CS 341 – Algorithms

Lecture 19 – Hard Graph Problems

28 July 2021

Today’s Plan

1. Hamiltonian Cycle

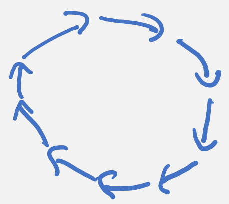
2. Graph Coloring

Hamiltonian Cycle

Directed Hamiltonian Cycle (DHC): A directed cycle is a Hamiltonian cycle if it touches every vertex exactly once.

Input: A directed graph $G = (V, E)$.

Output: Does G have a directed Hamiltonian cycle?



indeg = outdeg = 1 $\forall v$

Theorem. DHC is NP-complete.

Proof. It is easy to check that DHC is in NP.

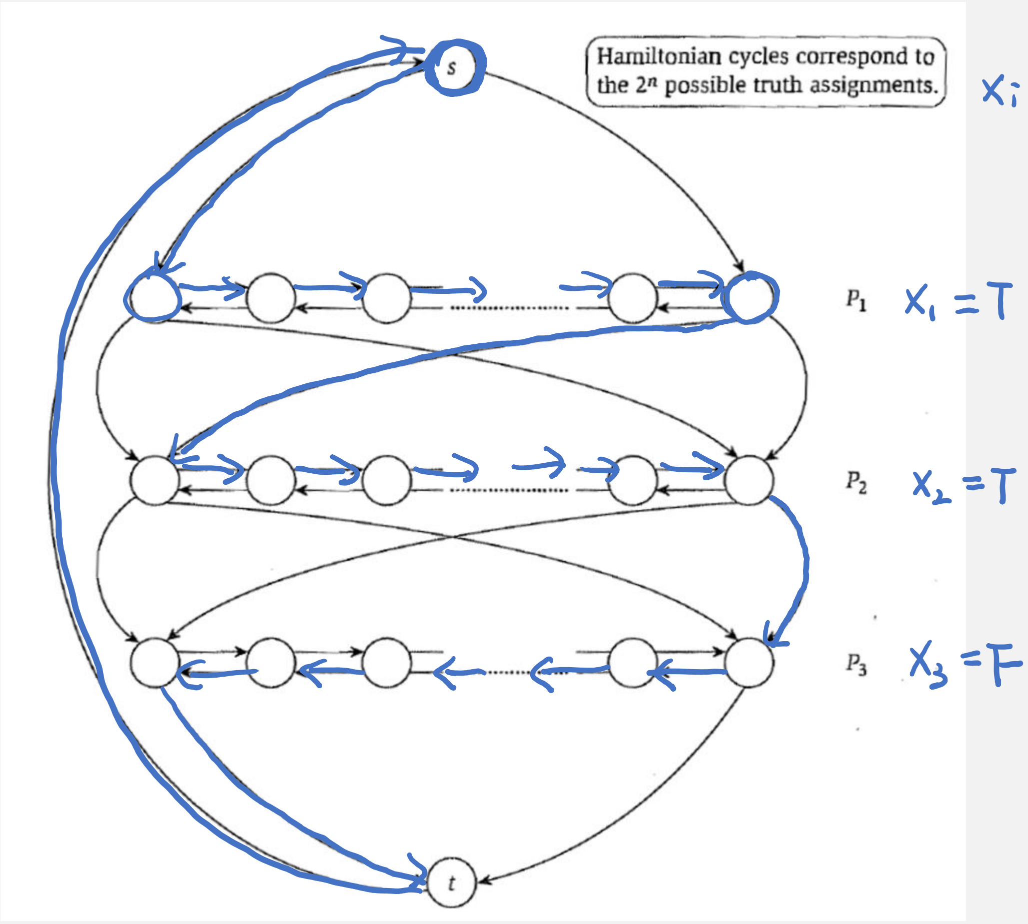
To prove that it is NP-complete, we will prove that 3SAT $\leq_p$ DHC.

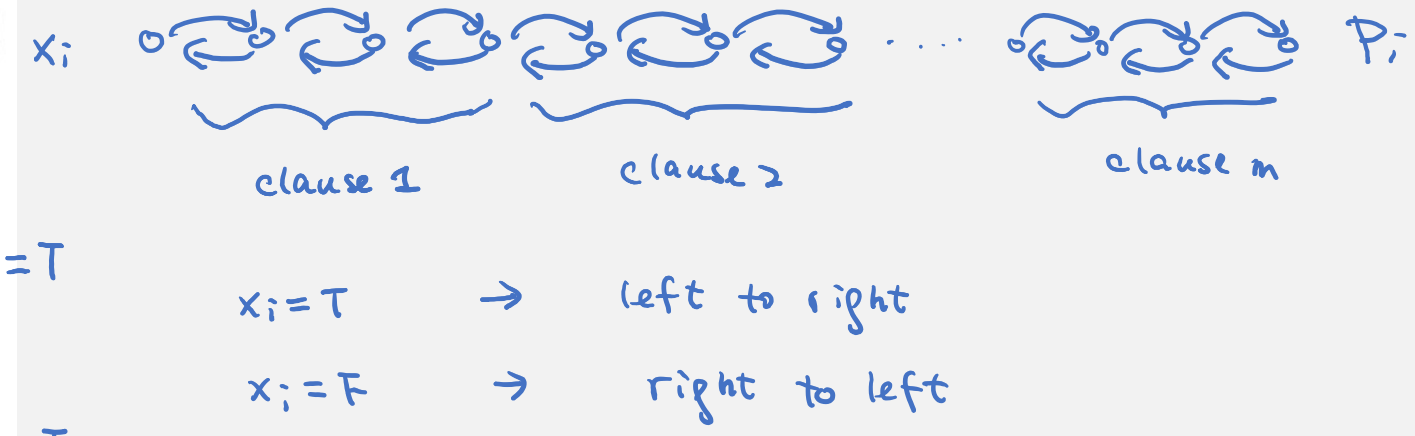
Given a 3SAT instance with n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$, we would like to construct a directed graph G so that the formula is satisfiable iff G has a directed Hamiltonian cycle.

Graph Structure

For the reduction, we need some graph structures for the variables and the truth assignments.

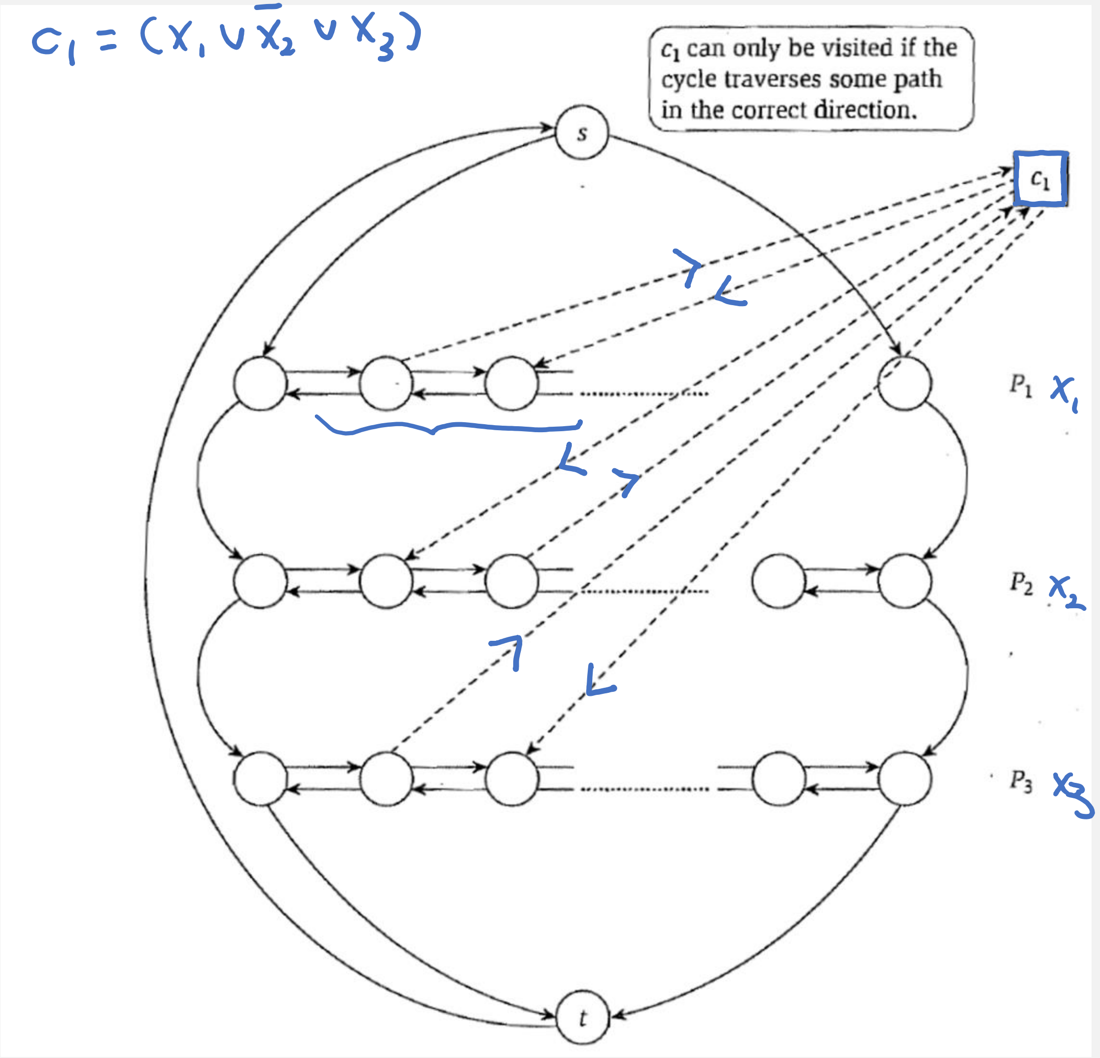
The idea is to create a long “two-way path” for each variable, so that going the path from left to right corresponds to setting the variable to True, while from right to left corresponds to setting it to False.

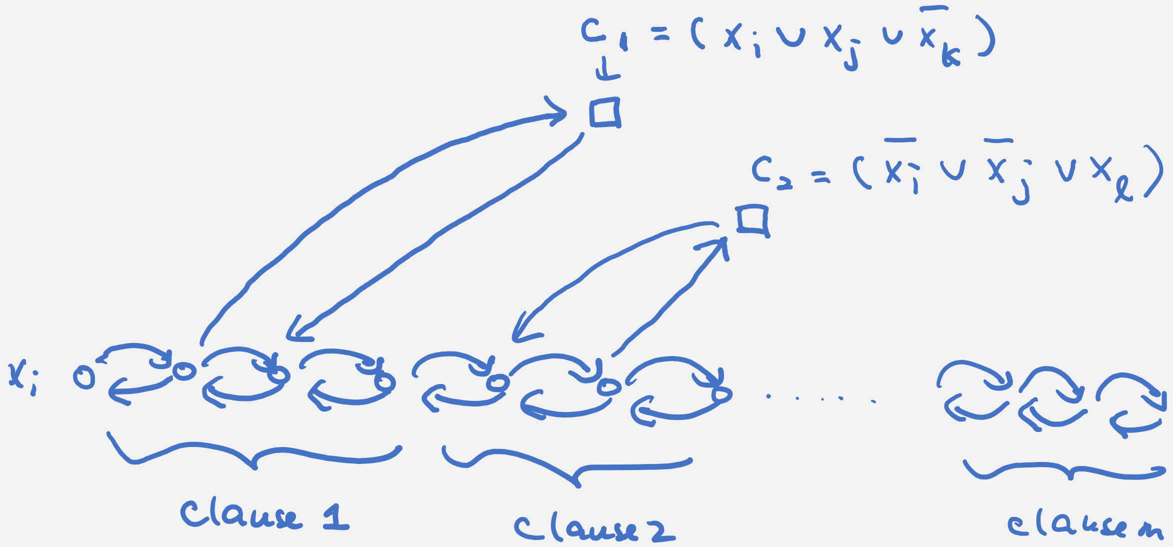




Reduction

Now, we would like to add some clause structures so that only satisfying assignments "survive".





Proof

Claim. The formula is satisfiable if and only if G has a directed Hamiltonian cycle.

proof $\Rightarrow$ satisfying asg.

$x_i = T \rightarrow$ go from left to right on P_i

$x_i = F \rightarrow$ go from right to left on P_i

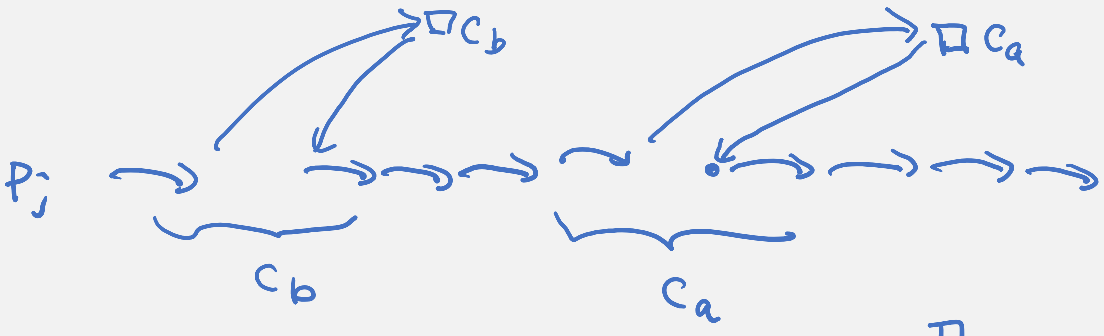
satisfying $\Rightarrow$ for each clause (C_a): one literal (x_j) is satisfied

$\Rightarrow$ we can "detour" to visit c_a in the path P_j

$C_a = (x_i \vee x_j \vee \bar{x}_k)$

$x_j = T$

$c_b = (x_\ell \vee x_j \vee x_k)$

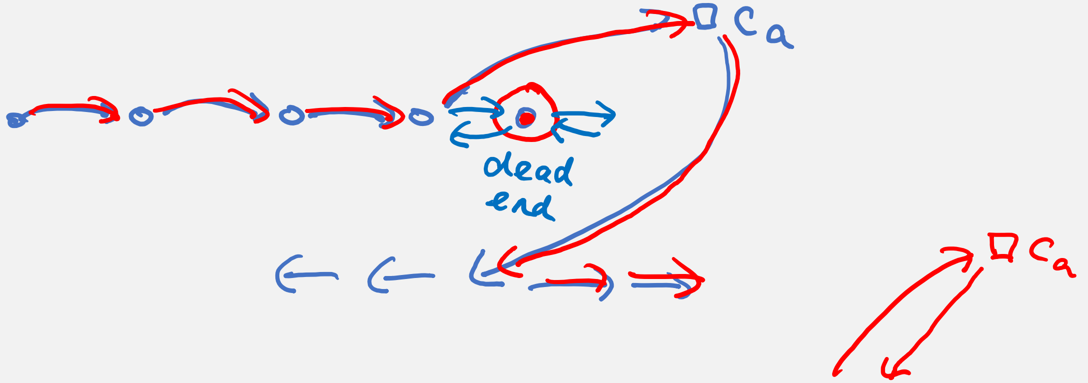


∎

Proof

Claim. The formula is satisfiable if and only if G has a directed Hamiltonian cycle.

$\Leftarrow$) directed HC. claim that it behaves exactly as we wanted.



P_i left to right $x_i = T$

right to left $x_i = F$

since visited every clause C

$\Rightarrow$ one variable/path goes in the correct direction

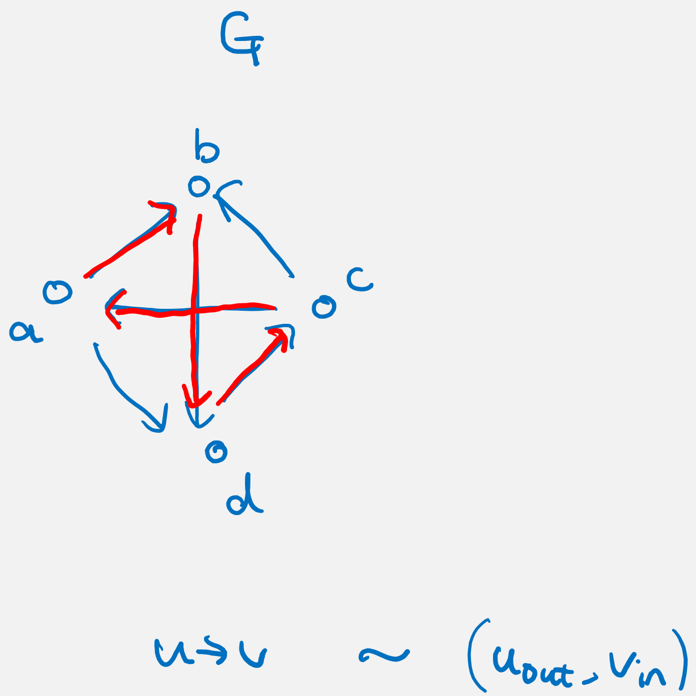
$\Rightarrow$ clause is satisfied.

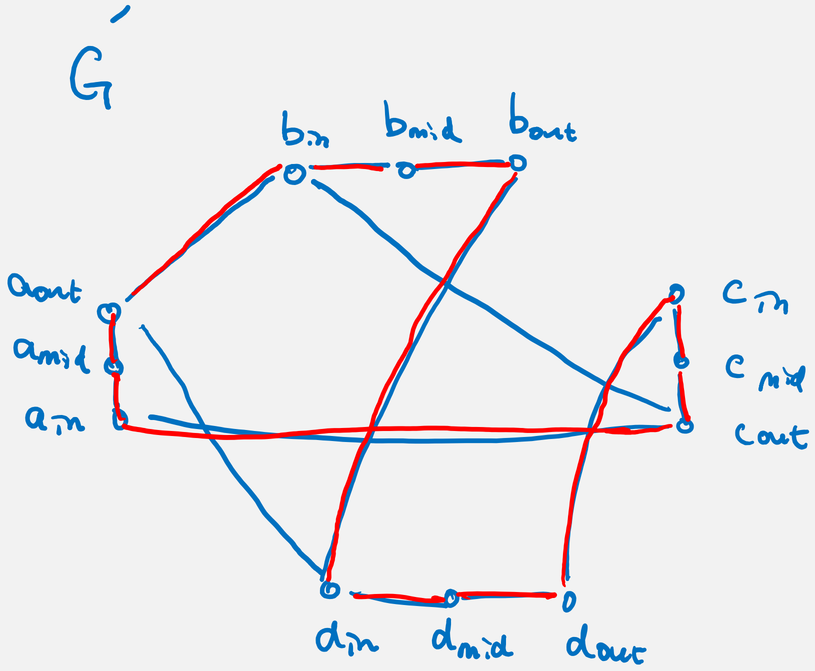
$\square$

Undirected Hamiltonian Cycle

Proposition DHC $\leq_p$ HC.

Proof. The idea is to use a length 3 path to simulate a directed edge.





Proof

Claim. G has a directed Hamiltonian cycle if and only if G' has a Hamiltonian cycle.

proof

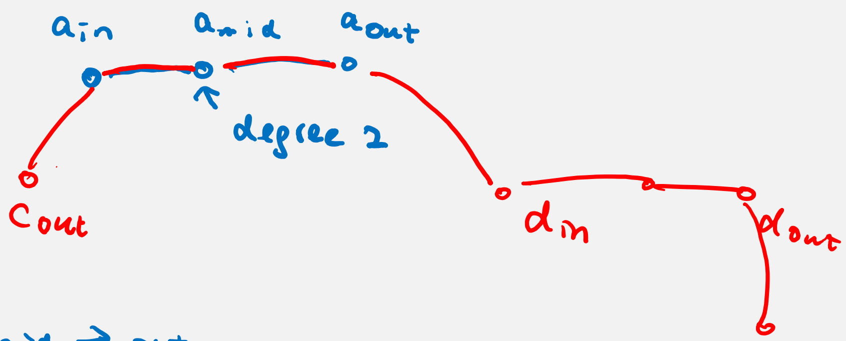
$\Rightarrow$ done

$\Leftarrow$ say start from a_{in}

go to a_{mid}, to a_{out}

$\downarrow$

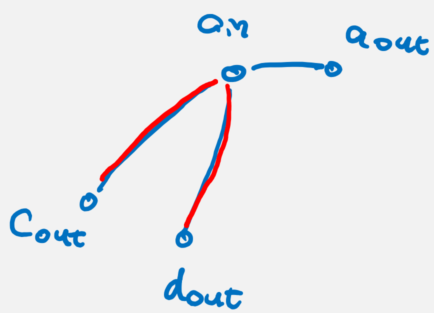
$v_{in} \to mid \to out$



every vertex is visited once

and corresponds to a directed

cycle using the edges (u_{out}, v_{in})



$\square$

Today's Plan

1. Hamiltonian Cycle
2. Graph Coloring

Graph Coloring

Input: An undirected graph $G = (V, E)$, an integer k.

Output: Is it possible to use k colors to color all vertices so that each vertex receives one color and any two adjacent vertices receive different colors?

Graph Structure

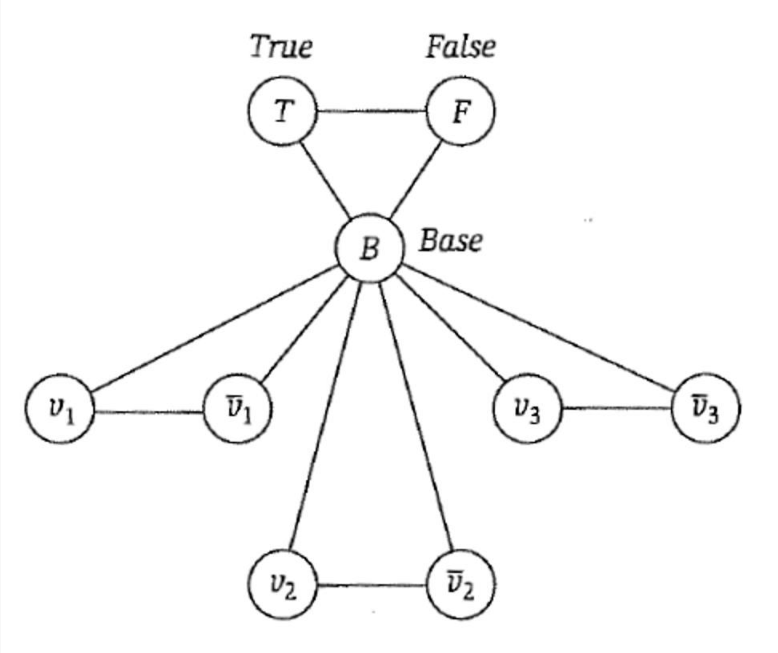
Theorem. 3-Coloring is NP-complete.

Proof. It is easy to check that 3-Coloring is in NP.

To prove that it is NP-complete, we will prove that 3SAT $\leq_p$ 3-Coloring.

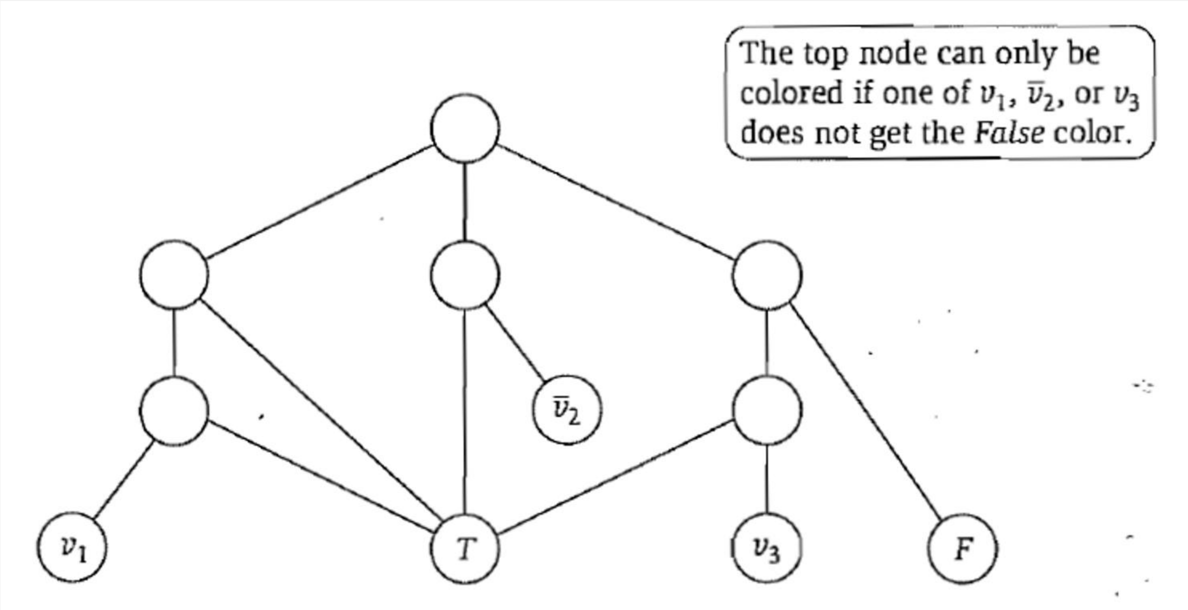
Given a 3SAT instance with n variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$, we would like to construct a graph G so that the formula is satisfiable iff G is 3 colorable.

Idea. Assign one color for True literals and one color for False literals.



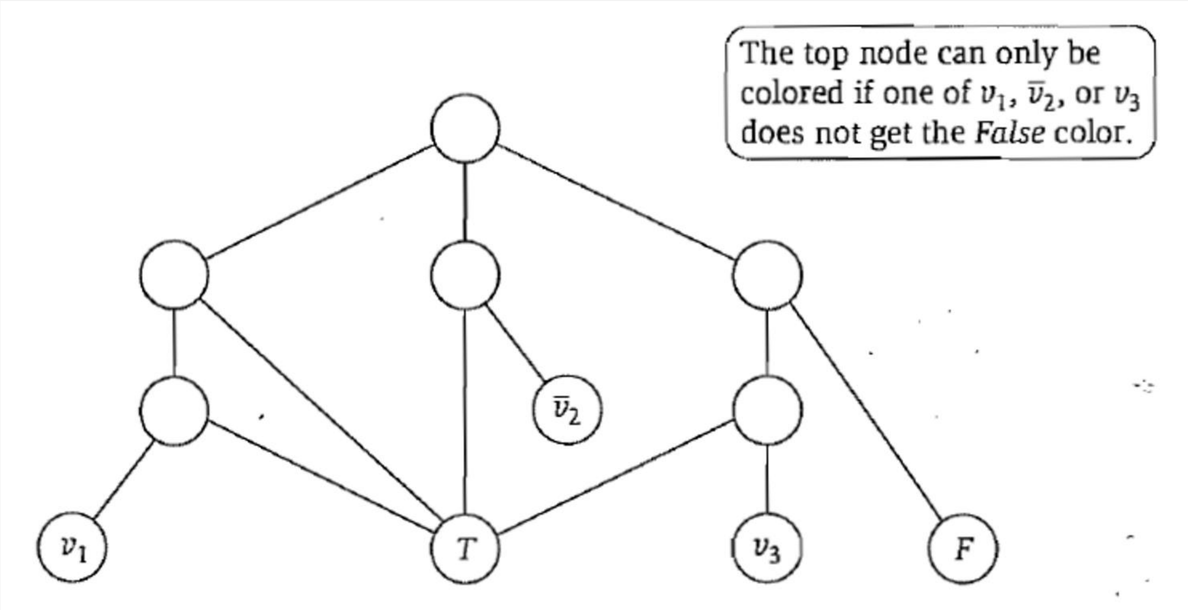
Gadget

<u>Idea</u>: For each clause, we create a "gadget" so that there is a 3-coloring of the gadget if and only if at least one literal in this clause gets the True color.



Gadget

<u>Idea</u>: For each clause, we create a "gadget" so that there is a 3-coloring of the gadget if and only if at least one literal in this clause gets the True color.



Proof

Claim. The formula is satisfiable if and only if G is 3-colorable.

NP-completeness Check List

Suppose we would like to prove a problem X is NP-complete.

1. Briefly explain that X is in NP.
2. State an NP-complete problem Y that we are going to prove Y $\leq_p$X.
3. Describe the reduction in details.
4. Briefly explain that the reduction can be done in polynomial time.
5. State the main "if and only if" claim.
6. Prove each direction of the "if and only if" claim.

~~3SAT~~, VC, IS, Clique, TSP, Knapsack, Subsum, 3-coloring, 3D-matching