Analysis of Algorithms

V. Adamchik

University of Southern California

CSCI 570

Lecture 12

NP-Completeness - II

Reading: chapter 9

In 1936 Alan Turing described

- A simple formal model of computation now known as Turing machines.
- A proof that TM can NOT solve the halting problem.
- A universal TM that can simulate any TM.
- A proof that NO Turing machine can determine whether a given proposition is provable from the axioms of first-order logic.
- Compelling arguments that a problem not computable by a Turing machine is not "computable" in the absolute (human) sense.
- A non-deterministic Turing machine: for each state it makes an arbitrary choice between a finite of possible transitions.

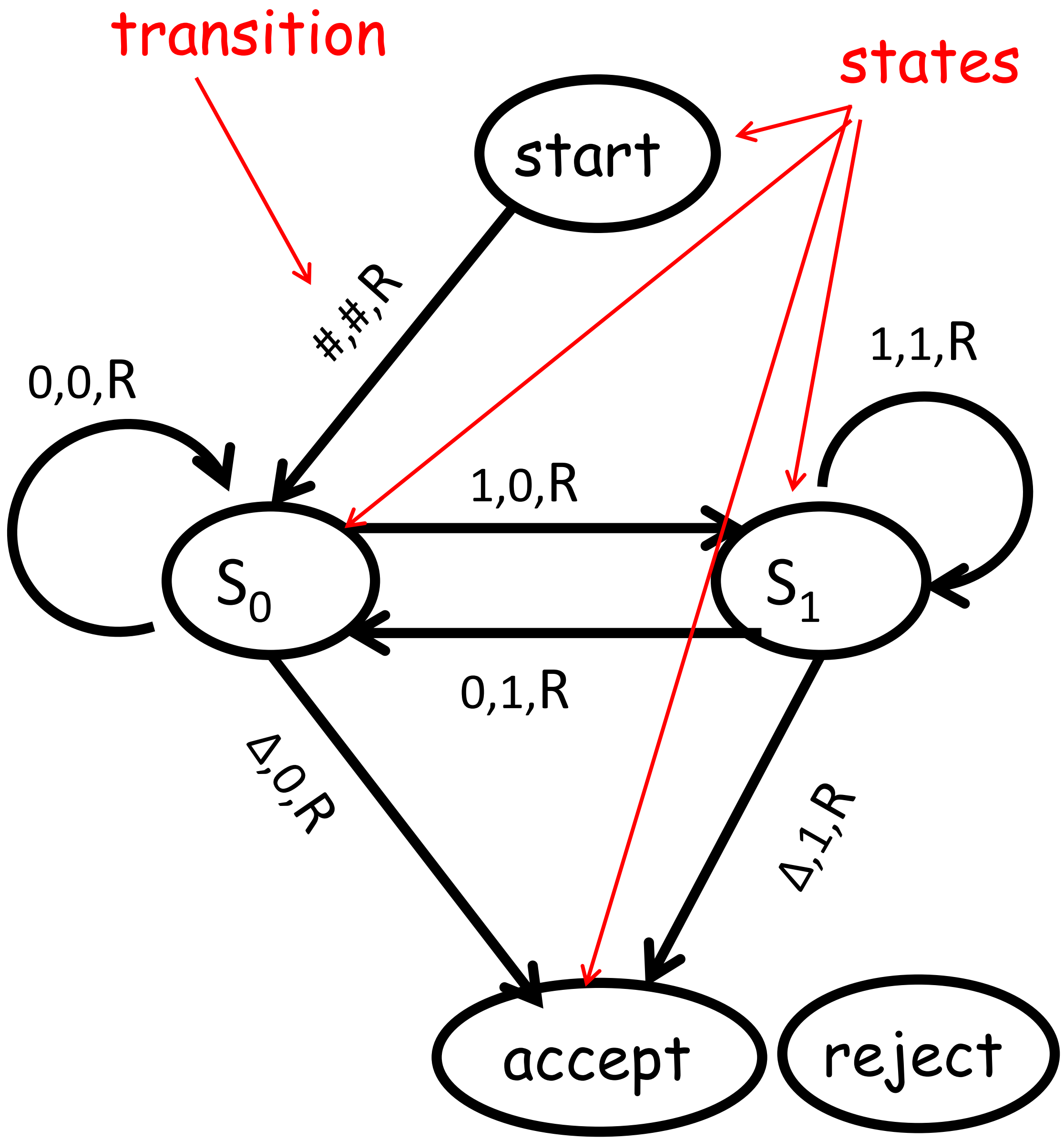
Deterministic Turing Machine

The machine that takes a binary string and appends 0 to the left side of the string.

Input: #10010Δ
Output: #**0**10010Δ

\# - leftmost char
Δ – rightmost char

Transition on each edge
read,write,move (L or R)



Non-Deterministic Turing Machine

- NDTM is a choice machine: for each state it makes an arbitrary choice between a finite (possibly zero) number of states.
- The computation of a NDTM is a tree of possible configuration paths.
- One way to visualize NDTM is that it makes an exact copy of itself for each available transition, and each machine continues the computation.
- Rabin & Scott in 1959 shown that adding non-determinism does not result in more powerful machine.
- For any NDTM, there is a DTM that accepts and rejects exactly the same strings as NDTM.
- P vs. NP is about whether we can simulate NDTM in polynomial time.

Complexity Classes

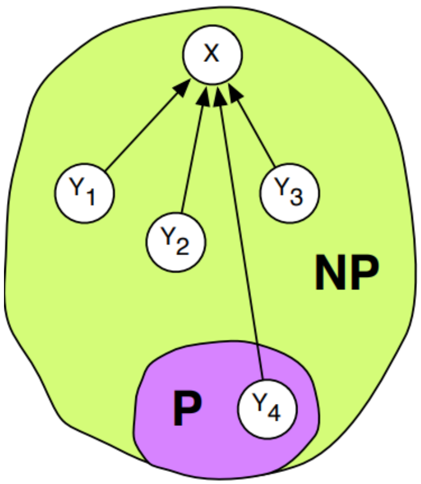
P = set of problems that can be solved in polynomial time by a DTM.

NP = set of problems that can be solved in polynomial time by a NDTM.

NP = set of problems for which solution can be verified in polynomial time by a deterministic TM.

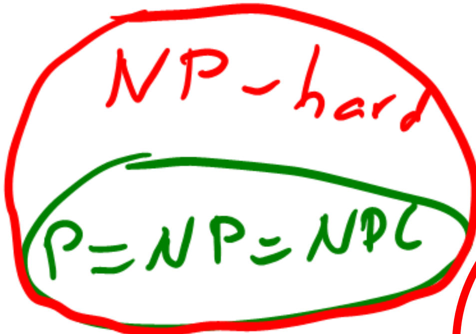
X is *NP-Hard*, if $\forall Y \in NP$ and $Y \leq_p X$.

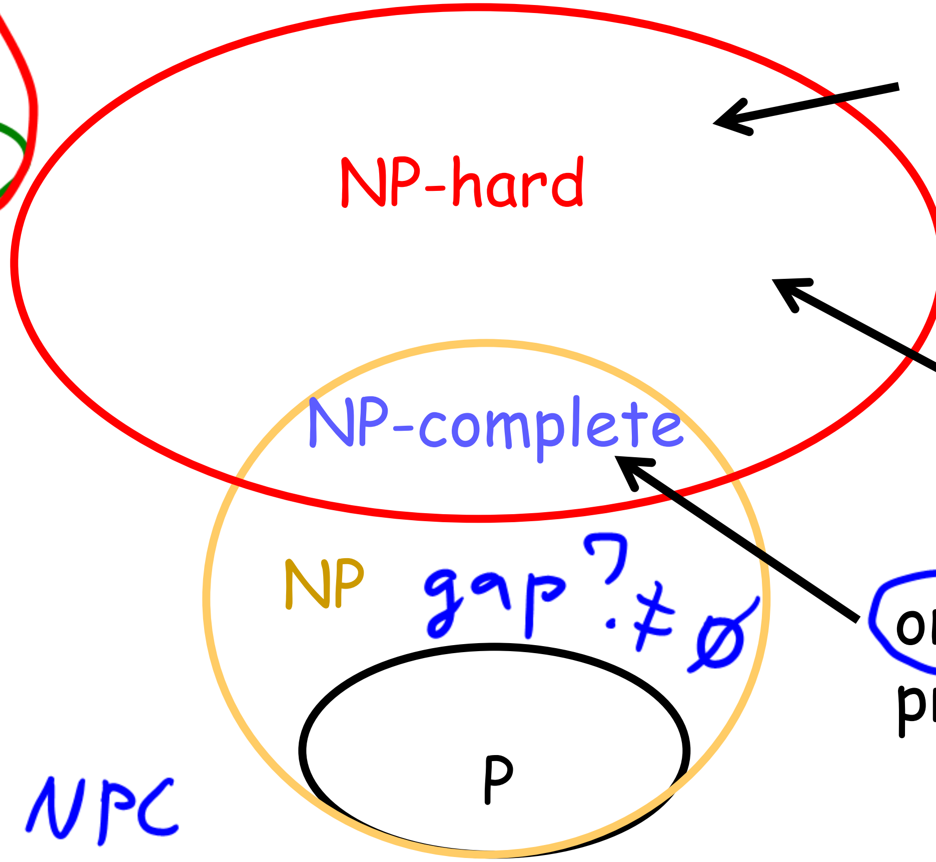
X is *NP-Complete*,
if X is NP-Hard and $X \in NP$.



$P = NP$

Venn Diagram ($P \neq NP$)





Undecidable:
Halting Problem

optimization
& decision
problems

only decision
problems

NPC problems are the most difficult NP problems.

Q: $\exists x$ s.t. $x \in NP$, $x \notin NPC$

It's not known if NPC problems can be solved by a *deterministic* TM in polynomial time.

NPC problems can be solved by a *non-deterministic* TM in polynomial time.

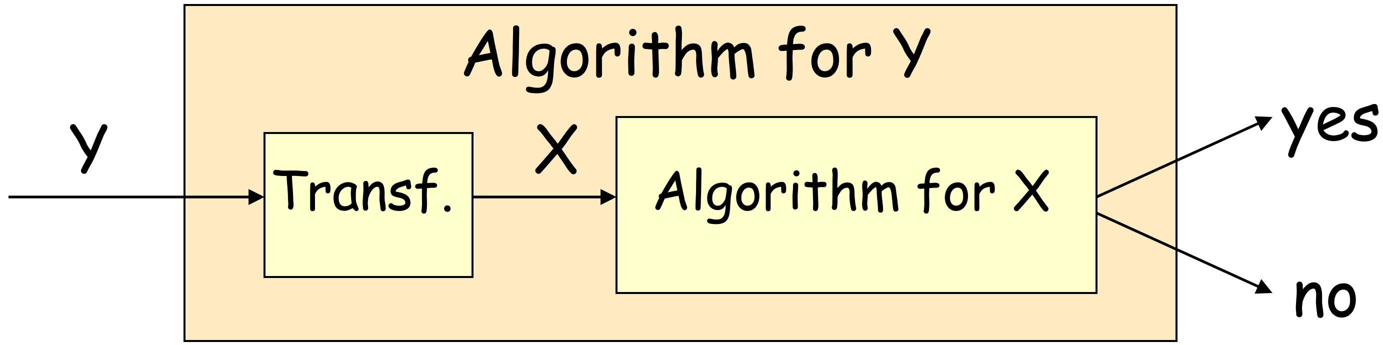
NP-Completeness Proof Method

To show that X is NP-Complete:

1) Show that X is in NP

2) Pick a problem Y, known to be an NP-Complete

3) Prove $Y \leq_p X$ (reduce Y to X)



Cook-Levin Theorem (1971)

Theorem. CNF SAT is NP-complete.

The theorem was proven without means of reduction

NP-Complete Problems

$3SAT \le_p IS$

Independent Set:

Given graph G and a number k, does G contain a set of at least k independent vertices?

Vertex Cover:

Given a graph G and a number k, does G contain a vertex cover of size at most k.

A Hamiltonian cycle: (no proof)

Given a graph G, does G contain a cycle that visits each vertex exactly once.

TSP

Discussion Problem 1

Given SAT in Conjunctive Normal Form (CNF)

$$(X_1 \vee \neg X_3) \wedge (X_1 \vee \neg X_2 \vee X_4 \vee X_5) \wedge \ldots$$

with any number of clauses and any number of literals in each clause. Prove that SAT is polynomial time reducible to 3SAT.

SAT $\in$ NPC (Cook-Levin)

does it follow that 3SAT $\in$ NPC ?

general $\not\Rightarrow$ particular

2SAT $\in$ P

$$\boxed{SAT \le_p 3SAT}$$

1) a clause has ≤ 3 literals

2) $(\overbrace{a \vee b \vee c \vee d}^{T}) \Rightarrow \overbrace{\underbrace{(a \vee b \vee c)}_{F} \wedge \underbrace{(b \vee c \vee d)}_{T}}^{}$

$a = b = c = F,\ d = T$ $\qquad$ F

add extra variables

$a \vee b \vee c \vee d = (\underbrace{a \vee b}_{T} \vee x) \wedge (\bar{x} \vee \underbrace{c \vee d}_{T/F})$

$\qquad$ T $\qquad$ T $\qquad$ $\boxed{x = F}$, $\bar{x} = T$

Proof.

a) $a \vee b = T$
$\quad c, d = \forall$

b) $c \vee d = T$
$\quad a, b = \forall$

$(\underbrace{a \vee b}_{T/F} \vee x) \wedge (\bar{x} \vee \underbrace{c \vee d}_{T})$

$\qquad$ $\boxed{x = T}$ $\qquad$ T

c) $a = b = c = d = F$, it never occurs

3) $(a \vee b \vee c \vee \overset{DE}{(d \vee e)}) = (a \vee b \vee c \vee DE) =$

$= (a \vee b \vee x) \wedge (\bar{x} \vee c \vee DE) =$

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

$= (a \vee b \vee x) \wedge (\bar{x} \vee c \vee y) \wedge (\bar{y} \vee d \vee e)$

4) and so on...

is it polynomial?

SAT, n-literals, m-clauses

construction

3SAT, literals $O(m \cdot n)$

clauses $O(m \cdot n)$

<u>claim.</u> SAT is satisfiable iff 3SAT is satisfiable.

Proof.

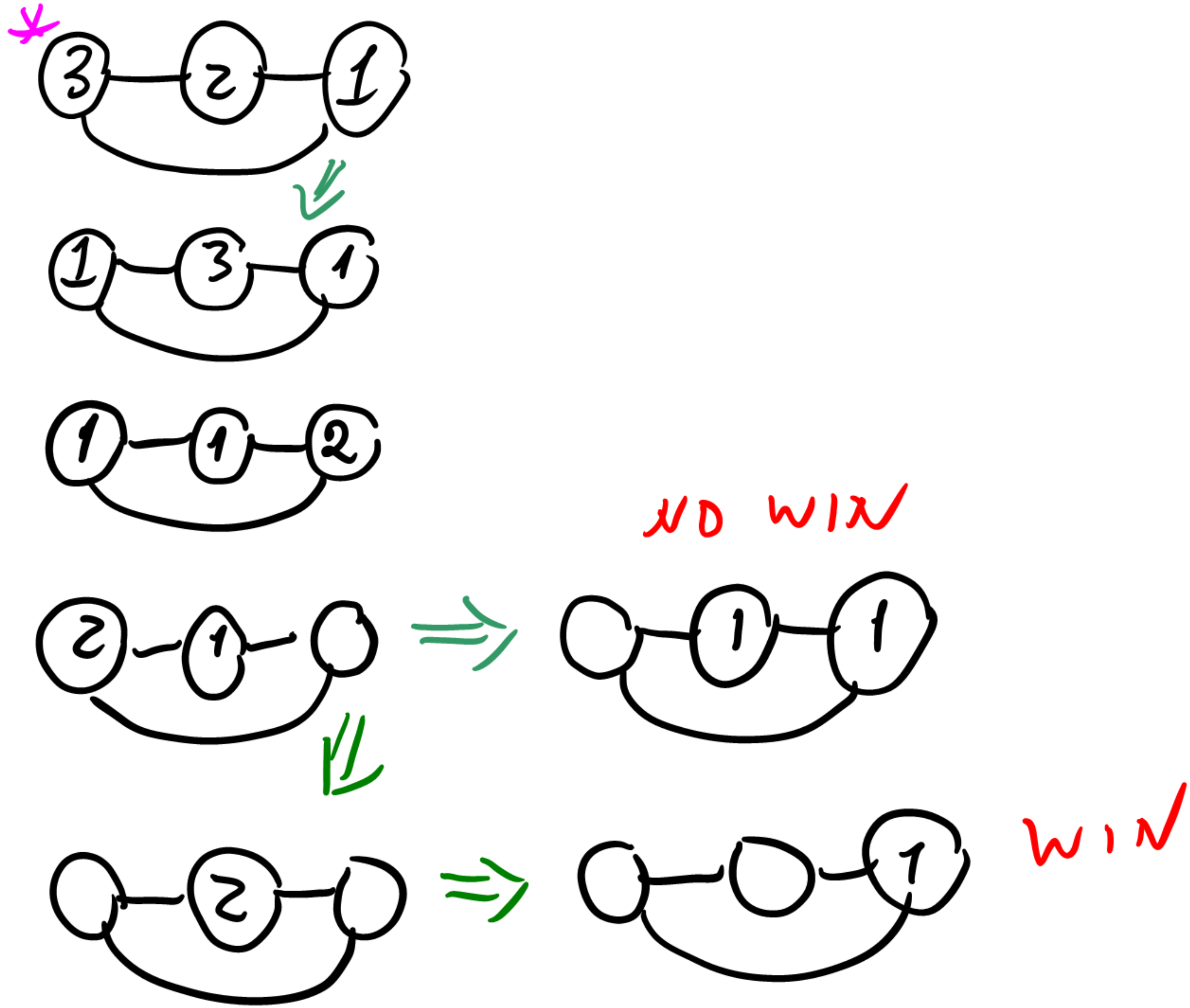
$\Rightarrow$ our construction, see above

$\Leftarrow$ nothing to prove! $3SAT \subset SAT$

Discussion Problem 2

You are given an undirected graph G = (V, E) and for each vertex v, you are given a number p(v) that denotes the number of pebbles placed on v. We will now play a game where the following move is the only move allowed. You can pick a vertex u that contains at least two pebbles, and remove two pebbles from u and add one pebble to an adjacent vertex. The objective of the game is to perform a sequence of moves such that we are left with exactly one pebble in the whole graph. Show that the problem of deciding if we can reach the objective is NP-complete. Reduce from the Hamiltonian Path problem.

$HP \le_p$ Pebbes


NO WIN
WIN

1. Pebbles $\in NP$
2. $HP \leq_p$ Pebbles

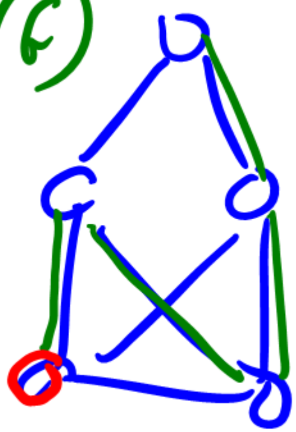
Reduction

$Y \leq_p X$

Construction

Given $\forall$ graph G with HP

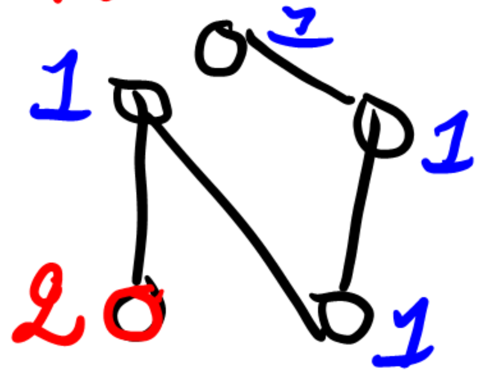
HP(G)



$\Longrightarrow$

$G' =$ pebble board

$p(v) = ?$



$p(v) \notin HP(G)$

Claim. G has a HP iff G' has a winning seq.

Proof.

HP: 2 1 1 1 1

0	2	1	1	1
0	0	2	1	1
0	0	0	2	1
0	0	0	0	2
0	0	0	1	0

$\Rightarrow$ G has a HP

find a winning seq.

follow the HP

$\Leftarrow$ G' has a winning seq.

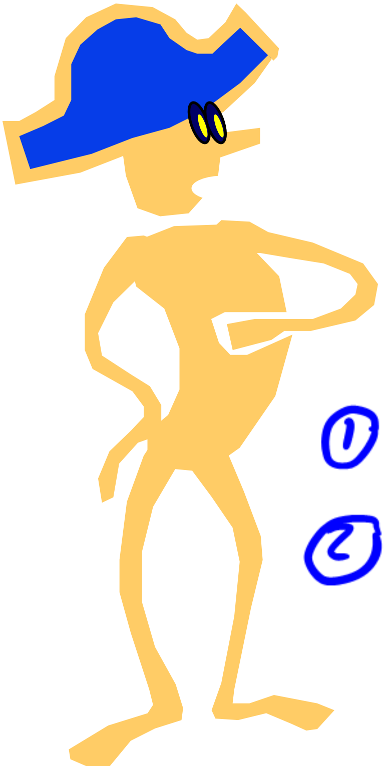
find a HP in G

winning seq is a HP

we won't visit the same vertex twice,

except the last move

Graph Coloring



Given a graph, can you color the nodes with ≤ k colors such that the endpoints of every edge are colored differently?

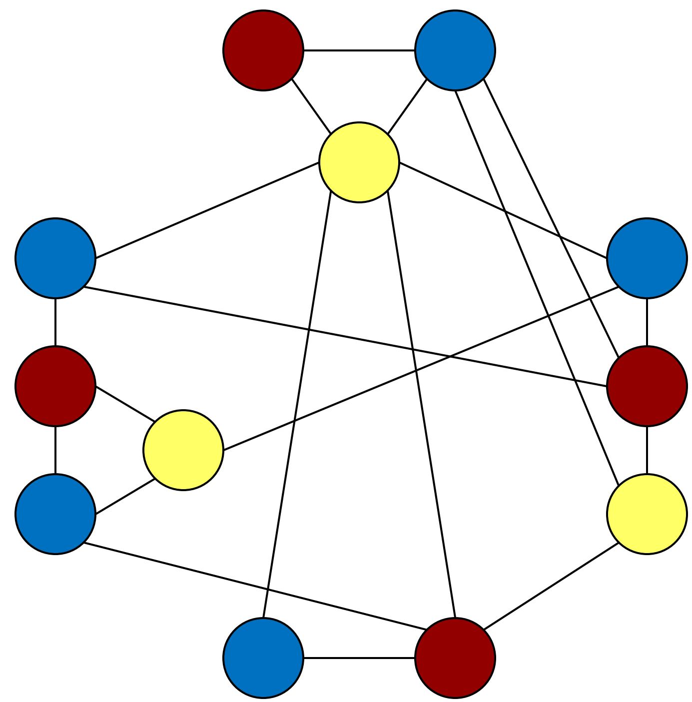
① Planar graph $\in P$, $k=4$

② $k=2$, in P

Theorem. (k>2)
k-Coloring is NP-complete.

$k=3$

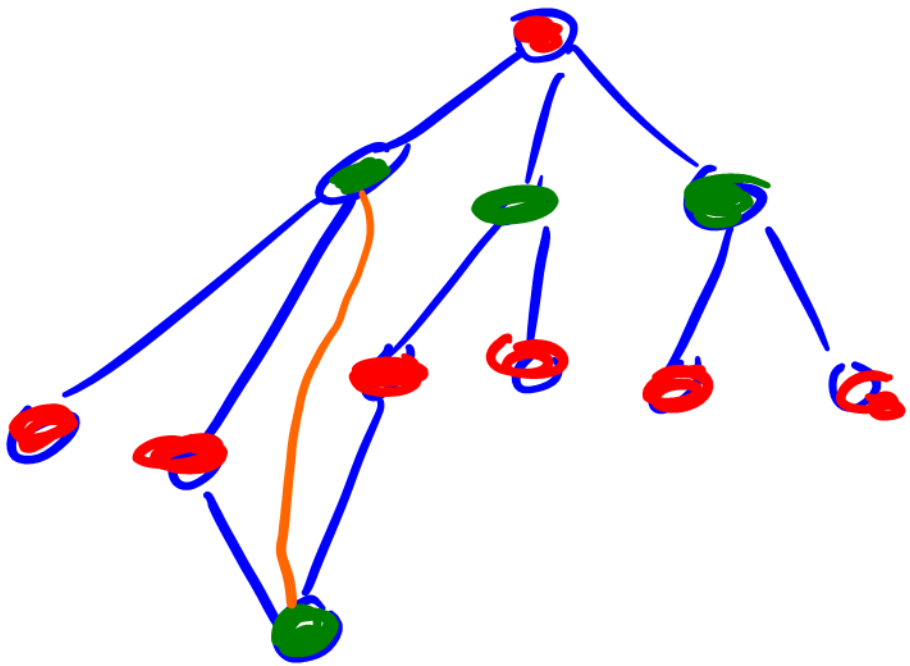
3-COLORS



Graph Coloring: k = 2

How can we test if a graph has a 2-coloring?

BFS



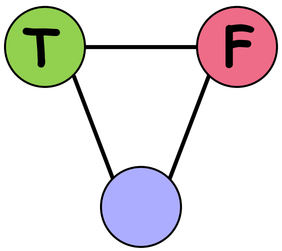
3-SAT $\leq_p$ 3-colorable

We construct a graph G that will be 3-colorable iff the 3-SAT instance is satisfiable.

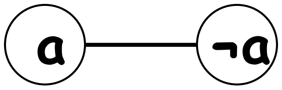
Graph G consists of the following gadgets.

only one

A truth gadget:



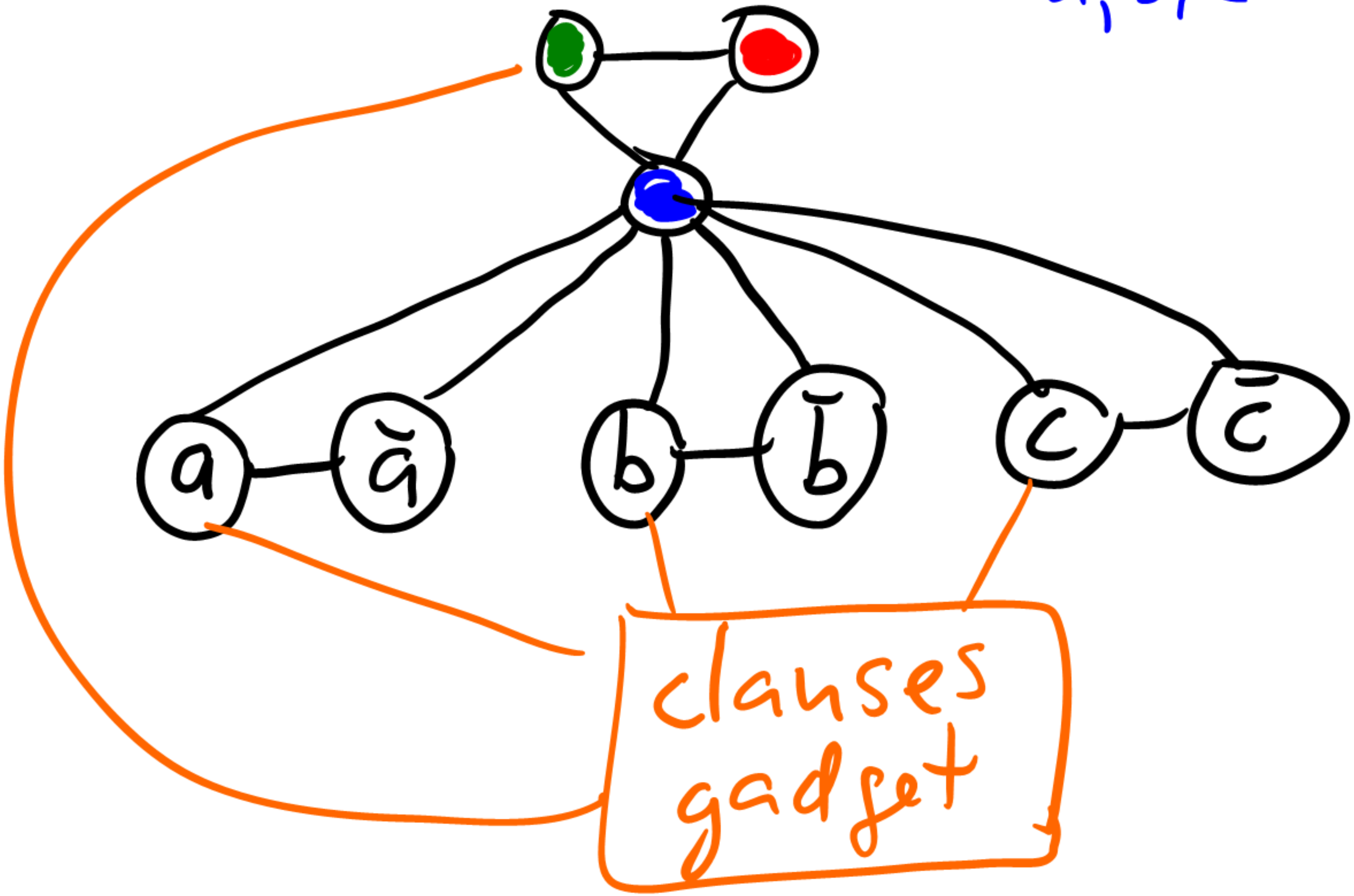
A gadget for each variable:



3-SAT $\leq_p$ 3-colorable

Combining those gadgets together (for three literals)

a, b, c



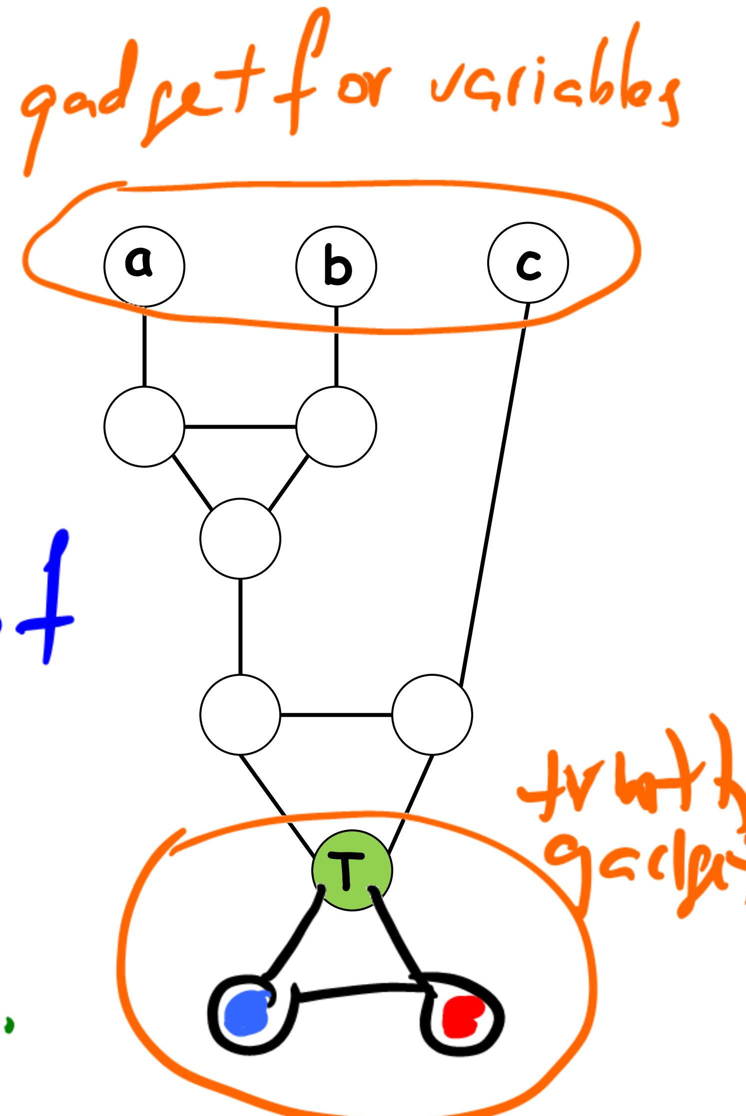
3-SAT $\leq_p$ 3-colorable $(a \vee b \vee c)$

A special gadget for each clause

This gadget connects a truth gadget with variable gadgets.

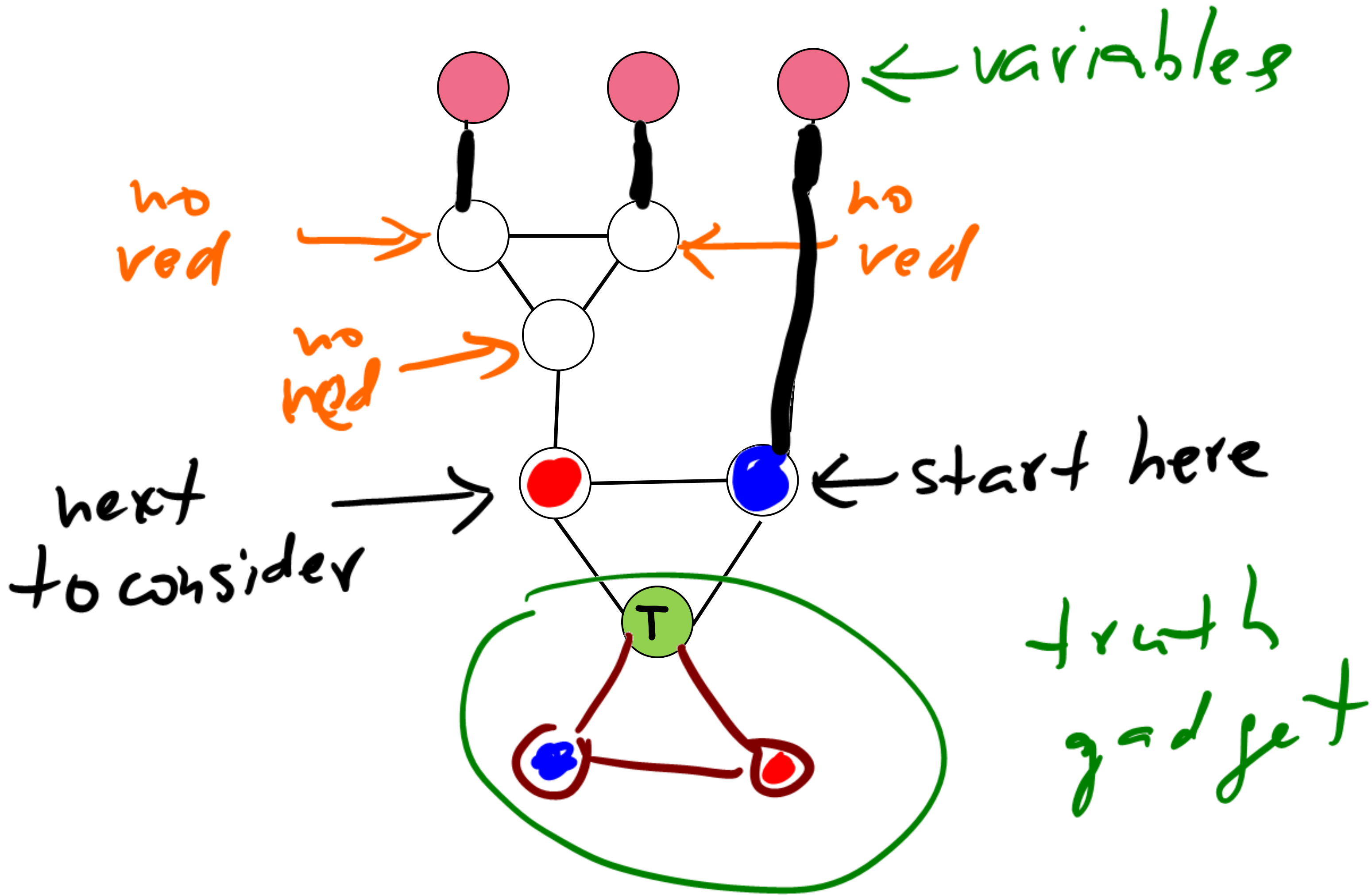
we can color this graph with 3 colors iff one of the literals is true.

If a=b=c=F, then we cannot color this graph.



3-SAT $\leq_p$ 3-colorable

Suppose all a, b and c are all False (red).

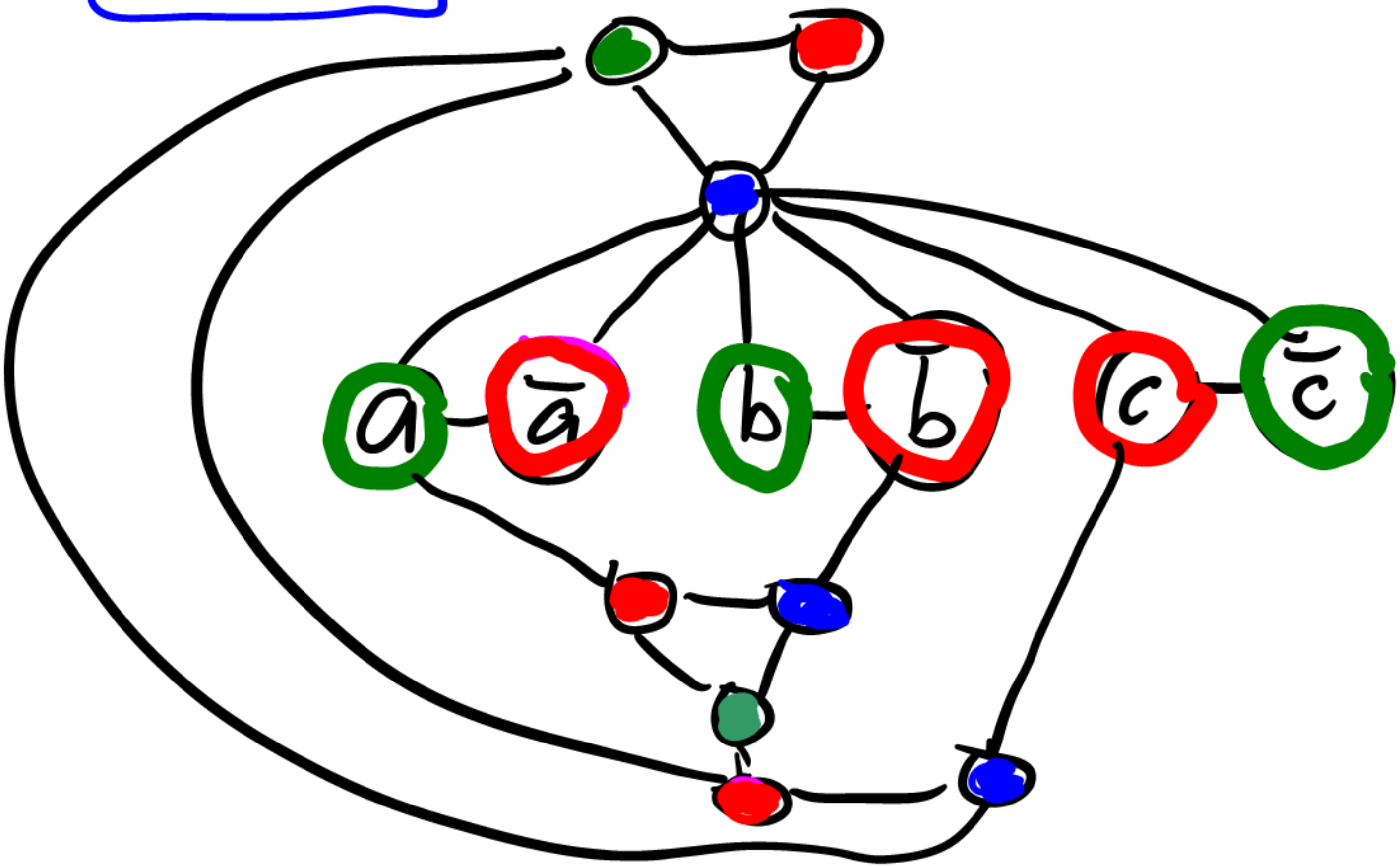


3-SAT $\leq_p$ 3-colorable

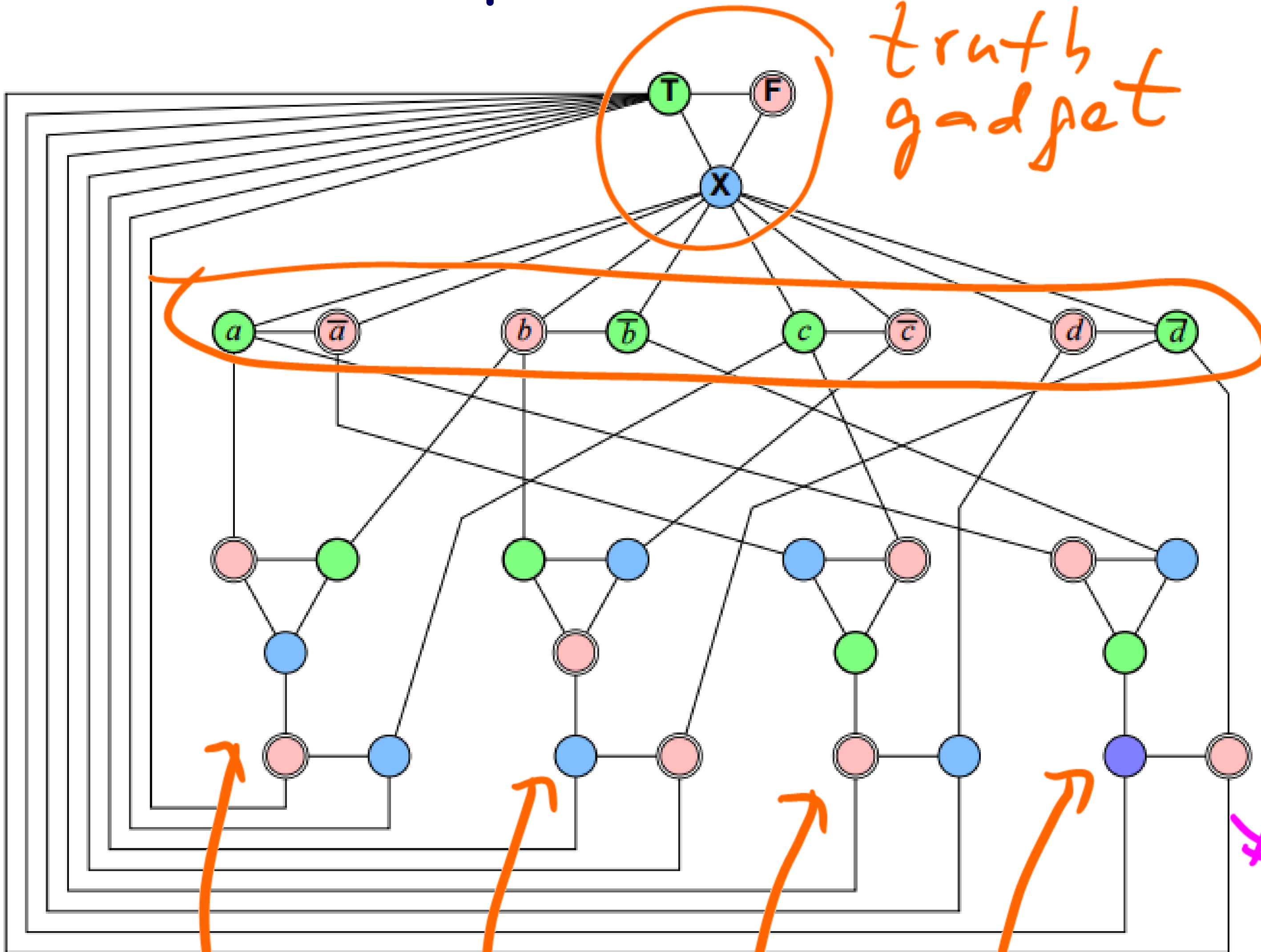
We have showed that if all the variables in a clause are false, the gadget cannot be 3-colored.

Let $a = T$, $\bar{b} = c = F$

Example: $\boxed{a \vee \neg b \vee c}$



Example with four clauses



a=c=T

b=d=F

A 3-colorable graph derived from a satisfiable 3CNF formula.

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

# of vertices = $5m + 2n + 3$

3-SAT $\leq_p$ 3-colorable

<u>Claim</u>: *3-SAT instance is satisfiable if and only if G is 3-colorable.*

Proof: $\Rightarrow$) Given satisfiable 3SAT, truth assignment

T = green
F = red

by construction
truth gadget → color
variable gadget → color
clause gadget → coloring is forced

3-SAT $\leq_p$ 3-colorable

<u>Claim</u>: *3-SAT instance is satisfiable if and only if G is 3-colorable.*

Proof: $\Leftarrow$) Given a special graph, which is 3-colorable.

Goal: find a truth assignment.

Look at the variable gadget and assign red to F, green to T.

Sudoku: $n^2 \times n^2$

$n \to \infty$

NP-?

NP-hard?

9-COLORS

Sudoku graph

mini-grid

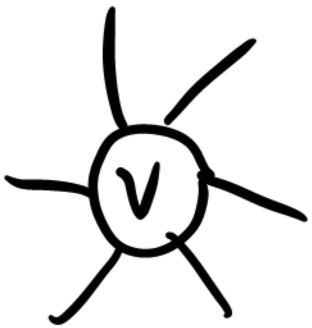
2			3		8		5	
		3		4	5	9	8	
		8			9	7	3	4
6		7		9				
9	8			?			1	7
				5		6		9
3	1	9	7			2		
	4	6	5	2		8		
	2		9		3			1

vertex: each cell, 81

edge: two vertices connected if they are in the same row, col and mini-grid

Sudoku Graph (9×9)

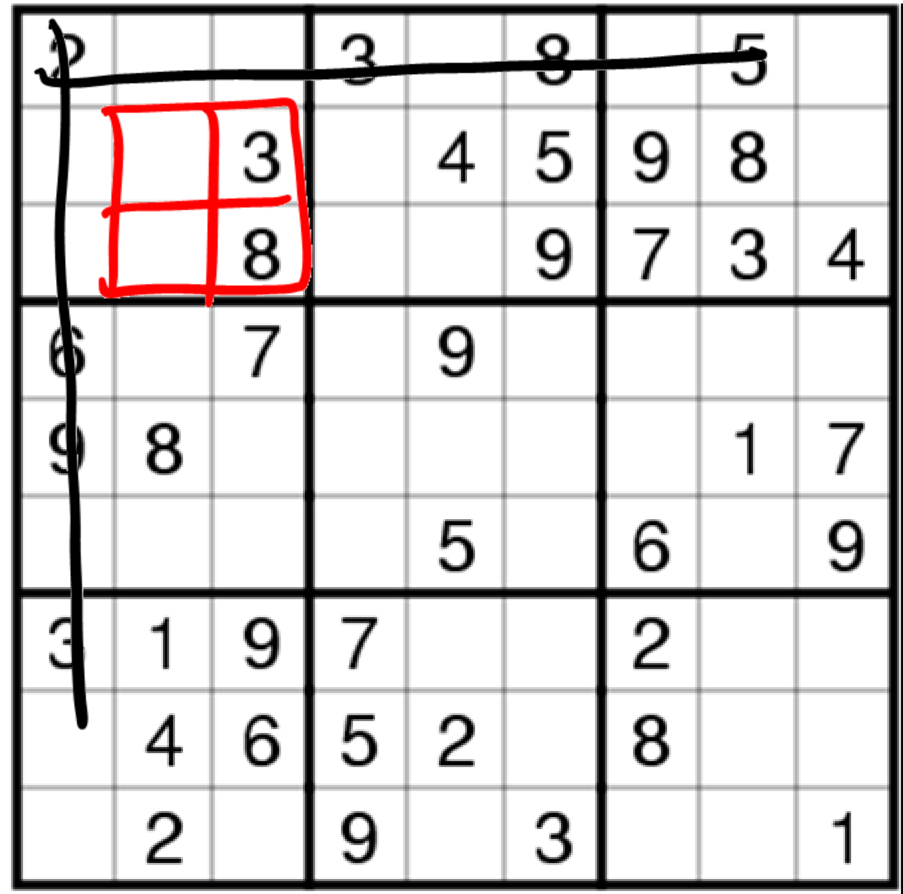
how many edges?



degree?

$8+8+4=20$

$$81 \cdot \frac{20}{2} = 810$$



Sudoku

Constructing a Sudoku graph we have proved:

did we prove that Sudoku $\in$ NPC? NO

$$\text{Sudoku} \le_p \text{9-COLORS}$$

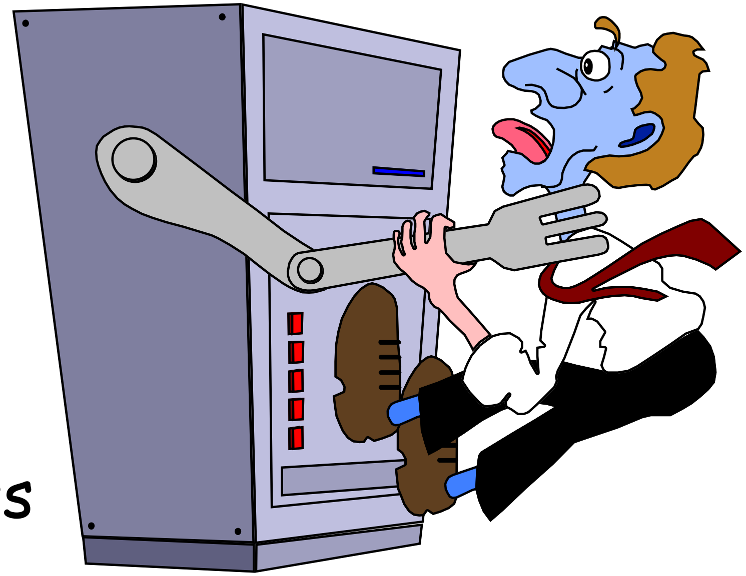
how can we use this reduction?

it gives us another way to solve Sudoku

$$Y \le_p SAT$$

we can use SAT solver to solve Y.

Don't be afraid of NP-hard problems.



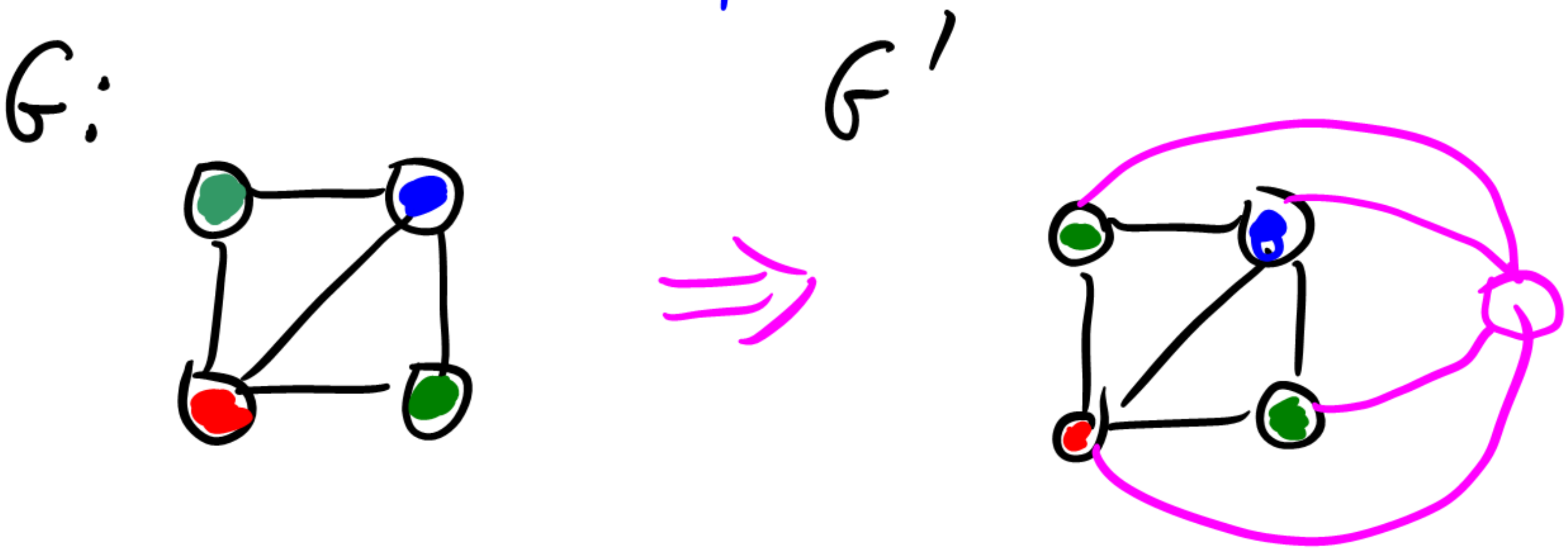
Many reasonable instances (of practical interest) of problems in class NP can be solved!

The largest solved TSP an 85,900-vertex route calculated in 2006. The graph corresponds to the design of a customized computer chip created at Bell Laboratories, and the solution exhibits the shortest path for a laser to follow as it sculpts the chip.

Discussion Problem 3

Prove that 4-COLOR is NP-complete. in 5 min.

3-COLOR $\leq_p$ 4-COLOR



K-COLOR $\leq_p$ (K+1)-COLOR

NOT AN EXAM PROBLEM

Discussion Problem 4

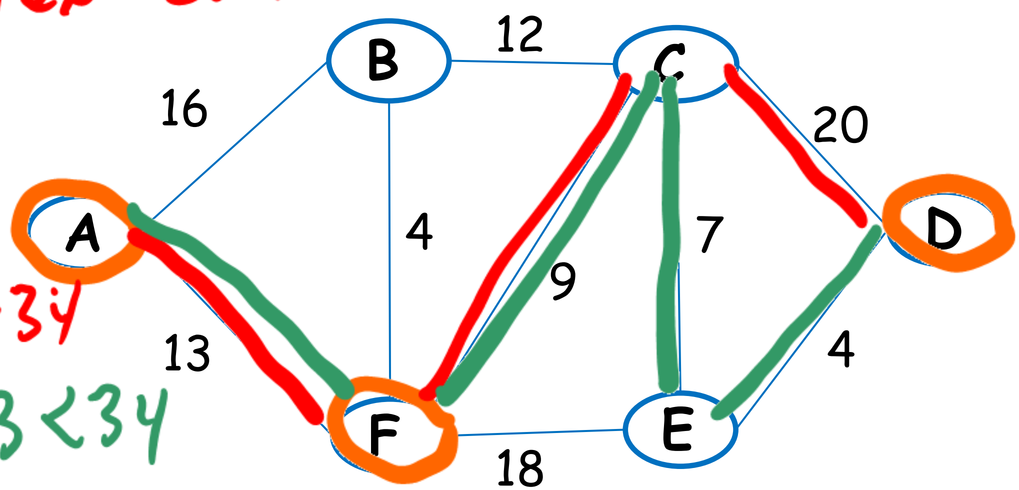
The Steiner Tree problem is as follows. Given an undirected weighted graph G=(V,E) with positive edge costs, a subset of vertices $R \subseteq V$, and a number C. Is there a tree in G that spans all vertices in R (and possibly some other in V) with a total edge cost of at most C? Prove that this problem is NP-complete by reduction from Vertex Cover.

$R \notin$ Vertex Cover

$C \notin$ Vertex Cover

Example.

R = {A,F,D} and C = 34.



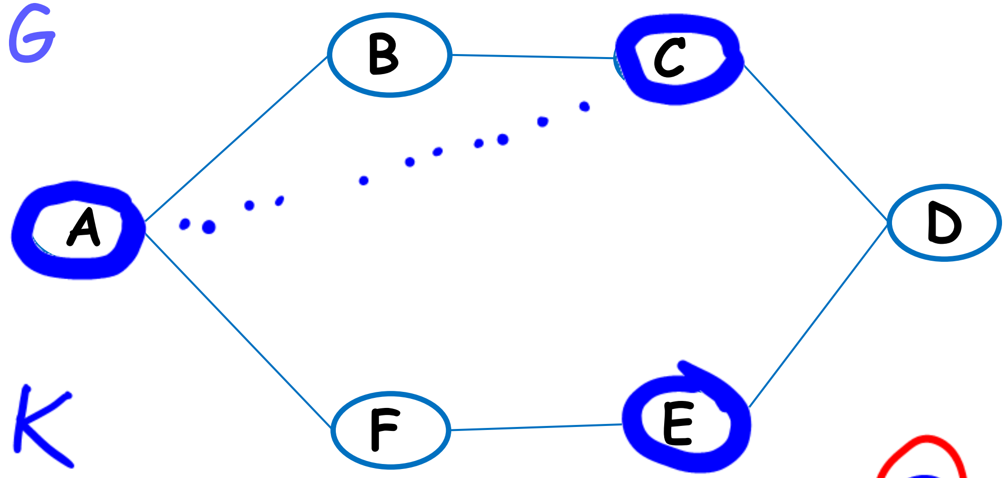
$MST \in P$

cost = 13+5+20 = 42 > 34

cost = 13+9+7+4 = 33 < 34

$|VC(G)| = 3$

G



INPUT

$|VC(G)| \leq K$

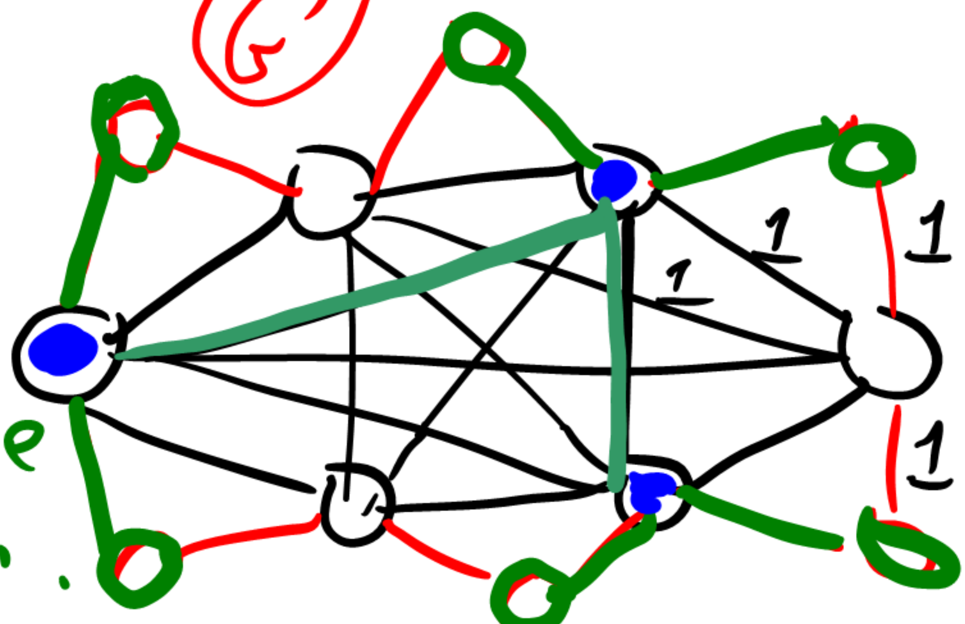
construct G'; also specify R and C.

G'

$R \not\subset G$

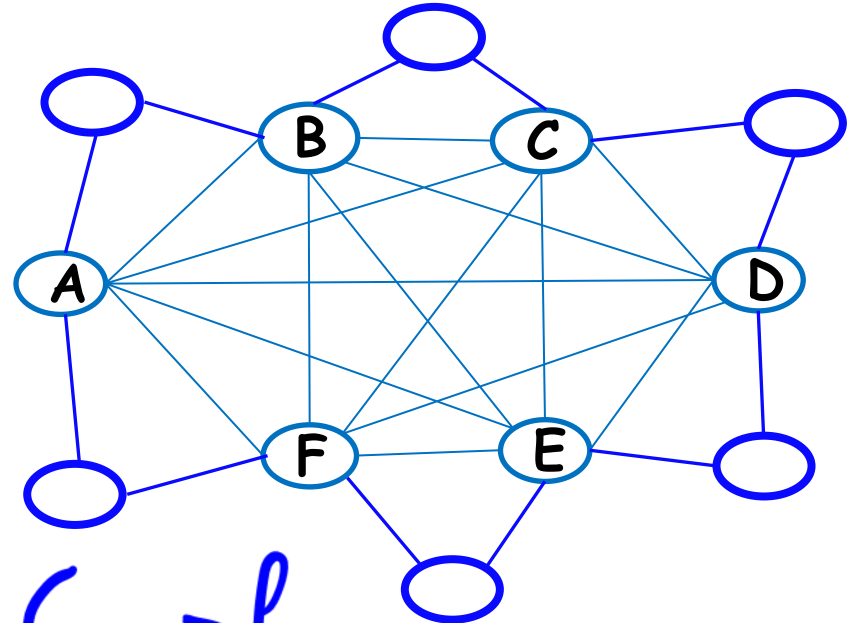
$|R| = G(E)$

Steiner tree is in green.



complete

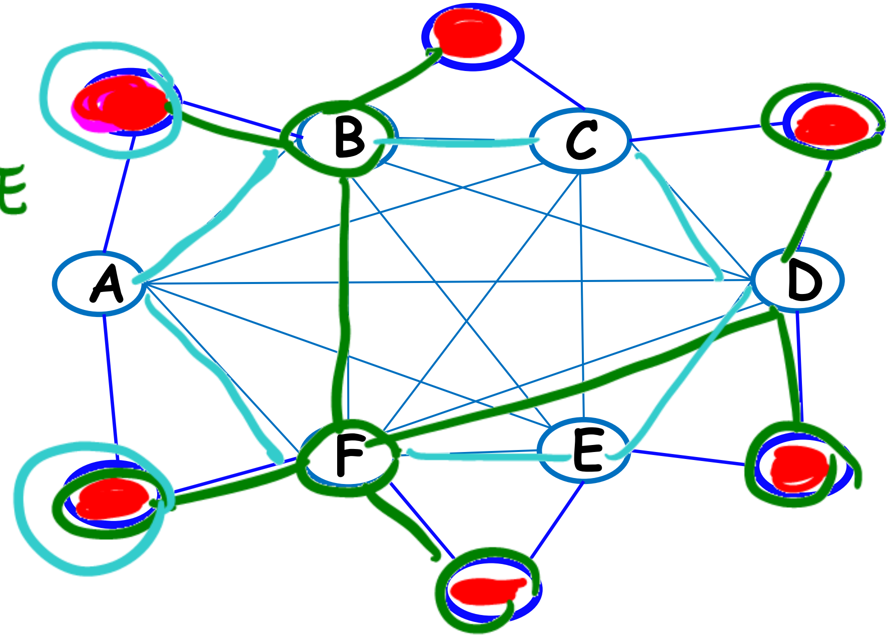
$C \le (K-1) + G(E)$



Claim. G has a VC of size $\le K$ iff G' has a Steiner tree with $R = G(E)$ and $C \le K-1+E$

$\Rightarrow$ by construction, see above

$\Leftarrow$) G' has a Steiner tree, R and $C \le k-1+E$



Goal: find VC(G)

Proof.

$VC(G) = ST - R$

a) prove that $VC(G)$ is a VC $\{B, F, D\}$

b) prove that $|VC(G)| \le K$

edges

$|VC| = |ST| - |R| \le |ST| - E = cost(ST) + 1 - E$

vertices

$\le k - 1 + E + 1 - E = K$