Homework Assignment-11 Dhayalini Nagaraj
A20359686

2. Let 2-CNF-SAT be the set of satisfiable boolean formulas in CNF with exactly 2 literals per clause. Show that 2 CNF-SAT $\in$ P. Make your algorithm as efficient as possible.
(Hint: Observe that $x \vee y$ is equivalent to $\neg x \rightarrow y$
Reduce 2-CNF-SAT to an efficiently solvable problem on a directed graph).

Let us construct a di-graph

$D = (V, A)$ as follows

i) For each literal x, we can use two vertices to represent T or F.

So V_x can represent T

$V_{\hat{x}}$ can represent F

ii) Similarly for each clause $a \vee b$, we can add two ~~arcs~~ arcs $(V_{\hat{a}}, V_b)$, $(V_{\hat{b}}, V_a)$ to A.

The 2-CNF-SAT is satisfiable iff there is no literal x such that $x = \text{True} \wedge \hat{x} = \text{True}$

This is in D and there is a path from V_x to $V_{\hat{x}}$ and $V_{\hat{x}}$ to V_x. $V_x, V_{\hat{x}}$ are in the same strong connected component.

The reason is, if $A \rightarrow B$ is true then either $A = \text{False}$ or $A = \text{True}$ and $B = \text{True}$.

So, if $A \rightarrow \hat{A}$ is true then we should have ~~A = False and~~ $A = \text{False}$ and this contradicts with the fact that $\hat{A} \rightarrow A$ is true

So, the algorithm is,

i) Construct the graph D

ii) Find the strong connected component (SCC) of D

iii) For each SCC, we should check if there is some x such that V_x and $V_{\hat{x}}$ are both in it.

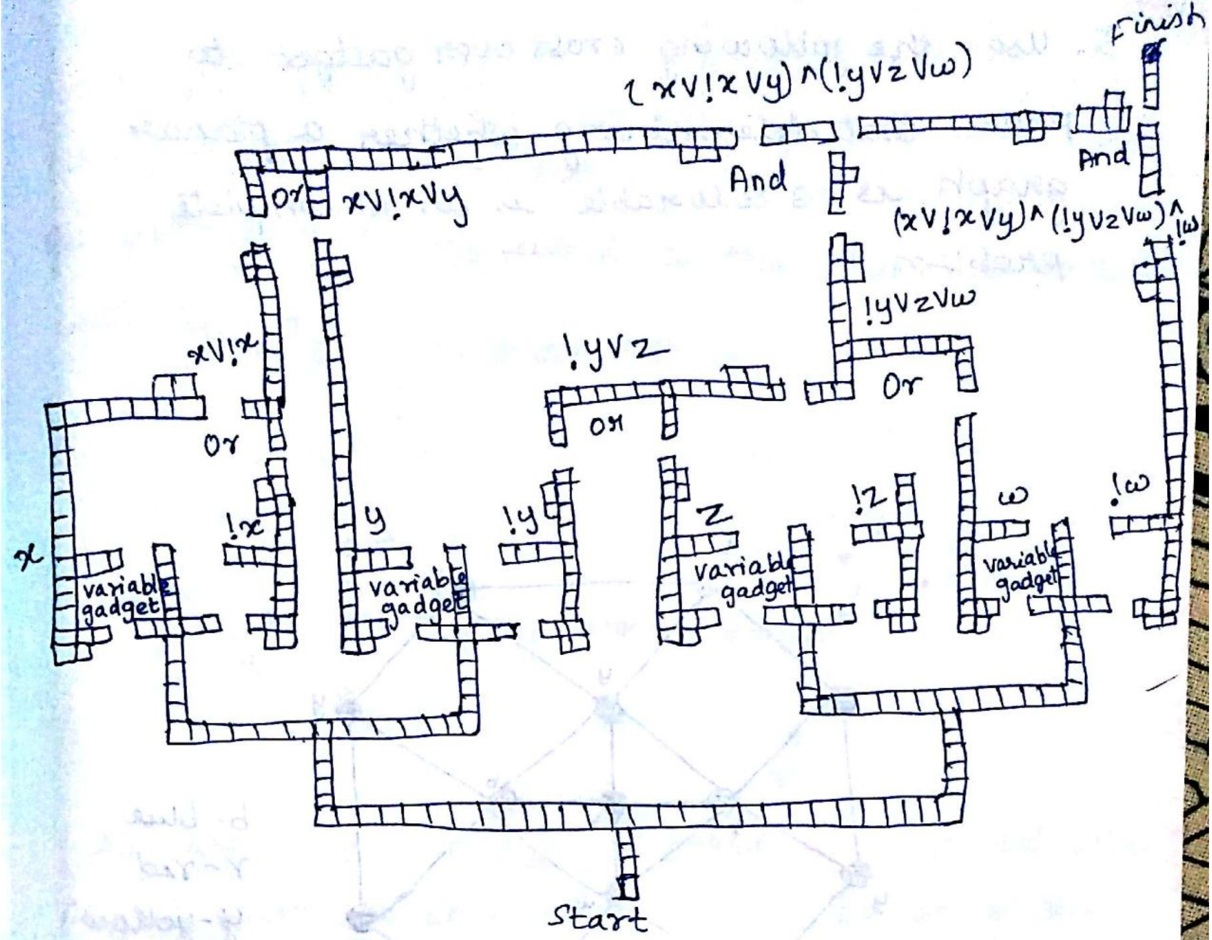
iv) So if we find any then return FALSE else return TRUE.

Scanned by CamScanner

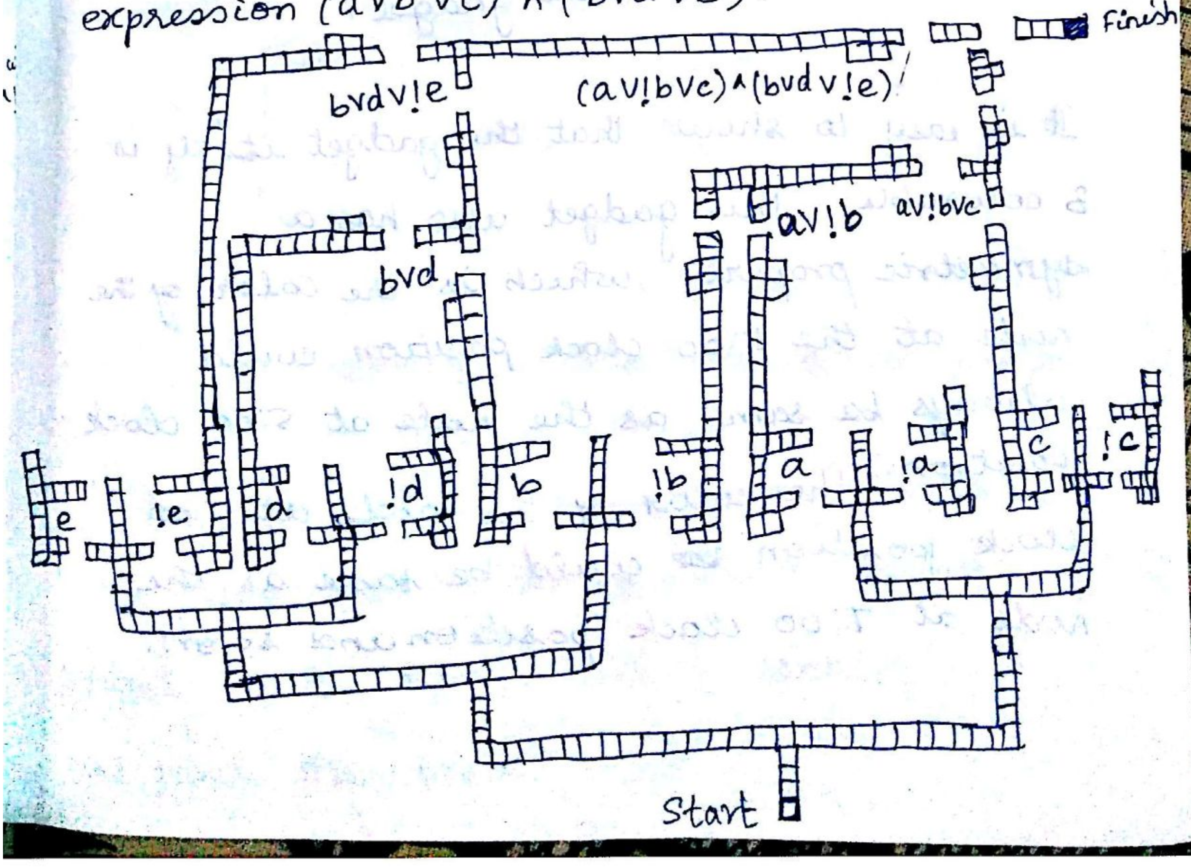
TIME COMPLEXITY: Suppose that there are n literals and m clauses. The construction of graph D costs $O(m+n)$. The cost to find scc $O(m+n)$. To check whether some $V_x, V_{\hat{x}}$ are in one SCC cost $O(n)$.

Totally the time $O(m+n)$.

3. a) Label all the variables, operators and clauses in Figure 10 in the Tipover article.



3b) Draw a Tip over diagram for the 3-SAT expression $(a \vee \bar{b} \vee c) \wedge (b \vee d \vee \bar{e})$.



4. Prove the claim on page 3 of the graph coloring slides (regarding the colorability of the 'or' gadget).

Claim: 3-SAT $\leq_p$ GRAPH 3 COLORING

Proof: Given 3-SAT instance Φ, we construct an instance of 3-COLOR that is 3-colorable iff Φ is satisfiable.

Construction:

i) Create one vertex for each literal

ii) Create 3 new vertices T, F and B. Connect them in a triangle and connect each literal to B.

iii) Connect each literal to its negation.

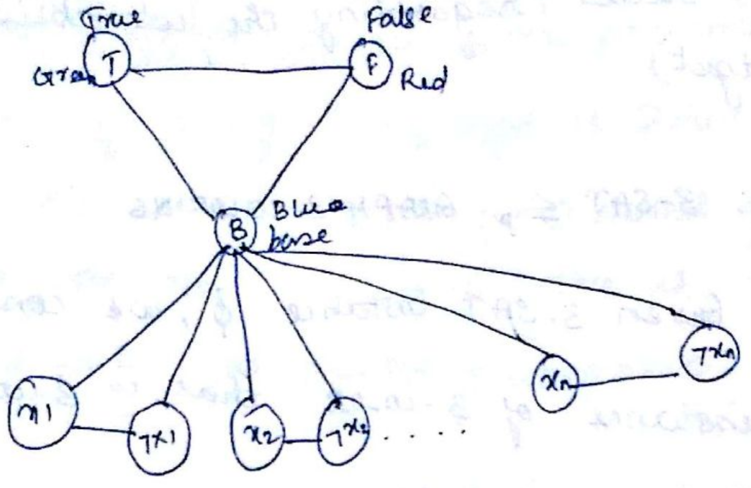
iv) For each clause, attach a gadget of 6 vertices and 13 edges.

Claim: Graph is 3 colorable iff Φ is satisfiable

Pf $\Rightarrow$ Suppose graph is 3 colorable

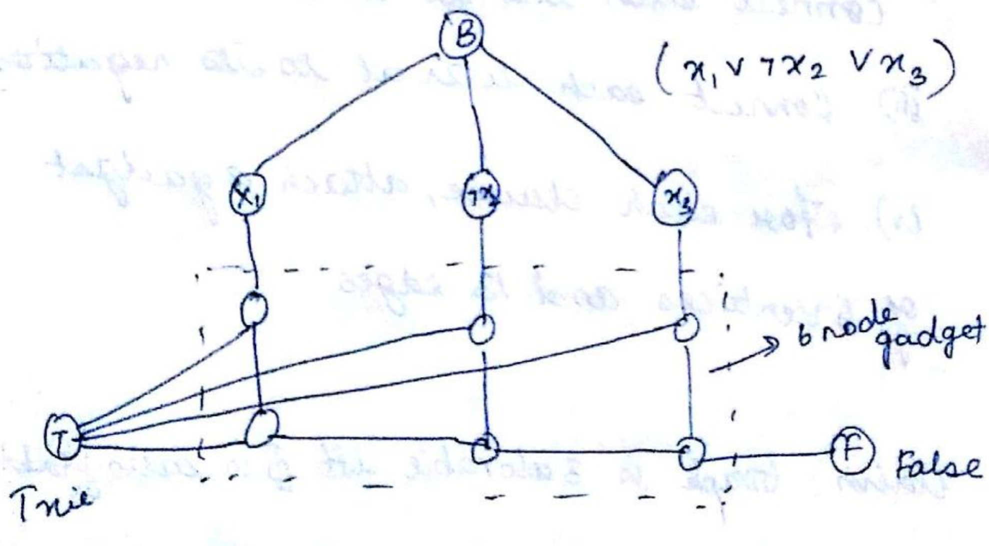
Consider assignment that sets all T literals to true.

(Triangle) ensures each literal is T or F

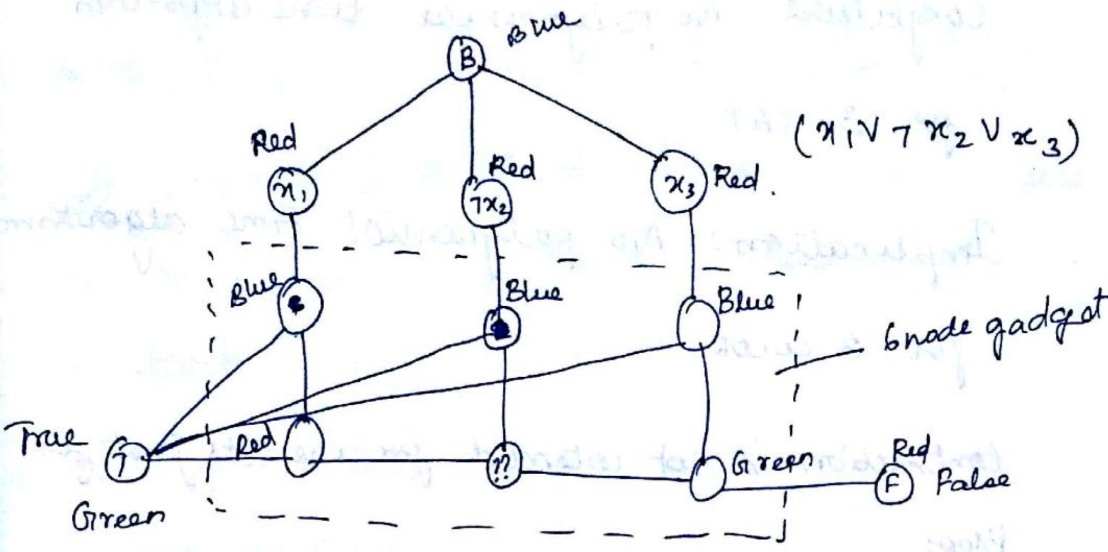


Ensure a literal and its negation are opposites.

[Gadget] ensures atleast one literal in each clause is T



Therefore, Φ is satisfiable



Suppose 3-SAT formula Φ is satisfiable

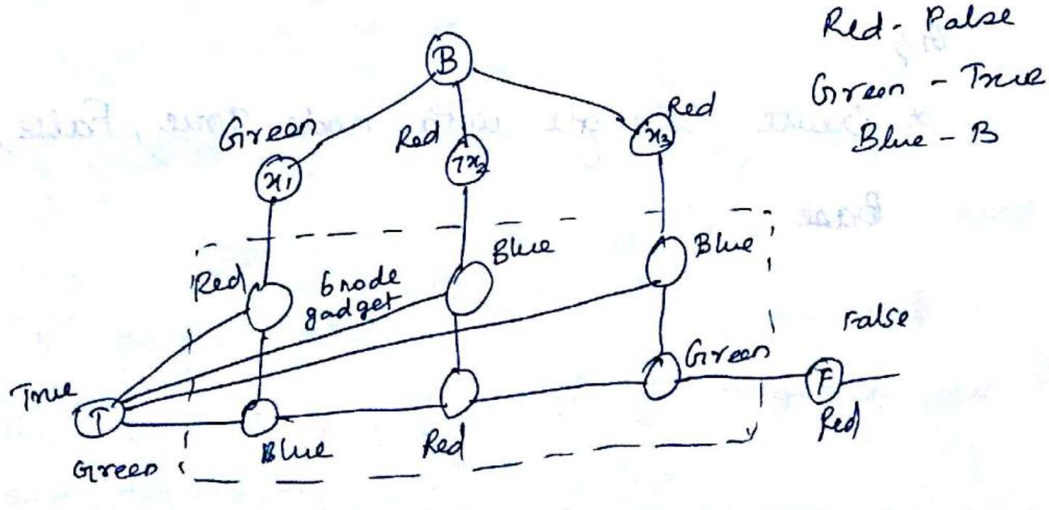
i) color all true literals T and false literals F.

ii) Color vertex below one green vertex F and and vertex below that B

iii) Color remaining middle row vertices B.

iv) color remaining bottom vertices T or F as forced.

Therefore graph is 3-colorable.



Conjecture : No polynomial time algorithm for 3-SAT

Implication: No polynomial time algorithm for 3 COLOR

Contruction is not intended for use, its just for proof.

Hence Proved.

3-coloring is in NP.

Reduction Idea:

Start with 3-SAT formula Φ with n-variables $x_1, x_2 \cdots x_n$ and m clauses $C_1, \dots C_m$. Create graph G_ϕ such that G_ϕ is 3 colorable iff Φ is satisfiable

* Need to establish truth assignment for $x_1, \dots x_n$ via colors for some nodes in G_ϕ

* Create triangle with node True, False, Base.

* For each variable x_i two nodes v_i and $\bar{v}_i$ connected in a triangle with common Base.

* If graph is 3-colored, either v_i or $\bar{v}_i$ gets the same color as True. Interpret this as a truth assignment to v_i

* For each clause $C_j = (a \vee b \vee c)$. create small gadget graph.

  Gadget graph connects to nodes corresponding to a, b, c.

  Needs to implement OR

OR-Gadget Graph.

<u>Property</u>: If a, b, c are colored False in a 3-coloring then output node of OR-gadget has to be colored False.

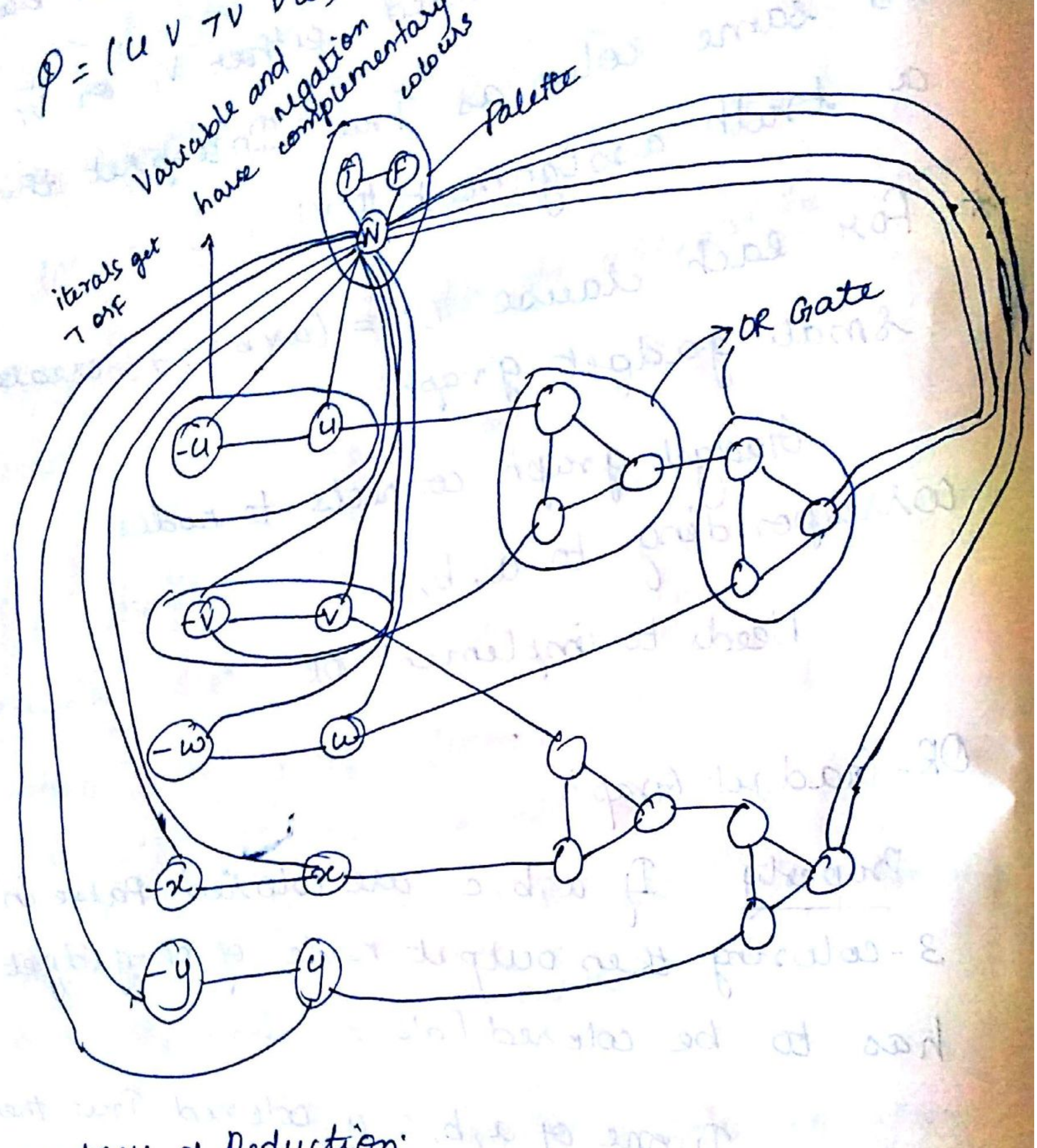
If one of a, b, c is colored True then OR-gadget can be 3-colored such that output node of OR-gadget is colored True.

<u>Reduction</u>:

For each clause $C_j = (a \vee b \vee c)$, add OR-gadget graph with input nodes a, b, c and connect output node of gadget to both False and Base.

eg:

$\phi = (u \vee \neg v \vee w) \wedge (v \vee x \vee \neg y)$



Correctness of Reduction:

ϕ is satisfiable implies G_ϕ is 3 colorable

* if x_i is assigned True, color v_i True and $\bar{v}_i$ False

* For each clause $C_j = (a \vee b \vee c)$ atleast

one of a, b, c is colored True, OR-gadget for c_j can be 3 colored such that output is True.

G_ϕ is 3 colorable implies ϕ is satisfiable

* If v_i is colored True then set x_i to be True, this is legal truth assignment
* Consider any clause $c_j = (a \vee b \vee c)$, it cannot be that all a, b, c are false. If so, output of OR-gadget for c_j has to be colored False but output is connected to Base and False!

Hence proved.

5. Use the following cross over gadget to prove that determining whether a planar graph is 3-colorable is an NP-complete problem.

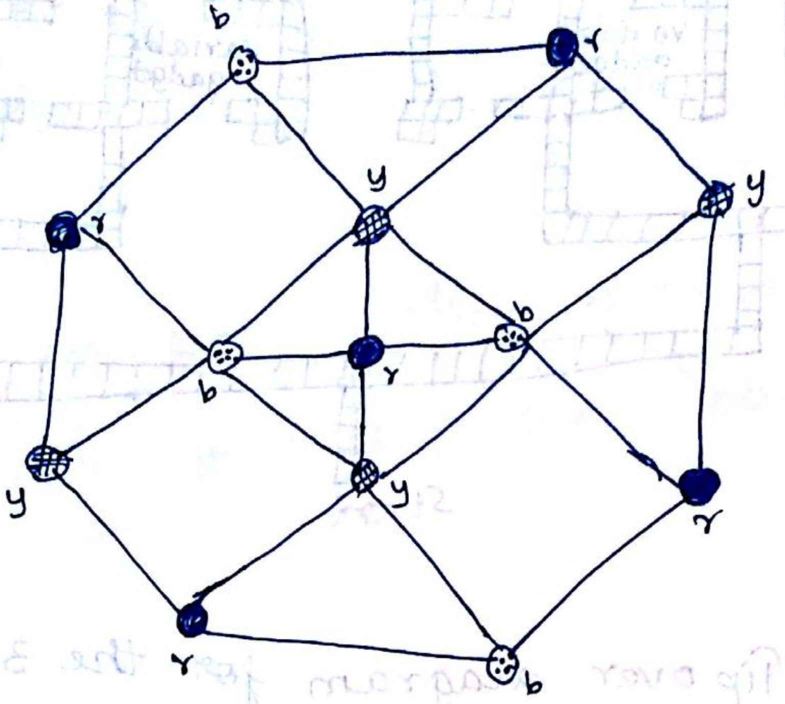
We can reduce 3-colorability of an arbitrary graph to the planar case. Given an undirected graph $G = (V, E)$ ~~pass~~ possibily non planar, embed the graph in the plane arbitrarily, letting edges cross if necessary. We will replace each edge crossing with the planar widget w.

The given cross over gadget should have the properties.

i) Any legal 3 coloring of w, the opposite corners are forced to have same color.

ii) Assignment of colors to the corners such that opposite corners have the same color extends to a 3-coloring of all w.

Scanned by CamScanner

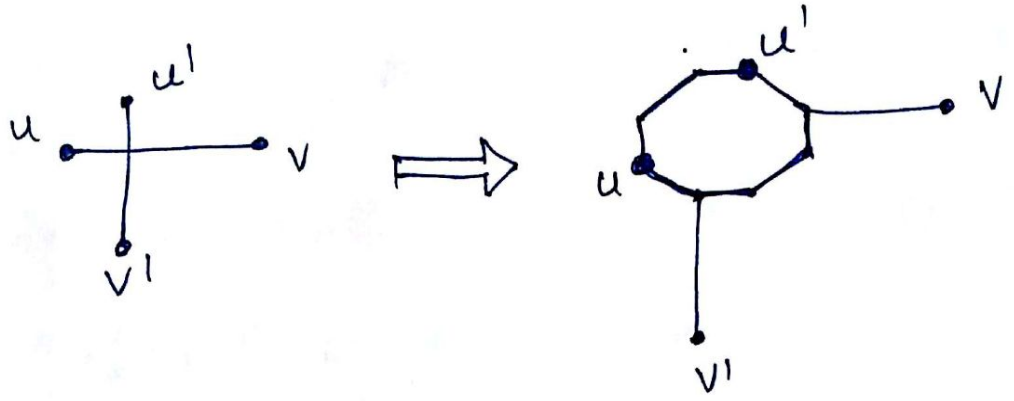


b- blue
r- red
y- yellow

3- colorable gadget.

It is easy to show that the gadget itself is 3 colorable. This gadget also has a symmetric property, which is the color of the node at the 11:00 clock position could always be same as the node at 5:00 clock position. The color of the node at 1:00 clock position ~~be~~ could be same as the node at 7:00 clock position and so on.

So we can always replace a crossed edge (u, v) with u embedded at a corner of the gadget and v connected to the opposite corner with an edge.

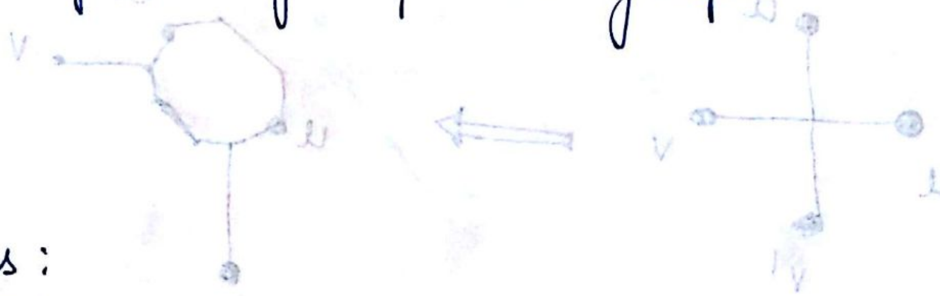


Replacing a cross with the gadget.

This takes $O(|E|^2)$ time. The resulting graph is planar and the symmetric property of the gadget implies that u and v cannot be assigned the same color.

A reduction of the 3-colorability for an arbitrary graph to the same problem for a planar by using this gadget is shown.

3 colorable is NP complete for general graphs, So we can prove that 3-colorable is NP-complete for planar graphs also.



References:

https://cgi.csc.liv.ac.uk/~igor

www.cs.princeton.edu.

www.cs.umd.edu

courses.csail.mit.edu