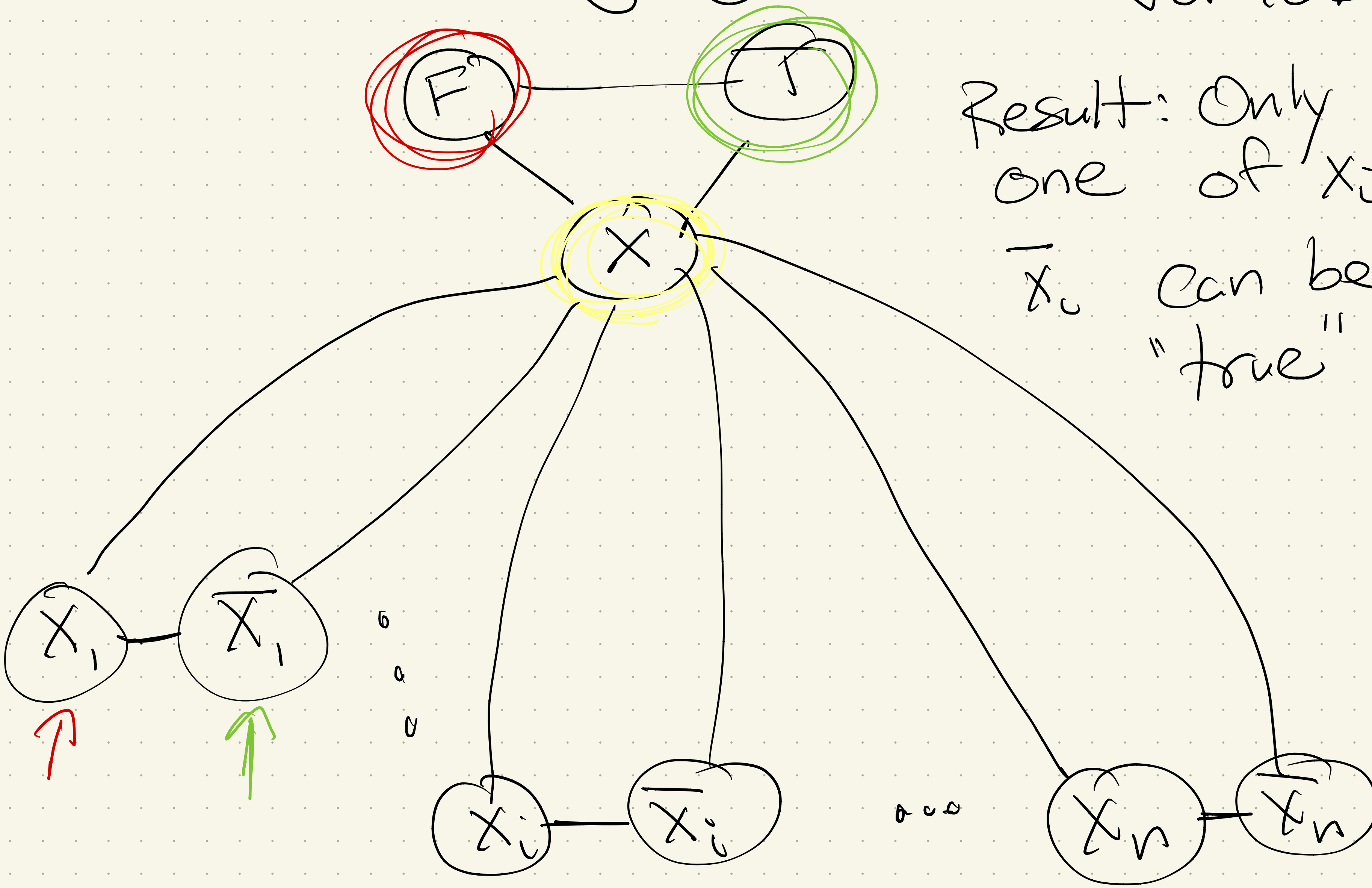


① Truth gadget –

Must use 3 colors –

so establishes a "true" color.

(2) Variable gadget: one per variable

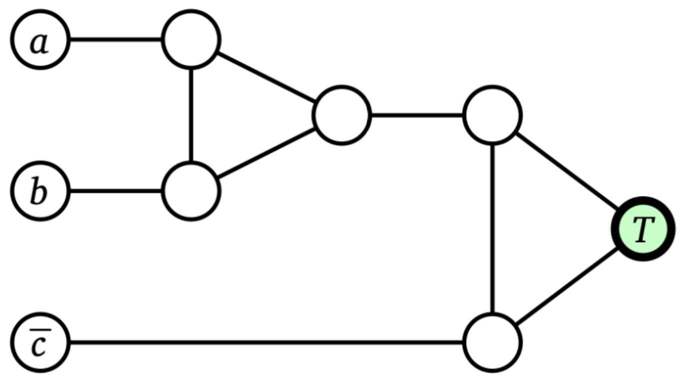


Result: Only one of x_i + $\overline{x}_i$ can be "true"

(3) Clause gadget:

For each clause, join 3 of the variable vertices to the "true" vertex from the truth gadget.

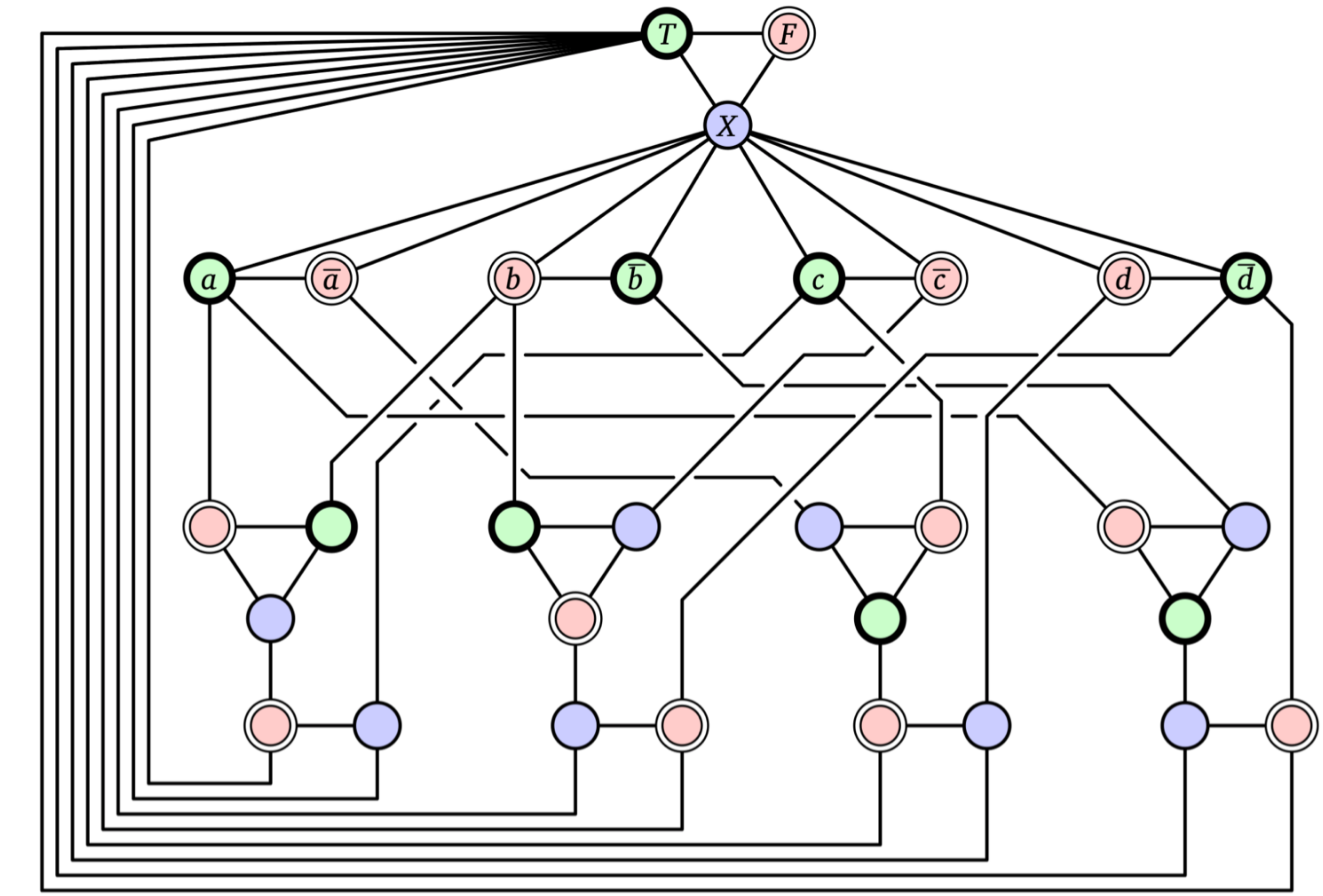
<u>Goal</u>: If all 3 are false, no valid 3-coloring



A clause gadget for $(a \lor b \lor \bar{c})$.

Why?? try to color all "false"

Final reduction image:



A 3-colorable graph derived from the satisfiable 3CNF formula
$(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

Now, need reduction proof:

3 colorings of G_Φ $\Longleftrightarrow$ Φ is satisfiable

<u>Pf:</u>

$\Rightarrow$: Consider a 3-coloring of G:

$\Leftarrow$ Consider a satisfying assignment to Φ!