ASSIGNMENT 3

PROBLEM 1

For this question, let F denote the set of well-formed formulas over a set Prop of propositional variables.

a) Show that the logical equivalence relation, ≡, is an equivalence relation on F.

For a relation to be an equivalence relation is must be reflexive, symmetric and transitive in nature.

F is the smallest set of words over $\sum$ which $\sum$ = PROP $\cup \{ \perp, T, \rightarrow, \leftarrow, \wedge, \vee, \neg, (,) \}$
Therefore,
R := $\{(\varphi, \psi)$, with $\varphi \equiv \psi$ and $\varphi, \psi \in F\}$

1) For all formulas $\varphi \in F$, $\varphi \equiv \varphi$ due to $v(\varphi) = v(\varphi)$
   Hence, for all the formulas $\varphi \in F$, (φ, φ) must in R
   Therefore, R is reflexive.
2) Suppose there is $(\varphi, \psi) \in R$, where $\varphi, \psi \in F$.
   Due to $(\varphi, \psi) \in R$, we get that (φ, ψ)
   Thus, $v(\varphi) = v(\psi)$
   Because $v(\varphi) = v(\psi)$ which implies that $v(\varphi) = v(\psi)$, $\varphi \equiv \psi$ and $(\varphi, \psi) \in R$
   Hence, if there is $(\varphi, \psi) \in R$, then $(\varphi, \psi) \in R$
   Therefore, R is symmetric.
3) Suppose there are $(\varphi, \psi), (\psi, \phi) \in R$.
   Since $\varphi \equiv \psi$, $\psi \equiv \phi$, we get $v(\varphi) = v(\psi)$ and $v(\psi) = v(\phi)$
   Hence, $v(\varphi) = v(\psi) = v(\phi)$
   Thus, $\varphi \equiv \phi$ and further means that $(\varphi, \phi) \in R$
   Hence, if there are $(\varphi, \psi), (\psi, \phi) \in R$, then $(\varphi, \phi) \in R$
   Therefore, R is transitive.

We have R as reflexive, symmetric and transitive therefore R is equivalence relation.

b) List four elements in [⊥], the equivalence class of ⊥.

Four elements in [⊥] are as follows :

1) $\neg T$
2) $\neg \neg \perp$
3) $\perp \wedge T$
4) $\perp \vee \perp$

c) For all $\varphi, \varphi', \psi, \psi' \in F$ with $\varphi \equiv \varphi'$ and $\psi \equiv \psi'$; show that:

i. $\neg\varphi \equiv \neg\varphi'$

$\varphi \equiv \varphi'$, so $v(\varphi) = v(\varphi')$
$v(\neg\varphi) = !v(\varphi)$ and $v(\neg\varphi') = !v(\varphi')$.
We have $v(\varphi) = v(\varphi')$, $!v(\varphi) = !v(\varphi')$
Also $!v(\varphi) = !v(\varphi')$, $v(\neg\varphi) = v(\neg\varphi')$
$v(\neg\varphi) = v(\neg\varphi')$ means $\neg\varphi \equiv \neg\varphi'$
Hence proved.

ii. $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$

$\varphi \equiv \varphi'$ and $\psi \equiv \psi'$, so $v(\varphi) \equiv v(\varphi')$ and $v(\psi) \equiv v(\psi')$
$v(\varphi \wedge \psi) = v(\varphi)$ && $v(\psi)$ and $v(\varphi' \wedge \psi') = v(\varphi')$ && $v(\psi')$
We have $v(\varphi) = v(\varphi')$ and $v(\psi) = v(\psi')$, $v(\varphi)$ && $v(\psi) = v(\varphi')$ && $v(\psi')$
Also $v(\varphi)$ && $v(\psi) = v(\varphi')$ && $v(\psi')$, $v(\varphi \wedge \psi) = v(\varphi' \wedge \psi')$
Now $v(\varphi \wedge \psi) = v(\varphi' \wedge \psi')$ means $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$
Hence proved.

iii. $\varphi \vee \psi \equiv \varphi' \vee \psi'$

$\varphi \equiv \varphi'$ and $\psi \equiv \psi'$, so $v(\varphi) \equiv v(\varphi')$ and $v(\psi) \equiv v(\psi')$
$v(\varphi \vee \psi) = v(\varphi) \mid\mid v(\psi)$ and $v(\varphi' \vee \psi') = v(\varphi') \mid\mid v(\psi')$
We have $v(\varphi) = v(\varphi')$ and $v(\psi) = v(\psi')$, $v(\varphi) \mid\mid v(\psi) = v(\varphi') \mid\mid v(\psi')$
Also $v(\varphi) \mid\mid v(\psi) = v(\varphi') \mid\mid v(\psi')$, $v(\varphi \vee \psi) = v(\varphi' \vee \psi')$
Now $v(\varphi \vee \psi) = v(\varphi' \vee \psi')$ means $\varphi \vee \psi \equiv \varphi' \vee \psi'$
Hence proved.

d) Show that $F_{\equiv}$ together with the operations defined above forms a Boolean Algebra.

For all $[\phi], [\varphi], [\psi] \in F_{\equiv}$

For commutative:

$[\phi] \wedge [\psi]$	$= [\phi \wedge \psi]$	[using given operations]
	$= [\psi \wedge \phi]$	[using Commutative Law]
	$= [\psi] \wedge [\phi]$	[using given operations]
$[\phi] \vee [\psi]$	$= [\phi \vee \psi]$	[using given operations]
	$= [\psi \vee \phi]$	[using Commutative Law]
	$= [\psi] \vee [\phi]$	[using given operations]

Therefore, commutative law holds.

For associative:

$([\phi] \wedge [\psi]) \wedge [\varphi]$	$= [\phi \wedge \psi] \wedge [\varphi]$	[using given operations]
	$= [(\phi \wedge \psi) \wedge \varphi]$	[using given operations]
	$= [\phi \wedge (\psi \wedge \varphi)]$	[using Associative Law]
	$= [\phi] \wedge [\psi \wedge \varphi]$	[using given operations]
	$= [\phi] \wedge ([\psi] \wedge [\varphi])$	[using given operations]
$([\phi] \vee [\psi]) \vee [\varphi]$	$= [\phi \vee \psi] \vee [\varphi]$	[using given operations]
	$= [(\phi \vee \psi) \vee \varphi]$	[using given operations]
	$= [\phi \vee (\psi \vee \varphi)]$	[using Associative Law]
	$= [\phi] \vee [\psi \vee \varphi]$	[using given operations]
	$= [\phi] \vee ([\psi] \vee [\varphi])$	[using given operations]

Therefore, associative law holds.

For distributive:

$[\phi] \wedge ([\psi] \vee [\varphi])$	$= [\phi] \wedge [\psi \vee \varphi]$	[using given operations]
	$= [\phi \wedge (\psi \vee \varphi)]$	[using given operations]
	$= [(\phi \wedge \psi) \vee (\phi \wedge \varphi)]$	[using Distributive Law]
	$= [\phi \wedge \psi] \vee [\phi \wedge \varphi]$	[using given operations]
	$= ([\phi] \wedge [\psi]) \vee ([\phi] \wedge [\varphi])$	[using given operations]
$[\phi] \vee ([\psi] \wedge [\varphi])$	$= [\phi][\psi \wedge \varphi]$	[using given operations]
	$= [\phi \vee (\psi \wedge \varphi)]$	[using given operations]
	$= [(\phi \vee \psi) \wedge (\phi\varphi)]$	[using Distributive Law]
	$= [\phi \vee \psi] \wedge [\phi \vee \varphi]$	[using given operations]
	$= ([\phi] \vee [\psi]) \wedge ([\phi] \vee [\varphi])$	[using given operations]

Therefore, distributive law holds.

For identity:

$[\varphi] \wedge [T]$	$= [\varphi \wedge T]$	[using given operations]
	$= [\varphi]$	[using Identity Law]
$[\varphi] \vee [\bot]$	$= [\varphi \vee \bot]$	[using given operations]
	$= [\varphi]$	[using Identity Law]

Therefore, identity law holds.

For complementation:

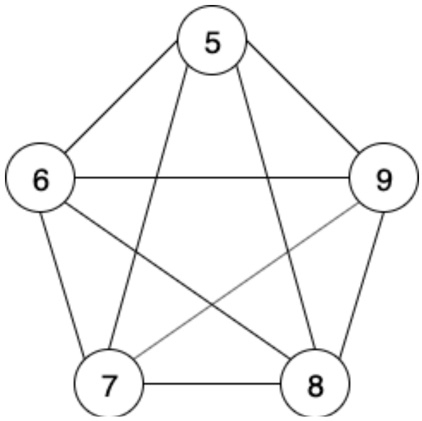
$[\varphi] \vee [\varphi]'$	$= [\varphi] \vee [\neg\varphi]$	[using given operations]
	$= [\varphi \vee \neg\varphi]$	[using given operations]
	$= [T]$	[using Complementation Law]
$[\varphi] \wedge [\varphi]'$	$= [\varphi] \wedge [\neg\varphi]$	[using given operations]
	$= [\varphi \wedge \neg\varphi]$	[using given operations]
	$= [\bot]$	[using Complementation Law]

Therefore, complementation law holds.

PROBLEM 2

a) Give an argument to show that the Petersen graph does not contain a subdivision of K_5.

In Petersen graph the maximum degree of each vertex is 3 but for a K_5 graph the degree of all the vertices must be equal to 5.
We know the degree of vertices of all the subgraphs is equal to the degree of the graph itself or exactly two. Since we cannot increase the degree of the given Petersen graph from 3 to 5, Hence cannot show it has a subdivision of K_5.
And yes by doing edge contractions we can find minor of K_5 but cannot find a subdivision of K_5 graph.

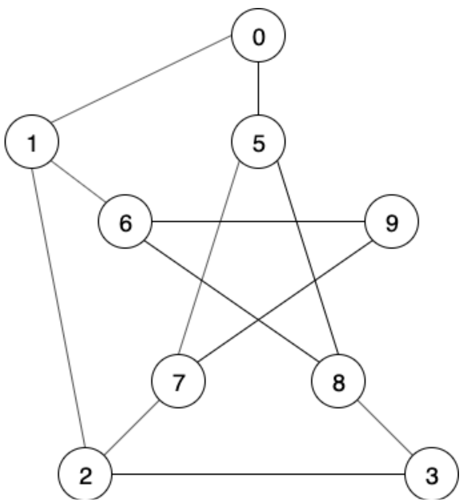


Hence we can find a K_5 minor and not subdivision of K5.

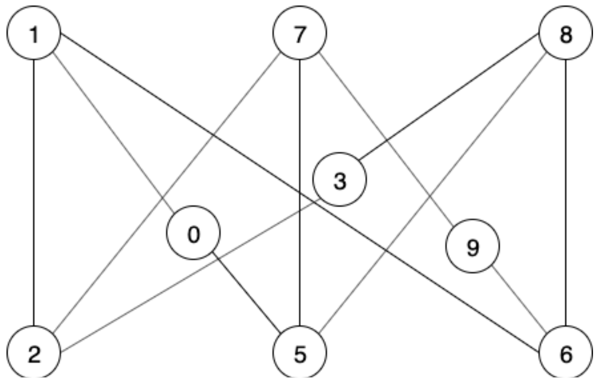
b) Show that the Petersen graph contains a subdivision of $K_{3,3}$.

To show the Petersen graph contains a subdivision of $K_{3,3}$
Firstly we will delete vertex 4 and all the edges from vertices 0, 9 and 3 which directly connects with the vertex 4.

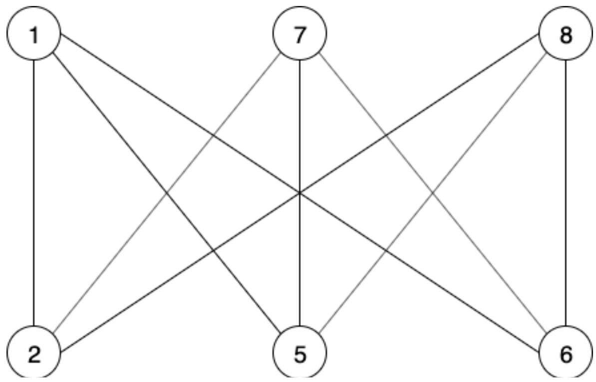
Hence we get



Now rearranging the above graph into a more recognizable graph, we get,



Now again we can easily ignore/delete vertices namely 3, 0 and 9, we get,



Here vertex 1 connects with vertices 2, 5 and 6. vertex 7 connects with vertices 2, 5 and 6 and vertex 8 connects with vertices 2, 5 and 6. Also vertices 1, 7, 8 and vertices 2, 5, 6 do not have any connecting edges among themselves respectively.

Hence the above graph is in $K_{3,3}$.

Therefore the Petersen graph is a subdivision of $K_{3,3}$ and is non-planar in nature.

PROBLEM 3

Harry would like to take each of the following subjects: Defence against the Dark Arts; Potions; Herbology; Transfiguration; and Charms. Unfortunately some of the classes clash, meaning Harry cannot take them both. The list of clashes are:

- **Defence against the Dark Arts clashes with Potions and Charms**
- **Potions also clashes with Herbology**
- **Herbology also clashes with Transfiguration, and**
- **Transfiguration also clashes with Charms.**

Harry would like to know the maximum number of classes he can take.

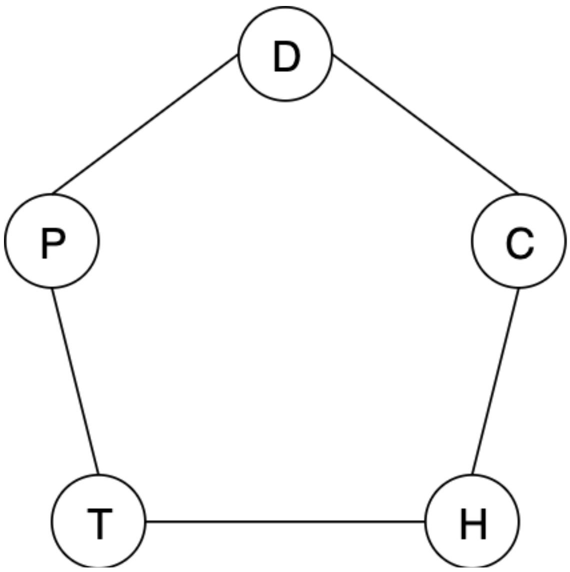
a) Model this as a graph problem. Remember to:

i. Clearly define the vertices and edges of your graph.

$\Rightarrow$ Vertices are defined as different subjects.

D : Defence against the Dark Arts
P : Potions
H : Herbology
T : Transfiguration
C : Charms

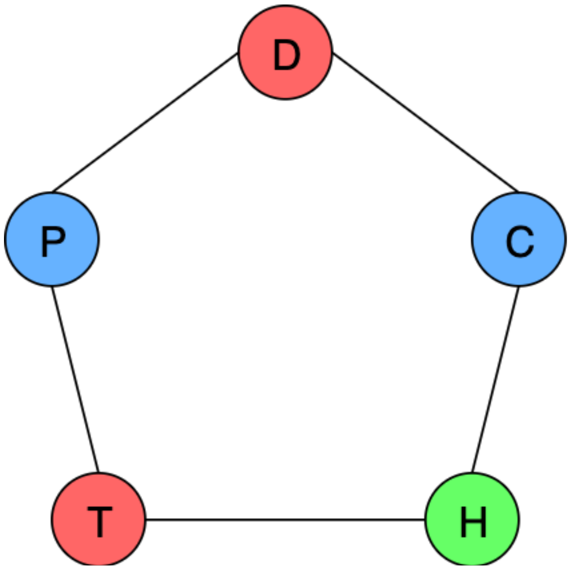
$\Rightarrow$ Edges between the two vertices are defined as the clashes between the subjects.



ii. **State the associated graph problem that you need to solve.**

The graph problem here is to find minimum number of vertices which are not directly connected with each other by edges.
In other words we will assign colours to each vertex and find the chromatic number of the graph formed.

b) Give the solution to the graph problem corresponding to this scenario; and solve Harry's problem.



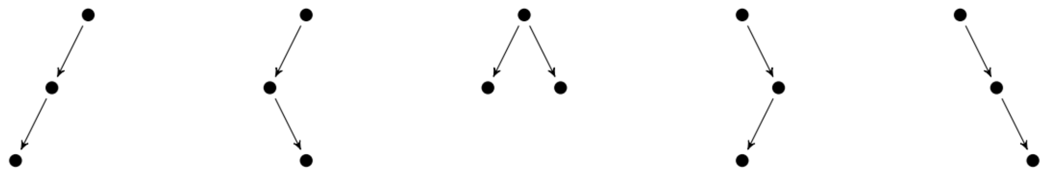
Let's start with assigning Red to D, then we can assign blue to P and C
Now we can assign Red and Green to either T or H.

Since max number of vertices with same colour is two. i.e. chromatic number is 2

Therefore Harry can take at most 2 subjects at the same time without clashes.

PROBLEM 4

Recall from Assignment 2 the definition of a binary tree data structure: either an empty tree, or a node with two children that are trees. Let T(n) denote the number of binary trees with n nodes. For example T(3) = 5 because there are five binary trees with three nodes:



a) Using the recursive definition of a binary tree structure, or otherwise, derive a recurrence equation for T(n).

Recurrence equation of a binary tree structure

$$T(n) = \sum_{i=0}^{n-1} T(i)T(n-i-1)$$

A full binary tree is a non-empty binary tree where every node has either two non-empty children (i.e. is a fully-internal node) or two empty children (i.e. is a leaf).

b) Using observations from Assignment 2, or otherwise, explain why a full binary tree must have an odd number of nodes.

A full binary tree has an odd number of nodes.

At level 0, (at the root), we have 1 node.
At level 1, we have 2 nodes.
At level 2, we have $2^2 = 4$ nodes.
At level 3, we have $2^3 = 8$ nodes
At level n-1, we have 2^{n-1} nodes.
At all other depths other than at level 0, we have even number of nodes. At level 0, we have 1 node. So total number is always an odd number.

Total = $1 + 2 + 2^2 + 2^3 + ... + 2^{n-1}$
$\Rightarrow$ $2^n - 1$

Therefore it is an odd number.

c) Let B(n) denote the number of full binary trees with n nodes. Derive an expression for B(n), involving of T(n’) where n’ ≤ n. Hint: Relate the internal nodes of a full binary tree to T(n).

Given,
B(n) is the number of full binary tree with n nodes.
T(n) is the number of binary tree trees with n nodes.
T(n’) where n’ ≤ n

$\Rightarrow$ B(n) = T(n’)
which means number of full binary trees with 2^n - 1 nodes is equal to the number of binary tree with n’ nodes where n’ ≤ n.

Hence
$\Rightarrow$ $B(2^n - 1) = T(n')$
$\Rightarrow$ $B(2^n) = T(n'+1)$
$\Rightarrow$ $B(n) = T(\log_2(n'+1))$

Since we cannot have negative or zero trees with n nodes. Therefore n’ > 1

d) Using your answer for part (c), give an expression for F(n).

Using precisely each propositional variable exactly one time.
Therefore, the formula must contain n variables and n - 1 of ($\wedge$, $\vee$).
And, each variables p can be p or $\neg$p
Hence, there are 2^{n-1}. 2^n. n! formulas without parentheses.

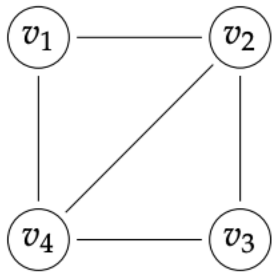
Suppose there is a full binary tree with n leaves represent n variables.
As we know if there are n leaves, the number of nodes is (2^n - 1).
And, each fully internal nodes represent the calculation ($\wedge$, $\vee$) between its children nodes, variables if it is leaf, or the result of calculation if it is another parent node.

The order of fully internal nodes from bottom to top represent the order of calculations.

Hence, there are B(2^n-1) orders of calculations.
So,
$\Rightarrow$ $F(n) = B(2^n - 1)\ 2^{n-1}.\ 2^n.\ n!$
$= B(2^n - 1)\ 2^{2n-1}.\ n!$

PROBLEM 5

Consider the following graph:



and consider the following process:

- **Initially, start at v_1.**
- **At each time step, choose one of the vertices adjacent to your current location uniformly at random, and move there.**

Let $p_1(n)$, $p_2(n)$, $p_3(n)$, $p_4(n)$ be the probability your location after n time steps is v_1, v_2, v_3, or v_4 respectively. So $p_1(0) = 1$ and $p_2(0) = p_3(0) = p_4(0) = 0$.

a) **Express $p_1(n + 1)$, $p_2(n + 1)$, $p_3(n + 1)$, and $p_4(n + 1)$ in terms of $p_1(n)$, $p_2(n)$, $p_3(n)$, and $p_4(n)$.**

$\Rightarrow \quad p_1(n+1) = \frac{1}{3} p_2(n) + \frac{1}{3} p_4(n)$

$\Rightarrow \quad p_2(n+1) = \frac{1}{2} p_1(n) + \frac{1}{2} p_3(n) + \frac{1}{3} p_4(n)$

$\Rightarrow \quad p_3(n+1) = \frac{1}{3} p_2(n) + \frac{1}{3} p_4(n)$

$\Rightarrow \quad p_4(n+1) = \frac{1}{2} p_1(n) + \frac{1}{3} p_2(n) + \frac{1}{2} p_3(n)$

b) **As n gets larger, each $p_i(n)$ converges to a single value (called the steady state) which can be determined by setting $p_i(n + 1) = p_i(n)$ in the above equations. Determine the steady state probabilities for all vertices.**

We have :-

$p_1(n) = \frac{1}{3} p_2(n) + \frac{1}{3} p_4(n)$

$p_2(n) = \frac{1}{2} p_1(n) + \frac{1}{2} p_3(n) + \frac{1}{3} p_4(n)$

$p_3(n) = \frac{1}{3} p_2(n) + \frac{1}{3} p_4(n)$

$p_4(n) = \frac{1}{2} p_1(n) + \frac{1}{3} p_2(n) + \frac{1}{2} p_3(n)$

$\Rightarrow \quad p_4(n) - p_2(n) = \frac{1}{3} p_2(n) - \frac{1}{3} p_4(n)$

$\Rightarrow \quad \frac{4}{3} p_4(n) = \frac{4}{3} p_2(n)$

Hence $p_4(n) = p_2(n)$

Substituting values of $p_4(n)$ and $p_2(n)$ in the above equations, we get,

$p_1(n) = \frac{2}{3} p_2(n)$

$p_3(n) = \frac{2}{3} p_2(n)$

Hence $p_1(n) = p_3(n)$

$\Rightarrow \quad p_1(n) + p_2(n) + p_3(n) + p_4(n) = 1$

$\Rightarrow \quad \frac{10}{3} p_2(n) = 1$

$\Rightarrow \quad p_2(n) = \frac{3}{10}$

Substituting $p_2(n) = \frac{3}{10}$ in the above equations, we get,

$p_1(n) = \frac{1}{5}$

$p_2(n) = \frac{3}{10}$

$p_3(n) = \frac{1}{5}$

$p_4(n) = \frac{3}{10}$

c) The distance between any two vertices is the length of the shortest path between them. What is your expected distance from v_1 in the steady state?

$\Rightarrow \quad p_1(n) + p_2(n) + p_3(n) + p_4(n) = \frac{1}{5} + \frac{3}{10} + \frac{1}{5} + \frac{3}{10}$

$\Rightarrow \quad p_1(n) + p_2(n) + p_3(n) + p_4(n) = \frac{2+3+2+3}{10}$

$\Rightarrow \quad p_1(n) + p_2(n) + p_3(n) + p_4(n) = 1$