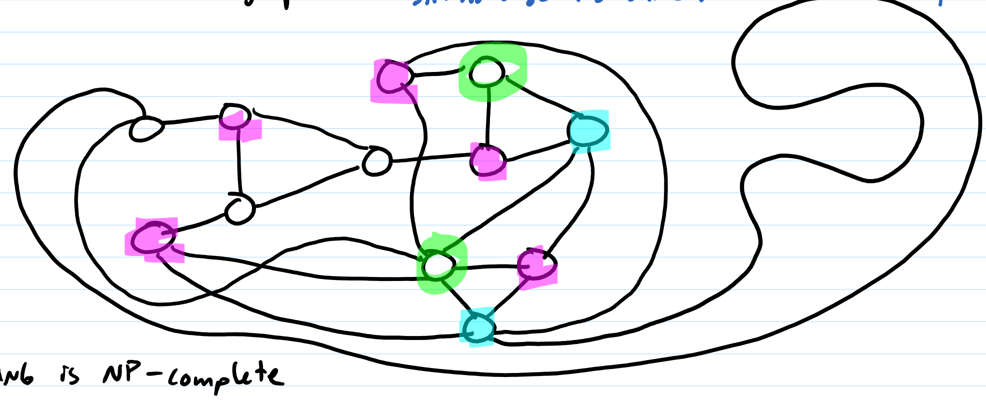
3-COLORING

Coloring of an undirected graph G : assignment of $\{1, \dots, k\}$ to vertices s.t. no edge has same value at both endpoints



Prove: 3-COLORING is NP-complete

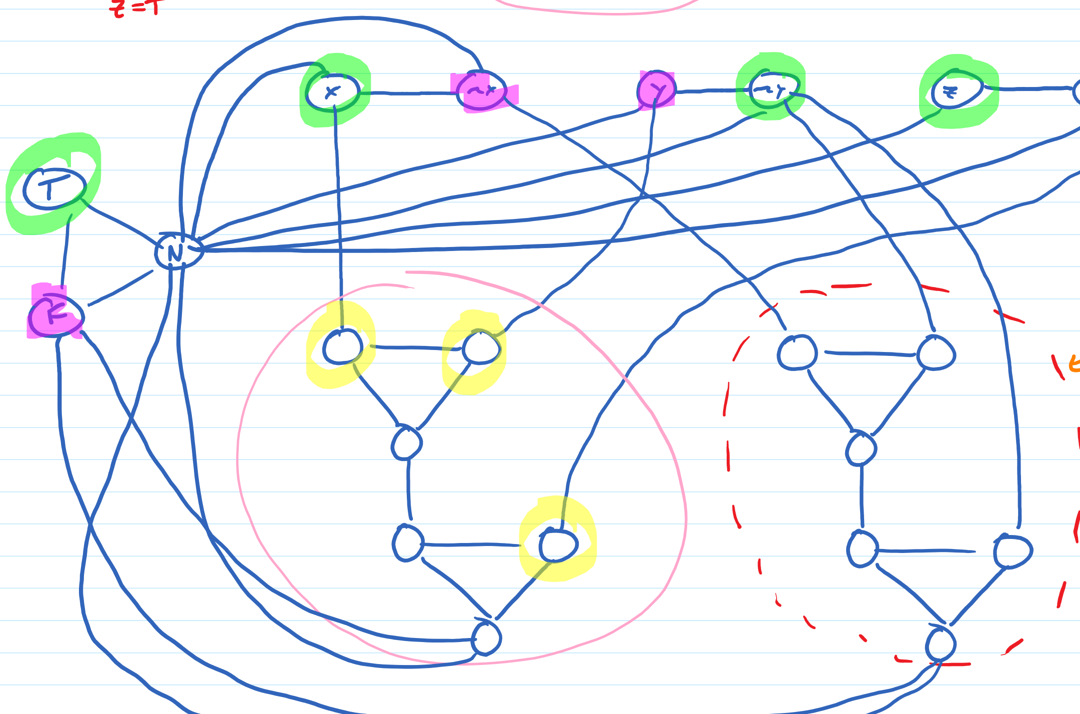
1) 3-COLORING $\in$ NP

2) 3-SAT $\le_p$ 3-COLORING [goal: given φ, create G s.t. φ is satisfiable $\Updownarrow$ G is 3-colorable]

Alg 3-SAT: 1) create G
2) return 3-COLORING(G)

$x = T$
$y = F$
$z = T$

$(x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg y)$

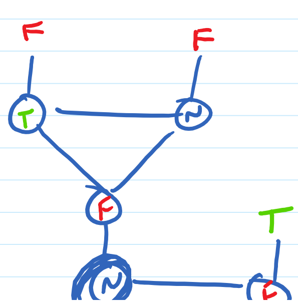


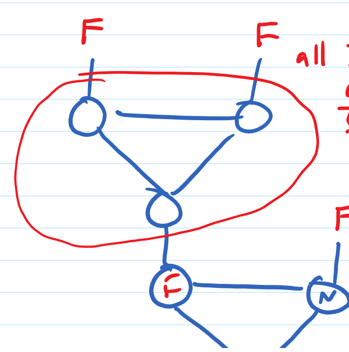
widget chosen so that valid assignment
$\Downarrow$
at least one T "input" per widget
$\Downarrow$
widgets (and rest of graph) are 3-colorable

AND

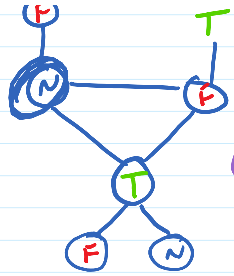
graph (including widgets) 3-colorable
$\Downarrow$
one T input per widget
$\Downarrow$
satisfying assignment

widget 3-colorable
$\Updownarrow$
one T "input"

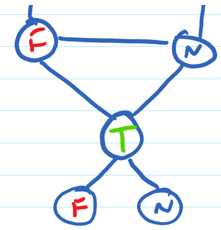




all 3 must be diff, but all 3 must not be F



(try other 6 possibilities too)



If 3-COLORING ∈ P then can find a 3-coloring in poly-time

If can't find coloring in poly-time then can't decide if 3-colorable in poly-time

FIND-3-COLORING(G)

If not 3-COLOR(G) then return nil

Add clique r, g, b to G to get G'

for each vertex v in G

INVARIANT:
a) G' is 3-colorable
b) 3-coloring of G' is a 3-coloring of G
c) all verts so far connected to two of r, g, b (so coloring easy to find)

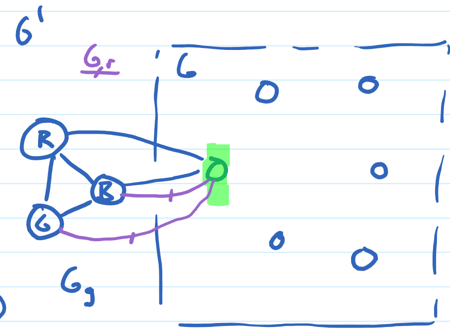
$G_r \leftarrow G'$ with added edges (v, G) (v, B)

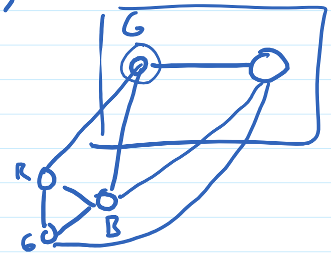
$G_g \leftarrow G'$ with added edges (v, R) (v, B)

$G_b \leftarrow G'$ with added edges (v, R) (v, G)

if 3-COLOR(G_r) = T
  $G' \leftarrow G_r$ color[v] = r
else if 3-COLOR(G_g) = T
  $G' \leftarrow G_g$ color[v] = g
else
  $G' \leftarrow G_b$ color[v] = b

return color





UNDIR-HC and UNDIR-HP

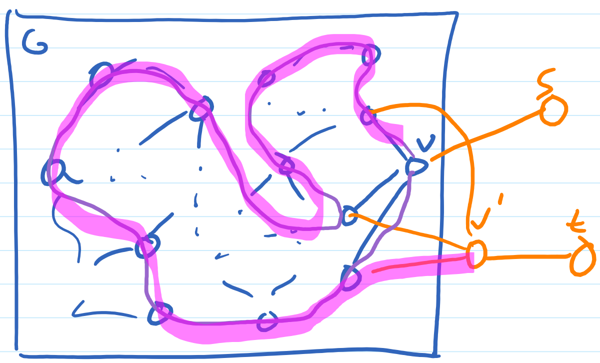
UNDIR-HC is NP-complete (see text)

Thm: UNDIR-HP is NP-complete

1) UNDIR-HP $\in$ NP
2) UNDIR-HC $\leq_P$ UNDIR-HP

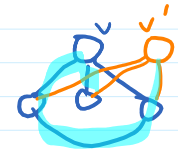
Goal: Given G, construct G' s.t. G' has HP $\longleftrightarrow$ G has HC

Alg for HC: 1) construct G'
2) return HP(G')



pick any vertex v; any HC in G visits v
make a copy v' of v
now HC is a HP

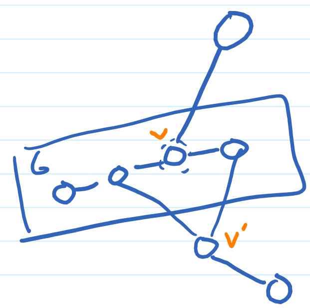
but may have introduced a HP where there was no HC
so add s,t to force HP to start/end s, v, ..., v', t



if $\exists$ HP in G', it must start/end at s + t

↳ $s, \underbrace{v, \ldots\ldots, v'}, t$

replace v' with v to get HC in HP



co-NP

Problem P is in NP means there is a poly-time algorithm A s.t. the set of instances x of P for which answer is YES is exactly the set of instances of x s.t. there exists y (poly-size) s.t. A(x,y) = OK

Problem X is in ~~NP~~ co-NP means there is a poly-time algorithm A s.t. the set of instances x of X for which answer is YES is exactly the set of instances of x s.t. there exists y (poly-size) s.t. A(x,y) = OK

Equivalent: $X \in$ co-NP means $\overline{X} \in NP$

Ex: $\overline{SAT}$ = given a Boolean formula φ, does it not have a satisfying assignment?

$\overline{SAT} \in NP$?? $\overline{SAT} \in$ co-NP

THM: If $X \in P$ then $\overline{X} \in P$

Proof: Suppose $X \in P$. Then $\dfrac{\text{X-comp}(x)}{\text{return } \neg X(x)}$ is a poly-time alg for $\overline{X}$, so $\overline{X} \in P$

THM: If P = NP then NP = co-NP

Proof: Suppose P = NP

NP $\subseteq$ co-NP : if $X \in NP$ then $X \in P$ and $\overline{X} \in P$ so $\overline{X} \in NP$ and $\overline{\overline{X}} \in$ co-NP, $\overline{\overline{X}} = X$

co-NP $\subseteq$ NP : $\longleftarrow$

COR: If NP $\neq$ co-NP then P $\neq$ NP

Thm: If NP $\neq$ co-NP then SAT $\notin$ co-NP (and $\overline{SAT} \notin$ NP)