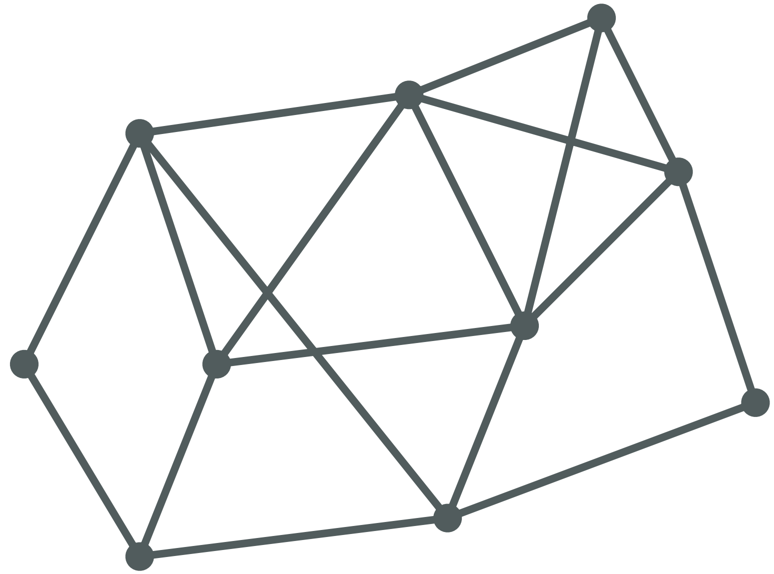
Graph Coloring & Independent Set

Graph Coloring



Goal: color vertices so that no adjacent vertices have the same color, w/ minimum # of colors.

decision version: does k colors suffice?

In "k-coloring problem", k is a fixed constant

Theorem: 3-coloring is NP-hard.

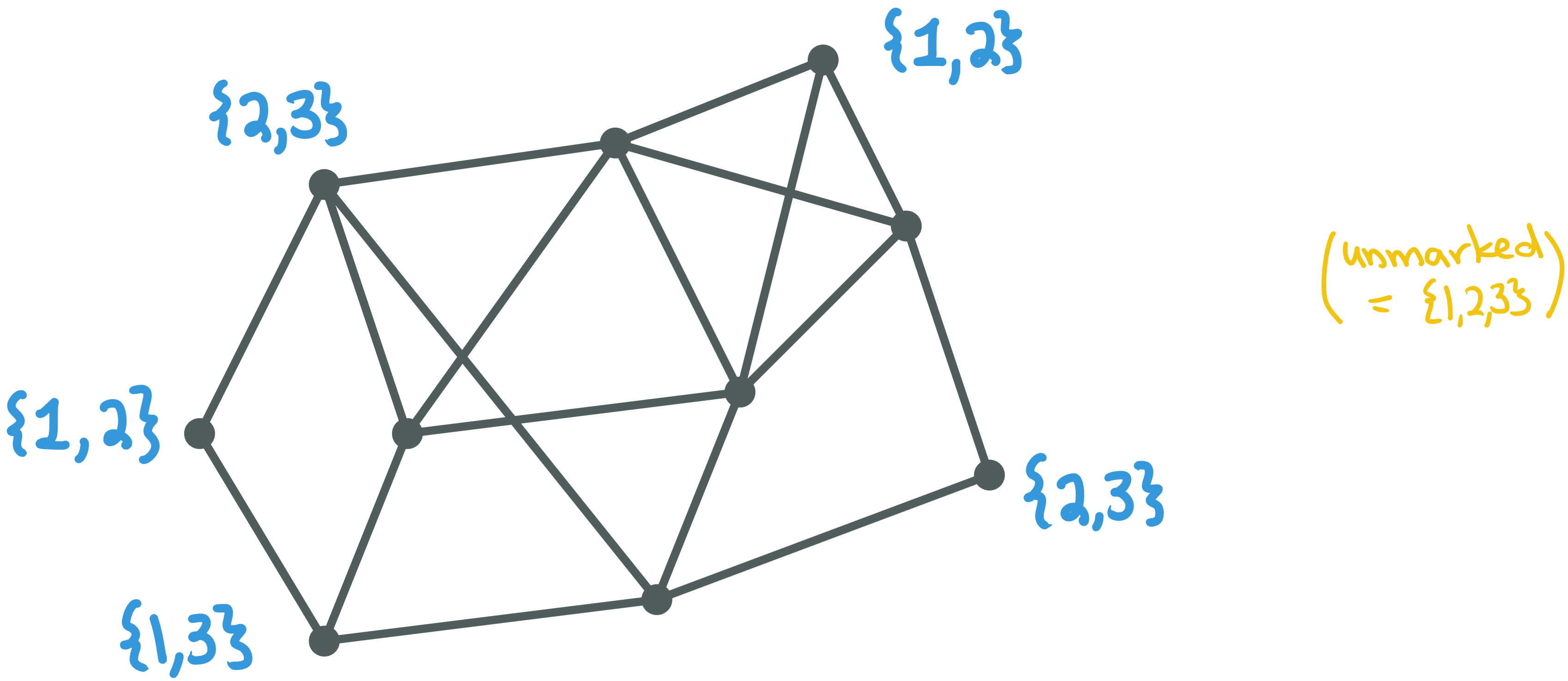
Theorem: 3-coloring is NP-hard.

High-level approach:

3SAT → "subset 3-coloring" → 3-coloring

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

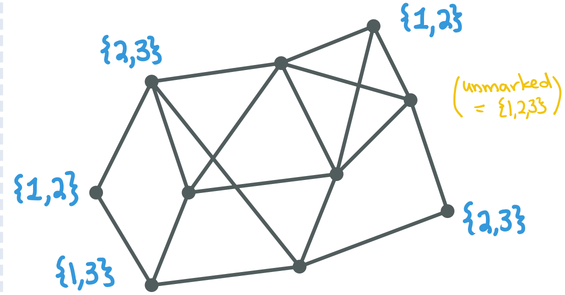


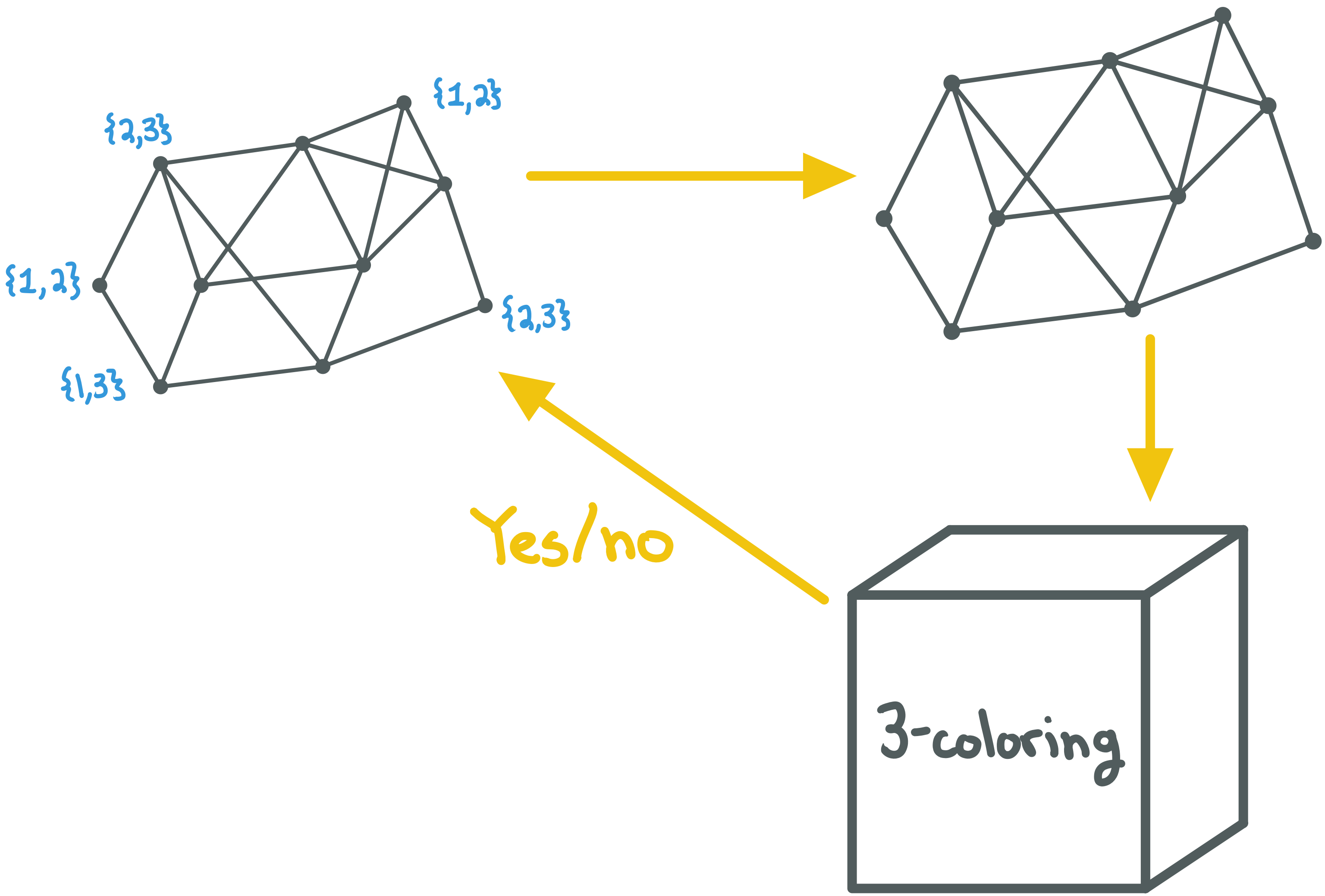
3SAT → "subset 3-coloring" → 3-coloring

claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

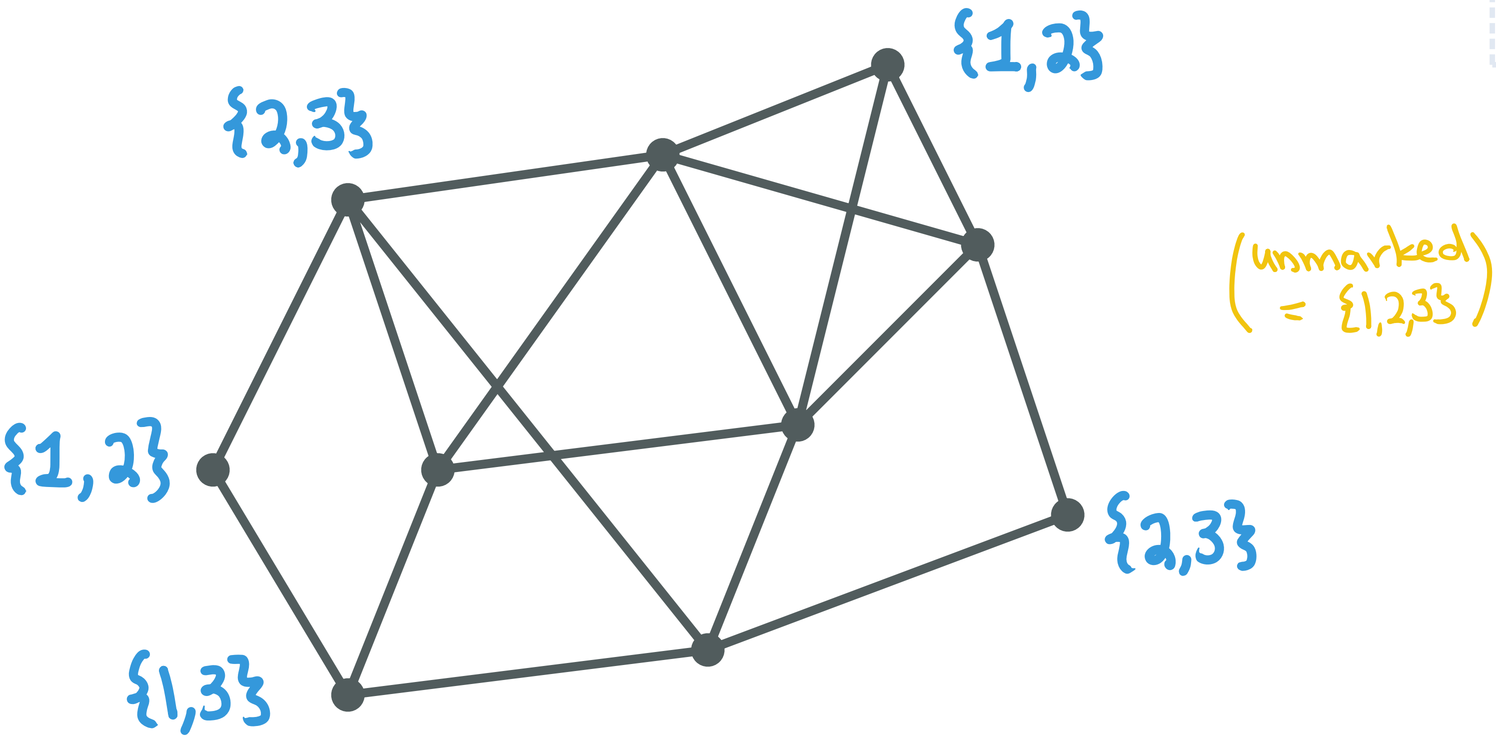




3SAT → "subset 3-coloring" → 3-coloring

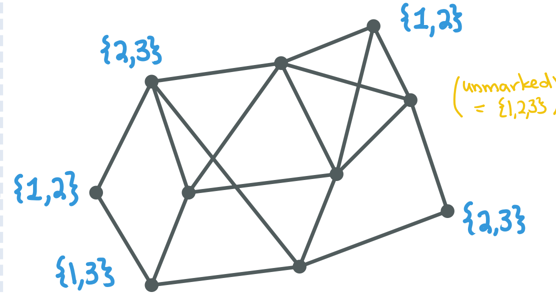
claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

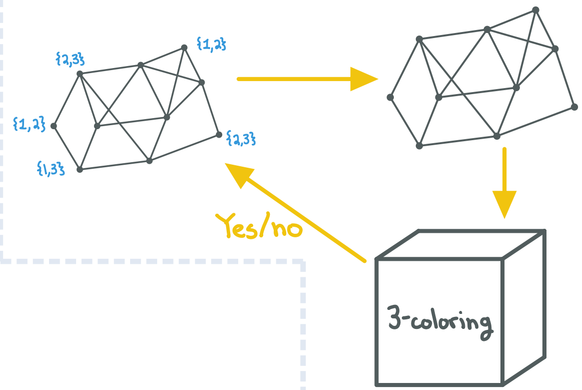
ideas?



Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

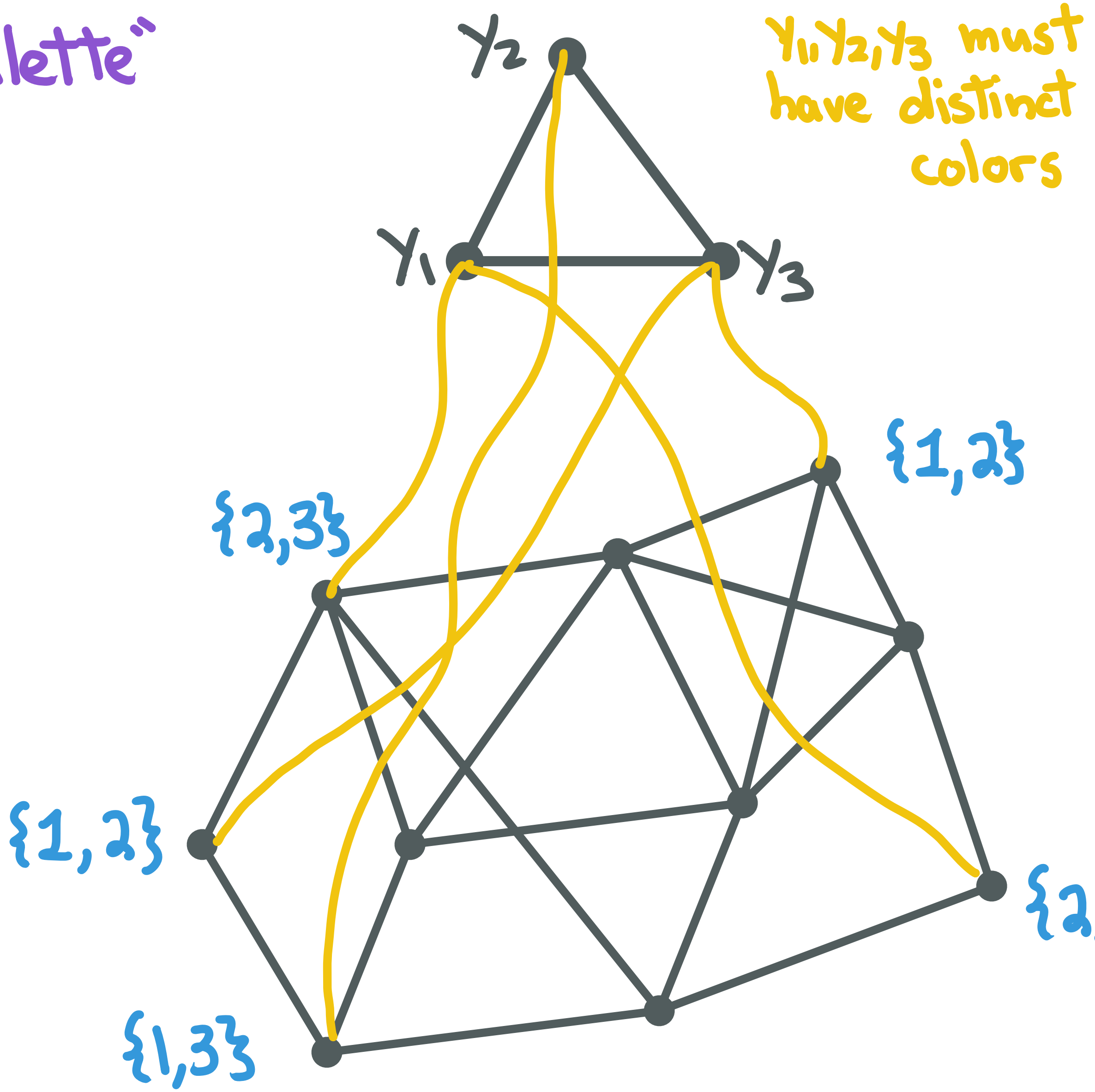




3SAT → "subset 3-coloring" → 3-coloring

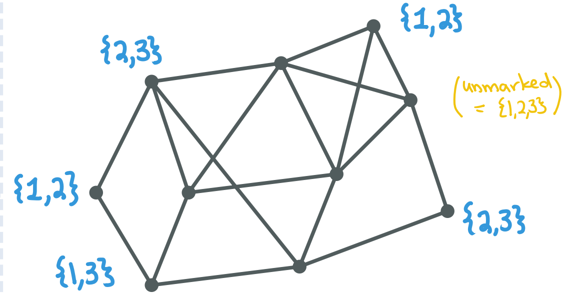
claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

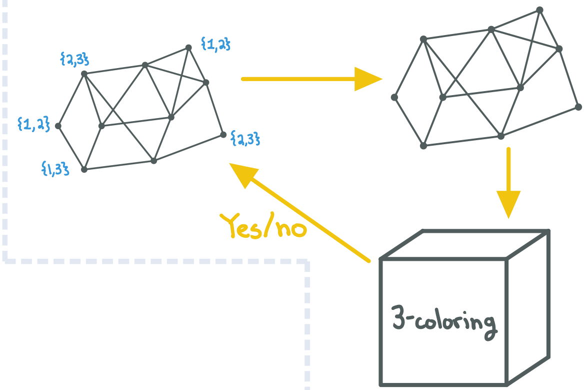
"color palette"



Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.





3SAT → "subset 3-coloring" → 3-coloring

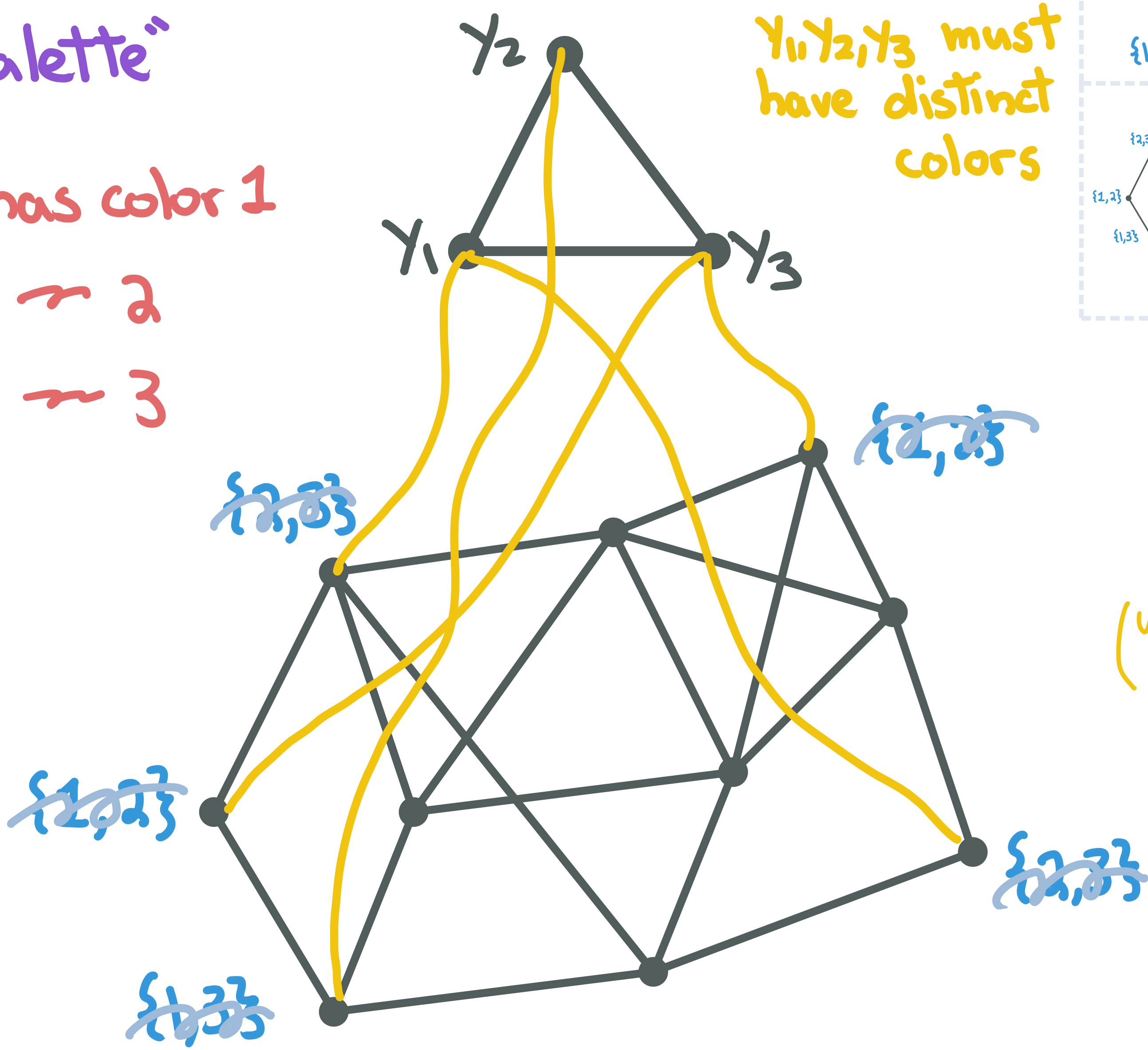
claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

"color palette"

wlog y_1 has color 1

$y_2 \rightsquigarrow 2$

$y_3 \rightsquigarrow 3$

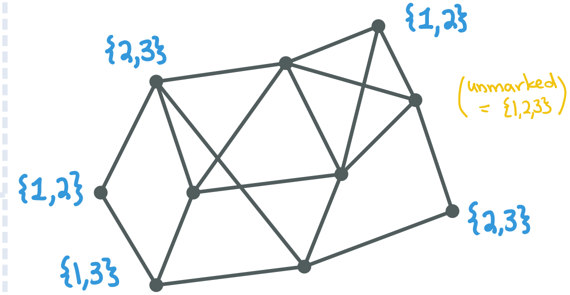


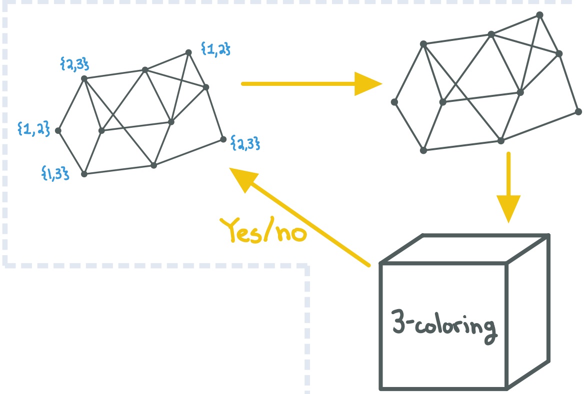
y_1, y_2, y_3 must have distinct colors

(unmarked) = {1,2,3}

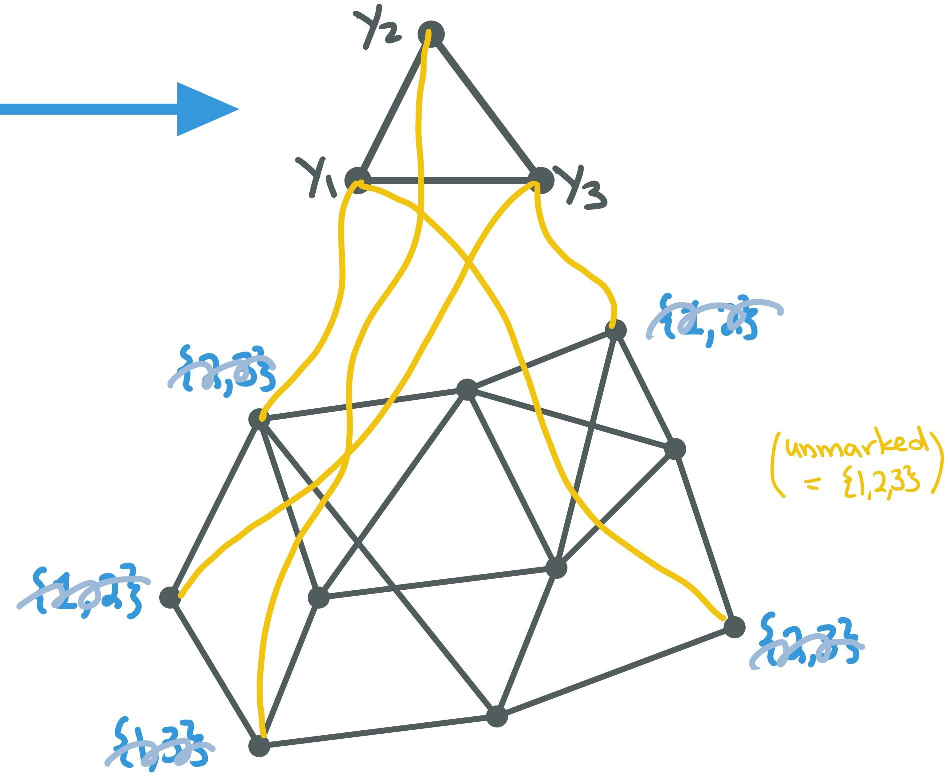
Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

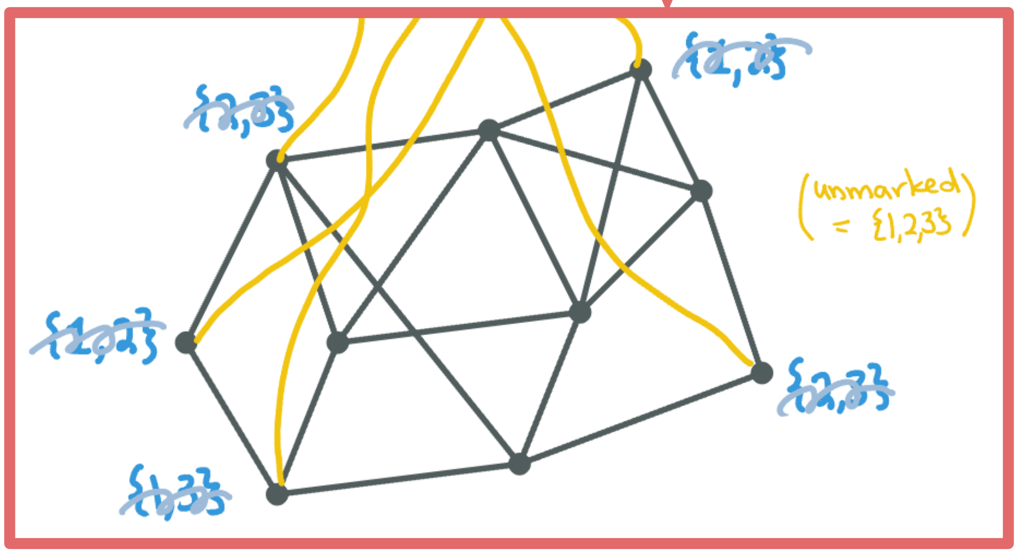




3-coloring on



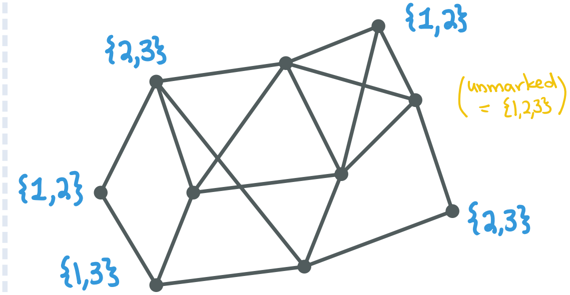
⇒ subset 3-coloring on



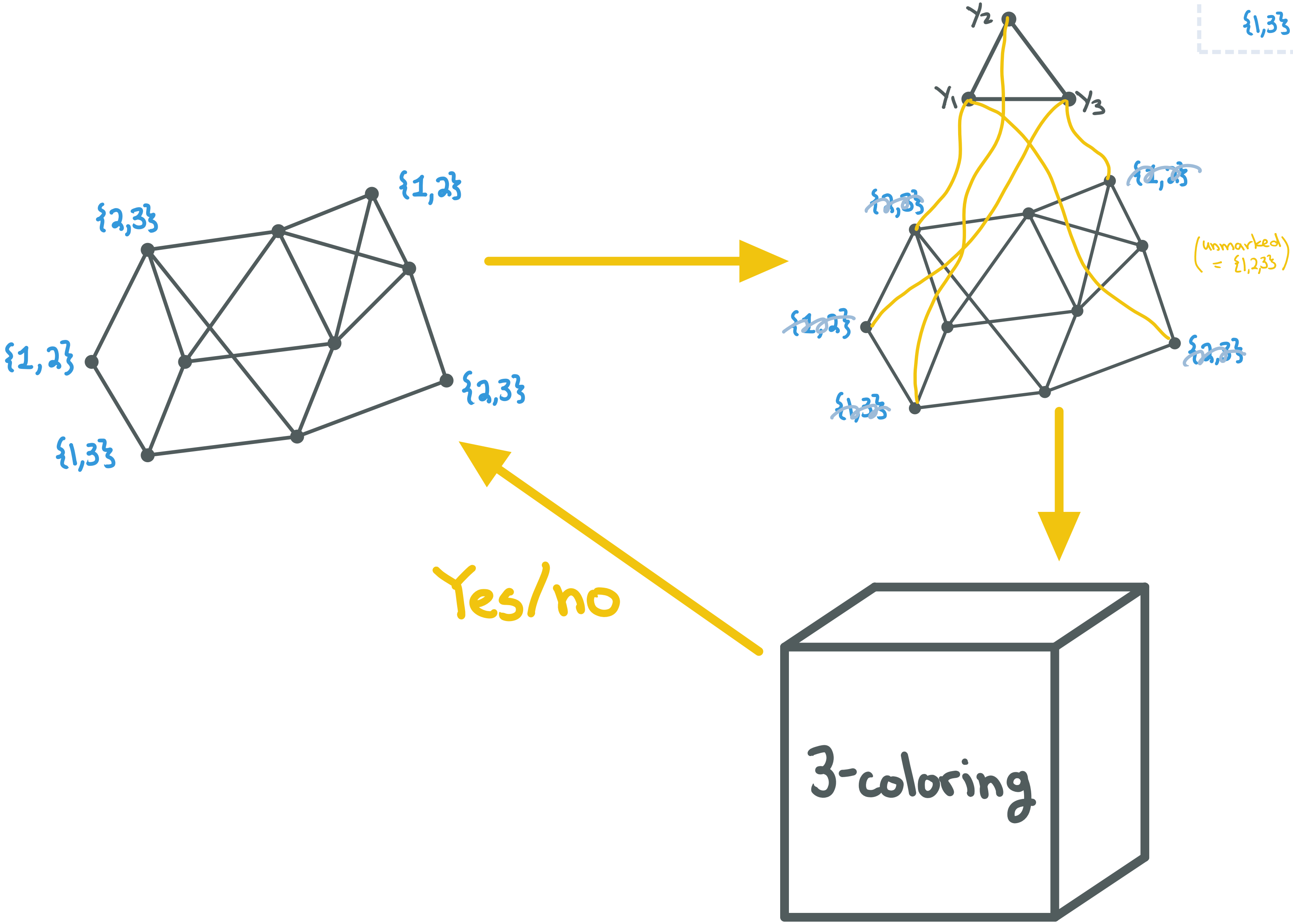
3SAT → "subset 3-coloring" → 3-coloring

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

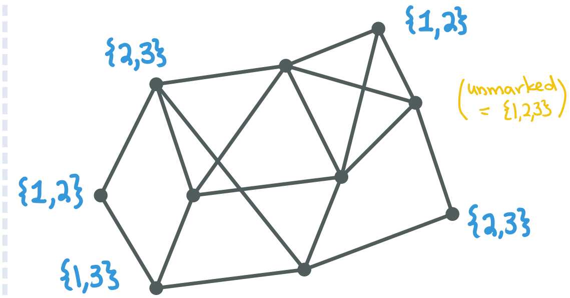


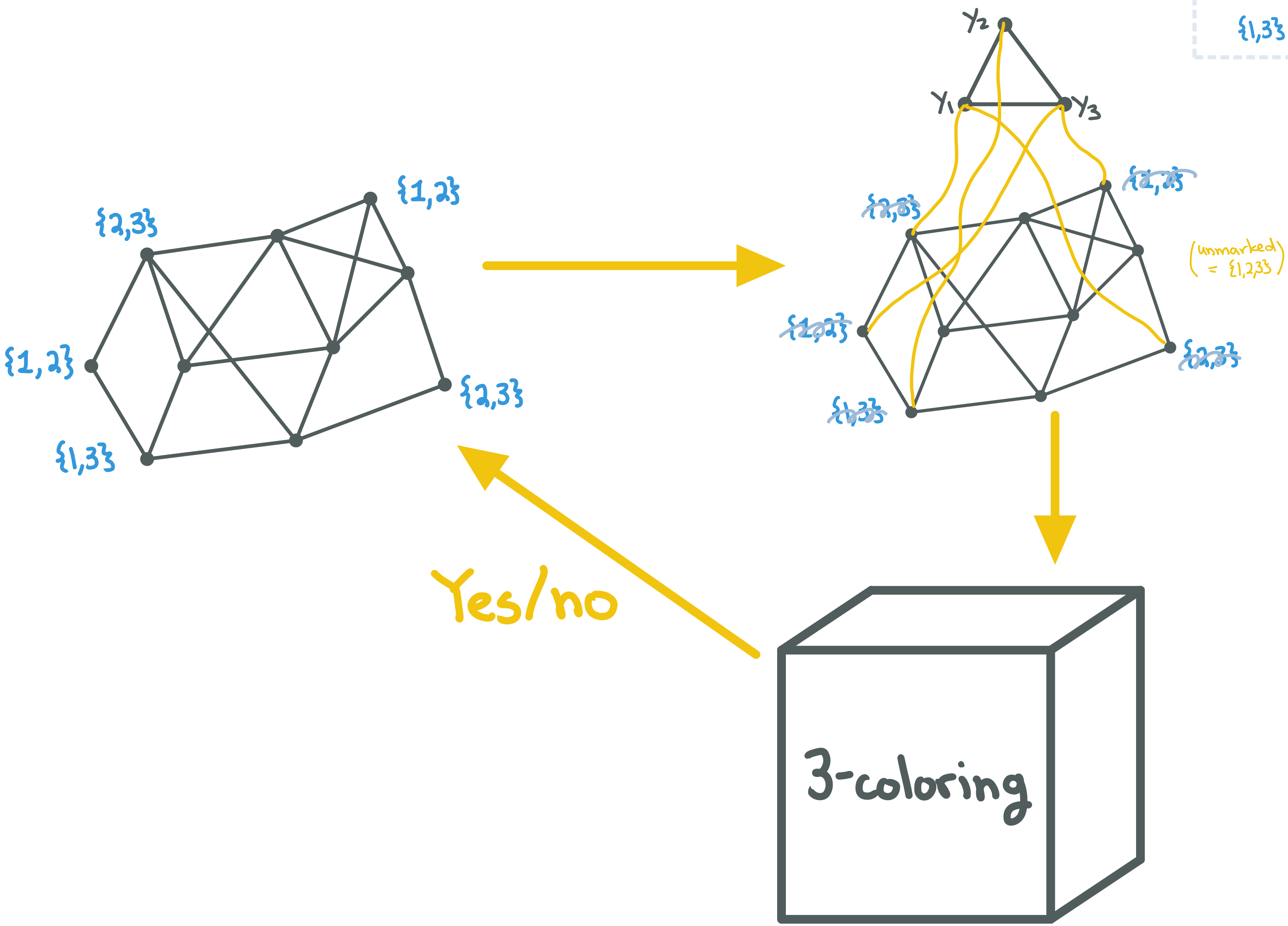
3SAT → "subset 3-coloring" → 3-coloring

claim: polytime algo for 3-coloring implies polytime algo for subset 3-coloring.

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



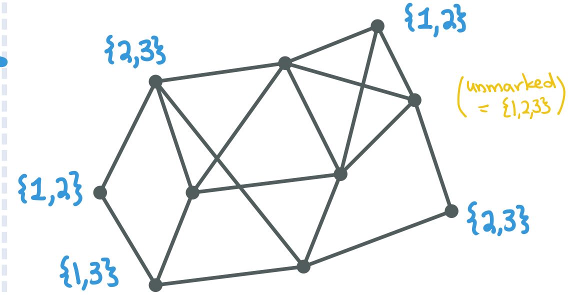


3SAT → "subset 3-coloring" → 3-coloring

claim: polytime algo for subset 3-coloring implies polytime algo for 3-SAT

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

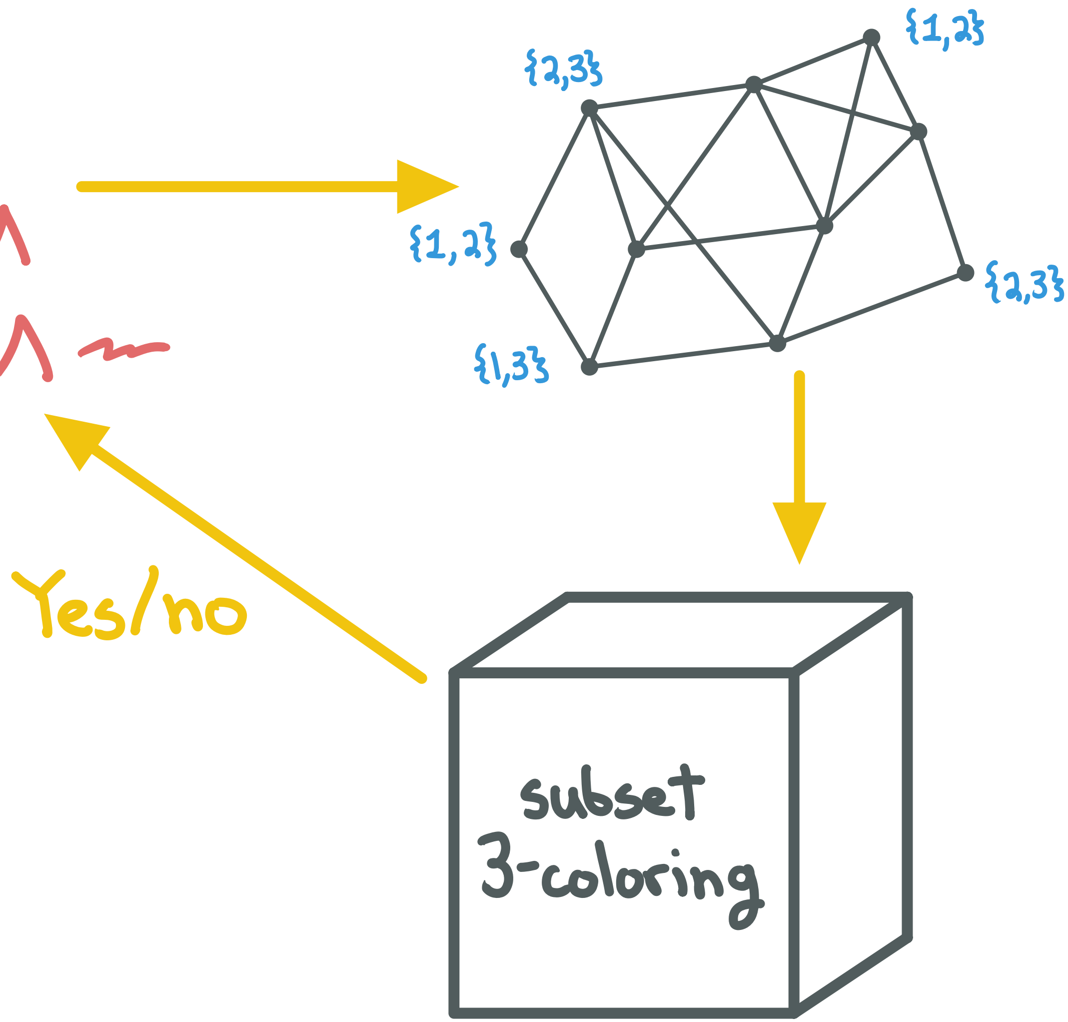


3-SAT

$f(x_1, \ldots, x_n) =$

$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$

$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \ldots$

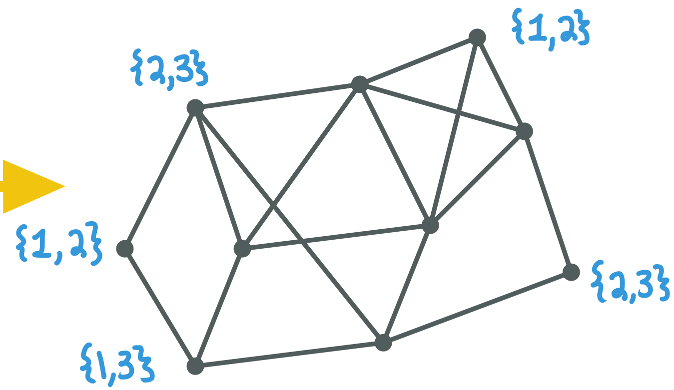


3-SAT

$f(x_1, \ldots, x_n) =$

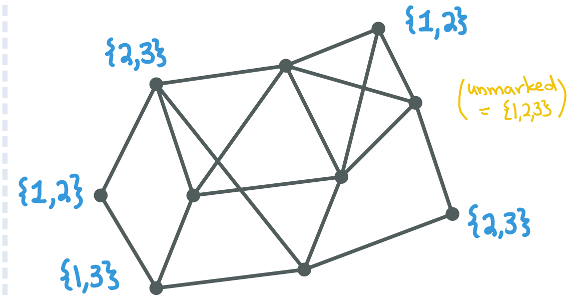
$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$

$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \ldots$



Subset 3-coloring

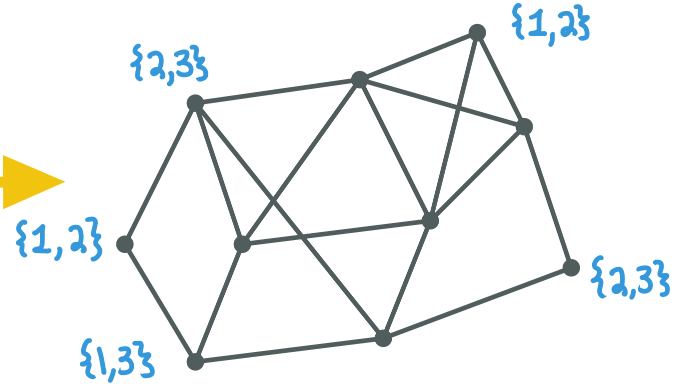
for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



ideas?

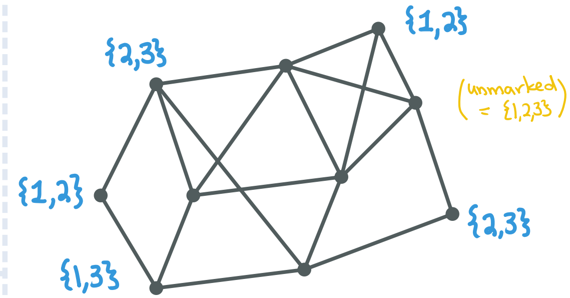
3-SAT

$f(x_1, \dots, x_n) =$
$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$
$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \dots$



Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



Convenient to name colors:

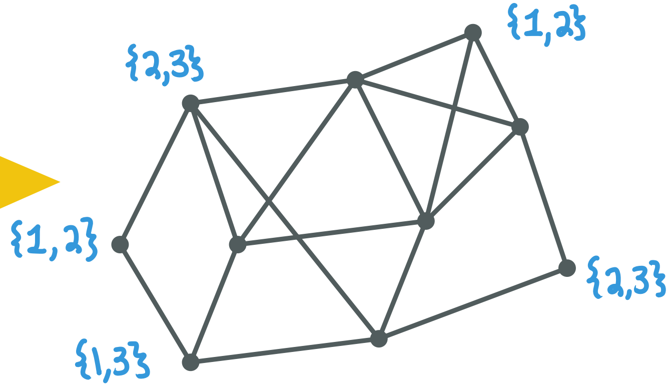
"T" = "True"

"F" = "False"

"N" = "Neither"

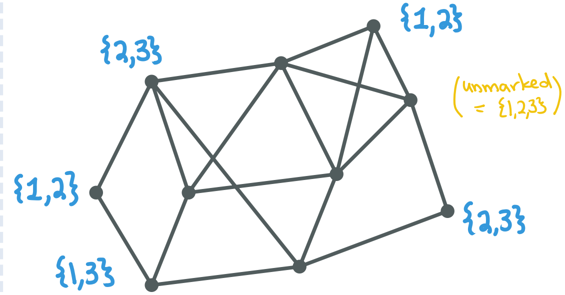
3-SAT

$f(x_1, \dots, x_n) =$
$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$
$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \dots$



Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



name colors:
"T" = "True"
"F" = "False"
"N" = "Neither"

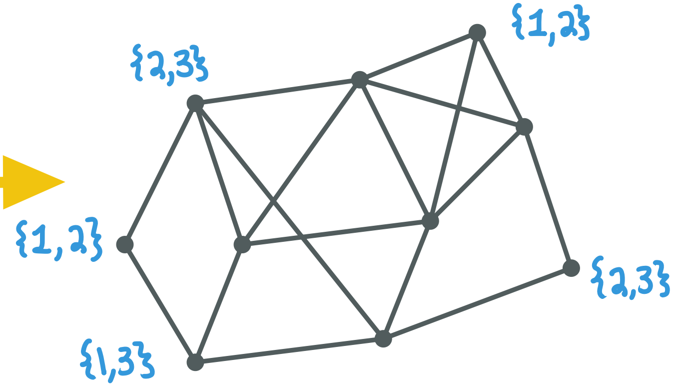
① variables

x_j

interpretation: $x_j = A_j.\text{color}$
$\bar{x}_j = \bar{A}_j.\text{color}$

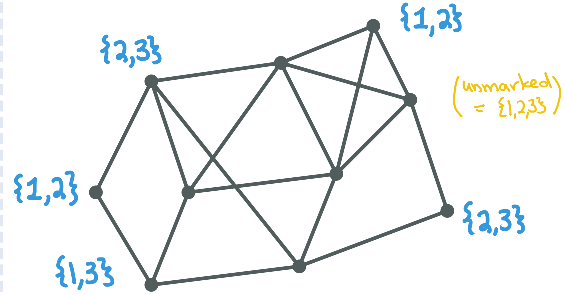
3-SAT

$f(x_1, \dots, x_n) = (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \dots$



Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



name colors:

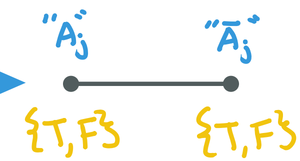
"T" = "True"

"F" = "False"

"N" = "Neither"

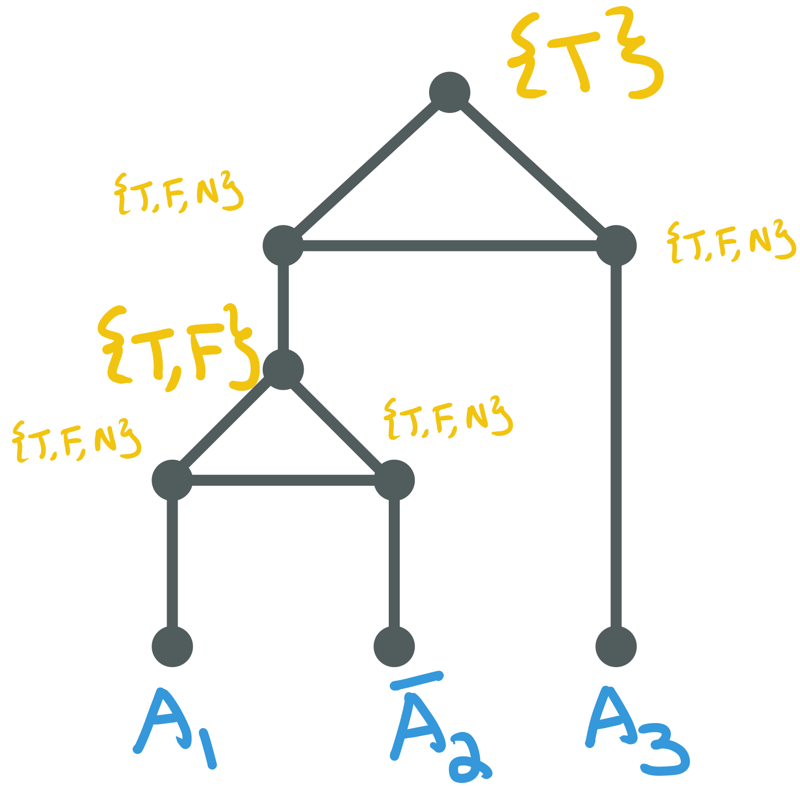
① variables

$x_j \rightarrow$

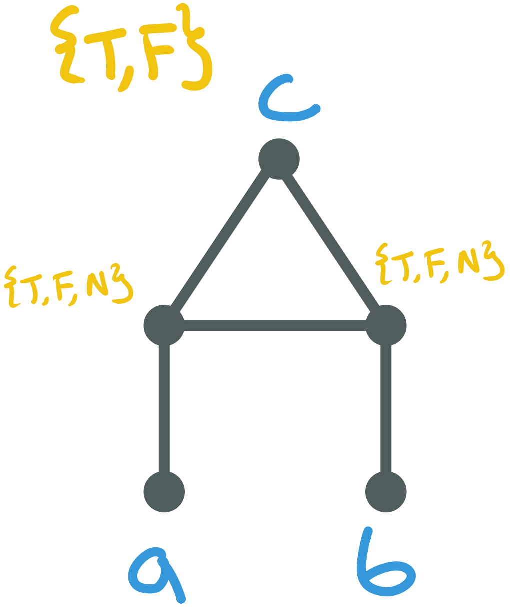


② clauses

$x_1 \vee \bar{x}_2 \vee x_3 \Rightarrow$



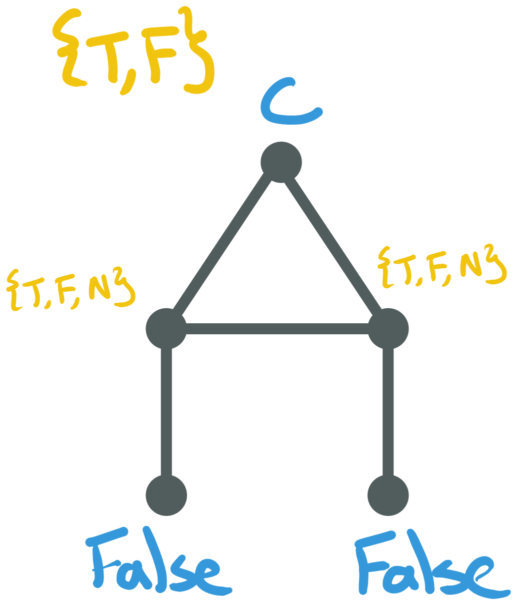
Idea: "or-gadget"



Lemma

c.color = True
possible iff $a \vee b$ = False

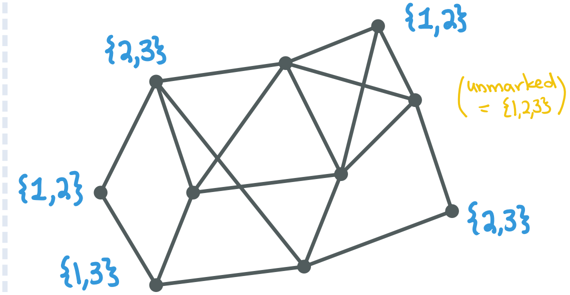
Proof



forces c=False

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



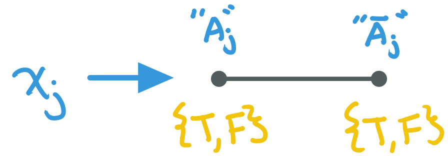
name colors:

"T" = "True"

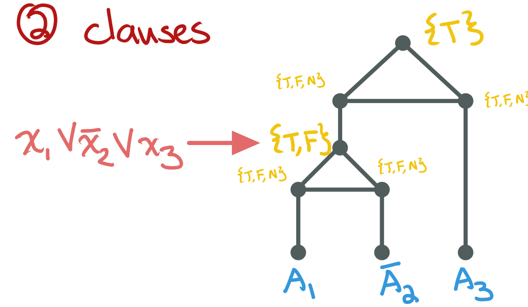
"F" = "False"

"N" = "Neither"

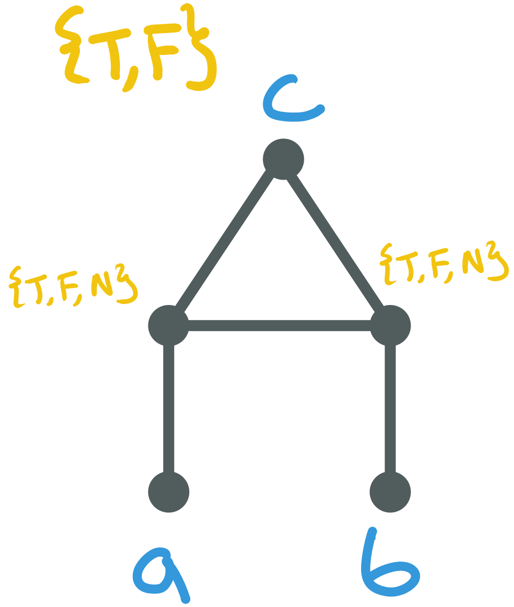
① variables



② clauses



Idea: "or-gadget"

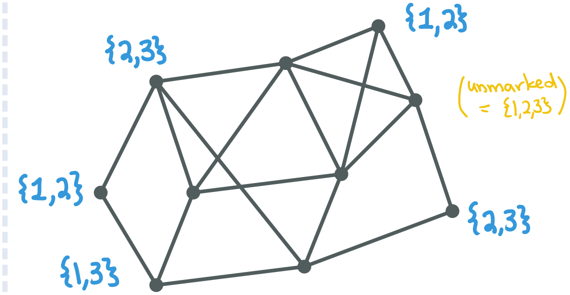


Lemma

c.color = True possible iff $a \vee b$ = False

Subset 3-coloring

for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.



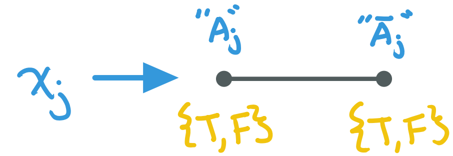
name colors:

"T" = "True"

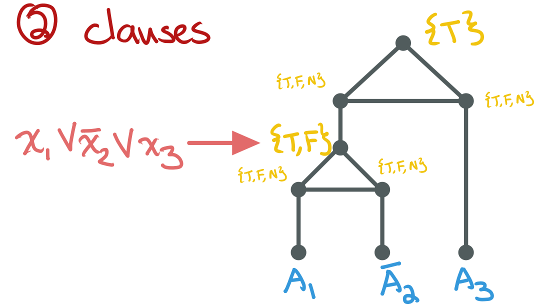
"F" = "False"

"N" = "Neither"

① variables

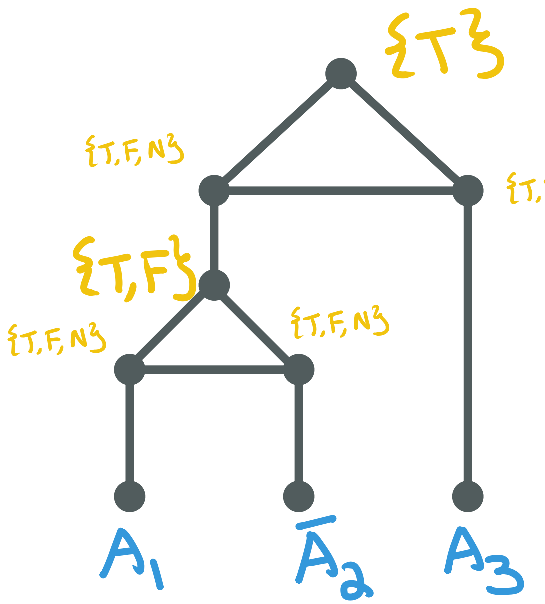


② clauses



$x_1 \vee \bar{x}_2 \vee x_3 \Rightarrow$

combines 2 ∨-gadgets

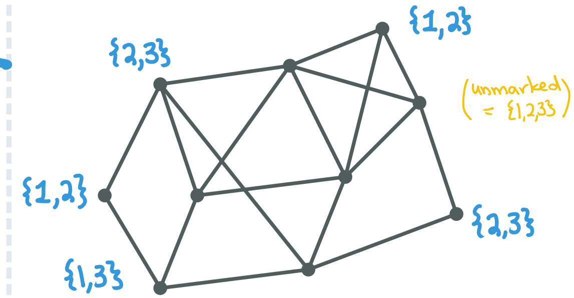


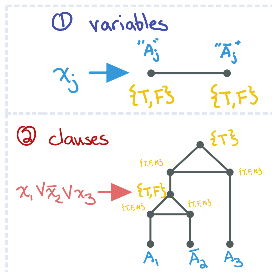
3SAT → "subset 3-coloring" → 3-coloring

claim: polytime algo for subset 3-coloring implies polytime algo for 3-SAT

Subset 3-coloring

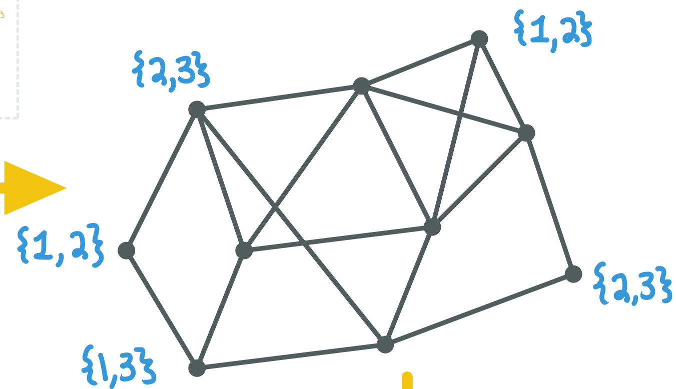
for each vertex v, subset $S_v \subseteq \{1,2,3\}$ of colors allowed for v.

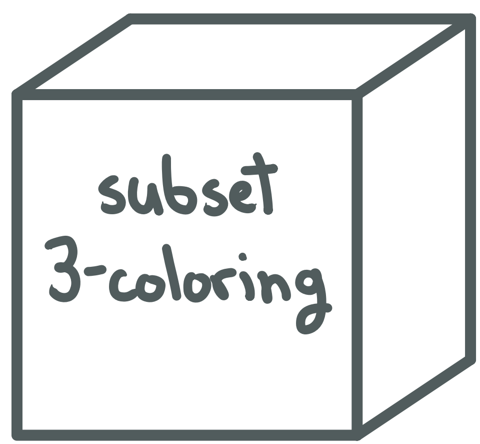




3-SAT

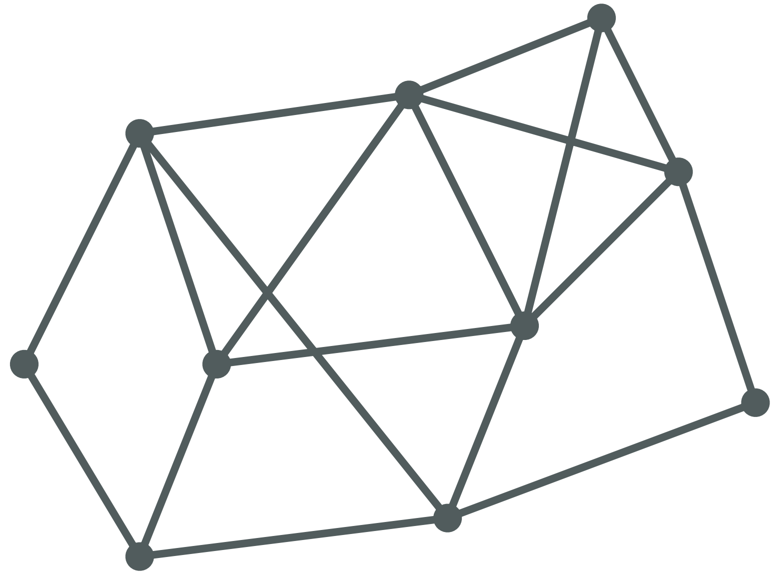
$$f(x_1, \ldots, x_n) = (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \ldots$$





Yes/no

Dominating Set



$S \subseteq V$ is **dominating** if all $v \in V$ either

(a) is in S (b) has a neighbor in S

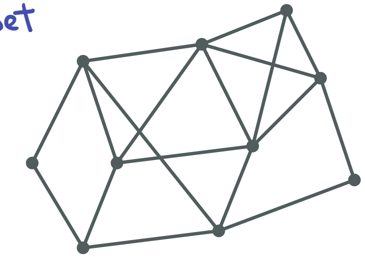
Goal: min cardinality dominating set

Decision version: dom. set w/ $\leq k$ vertices?

Theorem Dom set is NP-Hard ← Ideas?

claim: polytime algo for dominating set implies polytime algo for 3-SAT

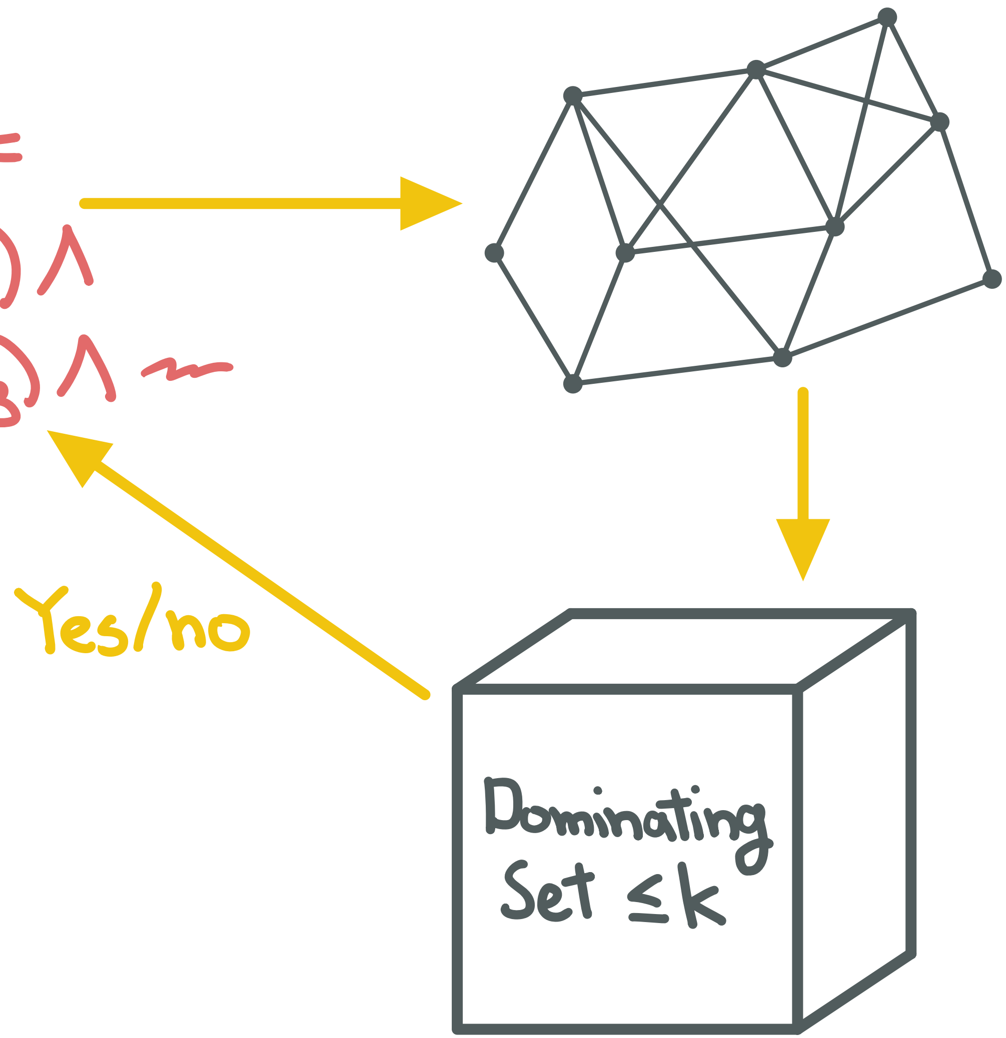
Dominating Set



$S \subseteq V$ is dominating if all $v \in V$ either (a) is in S (b) has a neighbor in S

3-SAT

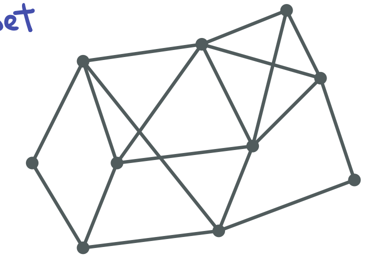
$f(x_1, \ldots, x_n) =$
$(\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge$
$(\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge \ldots$



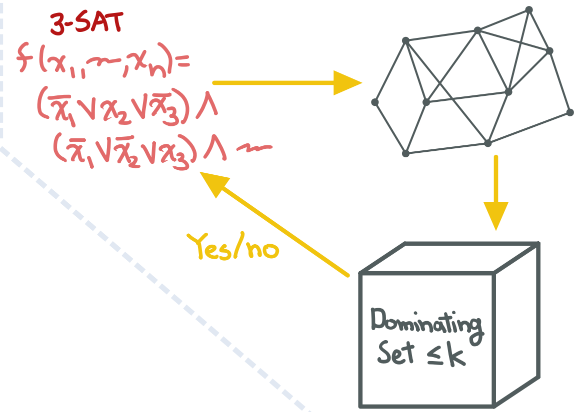
claim: polytime algo for dominating set implies polytime algo for 3-SAT

Ideas?

Dominating Set

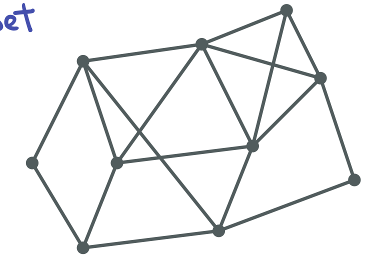


$S \subseteq V$ is dominating if all $v \in V$ either (a) is in S (b) has a neighbor in S

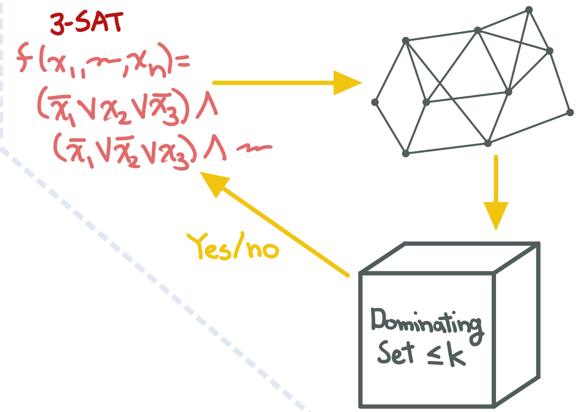


claim: polytime algo for dominating set implies polytime algo for 3-SAT

Dominating Set

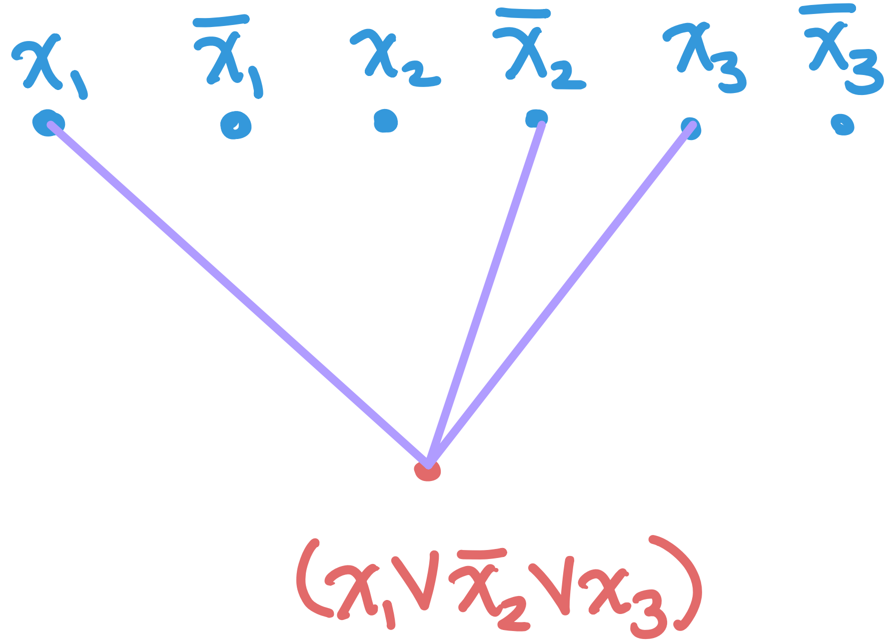


$S \subseteq V$ is dominating if all $v \in V$ either (a) is in S (b) has a neighbor in S



literals = vertices

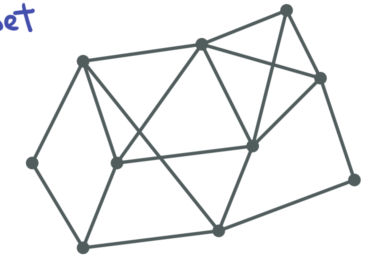
clauses = vertices



claim: polytime algo for dominating set implies polytime algo for 3-SAT

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

Dominating Set



$S \subseteq V$ is dominating if all $v \in V$ either (a) is in S (b) has a neighbor in S