COMP 9020: Foundations of Computer Science

Assignment 3

Name: Hongpei Luo

zID: z5246892

Problem 1:

(a) To prove the logical equivalence relation $\equiv$ is an equivalence relation on F, we need to prove that the logical equivalence relation can be reflexive, symmetric and trasitive.

Assume there has a logical equivalence relation L, $a, b, c \in F$ and there exists $L \subseteq F \times F$. When $a \equiv a$, according to the definition of logical equivalence relation, there exists $v(a) = v(a)$, then $(a, a) \in L$, according to the definition of equivalence relation, L is reflexive. For $a, b \in F$, when $a \equiv b$, according to the definition of logical equivalence relation, there exisits $v(a) = v(b)$, so $(a, b) \in L$, likewise, we can know $(b, a) \in L$, hence, L is symmetric. When $a \equiv b$, $b \equiv c$, according to the definition of logical equivalence relation, there exists $v(a) = v(b)$, $v(b) = v(c)$, therefore, $v(a) = v(c)$, which means $a \equiv c$, then $(a, b) \in L$, $(b, c) \in L$ and $(a, c) \in L$. So, L is transitive.

Above, L is reflexive, symmetric and transitive. Therefore, the logical equivalence relation $\equiv$ is an equivalence relation of F.

(b) Because the truth valuation of $v(\bot)$ is false, we can list four elements with truth valuation is false: $\bot \wedge p$, $\bot \vee \bot$, $\bot \wedge q$, $\bot$.

(c) (i) Because $\varphi \equiv \varphi'$, according to the definition of truth valuation, there has $v(\varphi) = v(\varphi')$, so $!v(\varphi) = !v(\varphi')$. Since $!v(\varphi) = v(\neg\varphi)$, $!v(\varphi') = v(\neg\varphi')$, it's obvious $v(\neg\varphi) = v(\neg\varphi')$, therefore $\neg\varphi \equiv \neg\varphi'$ is proved.

(ii) According to the definition of truth valuation, $v(\varphi \wedge \psi) = v(\varphi) \&\& v(\psi)$, $v(\varphi' \wedge \psi') = v(\varphi') \&\& v(\psi')$, because $\varphi \equiv \varphi'$ and $\psi \equiv \psi'$, we can know $v(\varphi) = v(\varphi')$, $v(\psi) = v(\psi')$. So, there has $v(\varphi) \&\& v(\psi) = v(\varphi') \&\& v(\psi')$, which means $v(\varphi \wedge \psi) = v(\varphi' \wedge \psi')$, therefore $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$ is proved.

(iii) The method to show $\varphi \vee \psi \equiv \varphi' \vee \psi'$ is similar with (ii). $v(\varphi \vee \psi) = v(\varphi) || v(\psi)$, $v(\varphi' \vee \psi') = v(\varphi') || v(\psi')$, because $\varphi \equiv \varphi'$ and $\psi \equiv \psi'$, we can know $v(\varphi) = v(\varphi')$, $v(\psi) = v(\psi')$. So, there has $v(\varphi) || v(\psi) = v(\varphi') || v(\psi')$, which means $v(\varphi \vee \psi) = v(\varphi' \vee \psi')$, therefore $\varphi \vee \psi \equiv \varphi' \vee \psi'$ is proved.

(d) We can show that $F_{\equiv}$ satisfy commutative, associative, distributive, identity, complementation to prove $F_{\equiv}$ forms a Boolean Algebra.

(1) Commutative

Assume that $[x], [y] \in F_{\equiv}$, R is the logical equivalence relation. From the defined operations, $[\varphi] \vee [\psi]\ to\ be\ [\varphi \vee \psi], [\varphi] \wedge [\psi]\ to\ be\ [\varphi \wedge \psi]$. Hence, $[x] \vee [y]$, $[y] \vee [x]$ can be defined to be $[x \vee y]$, $[y \vee x]$ respectively, and $[x] \wedge [y], [y] \wedge [x]$ be $[x \wedge y], [y \wedge x]$ respectively. From the commutative law of Boolean Algebra, we know $x \vee y \equiv y \vee x$, $x \wedge y \equiv y \wedge x$, then $(x \vee y, y \vee x) \in R$, $(x \wedge y, y \wedge x) \in R$. In this way, $[x \vee y] = [y \vee x], [x \wedge y] = [y \wedge x]$. Since the defined operations, $[x] \vee [y] = [y] \vee [x]$, $[x] \wedge [y] = [y] \wedge [x]$.

Therefore, the commutative law holds for $F_{\equiv}$.

(2) Associative

Assume that $[x], [y], [z] \in F_{\equiv}$, R is the logical equivalence relation. From the defined operations, we can derive that $[x] \vee [y]$, $[y] \vee [z]$, $[x] \wedge [y]$, $[y] \wedge [z]$ to be $[x \vee y]$, $[y \vee z]$, $[x \wedge y]$, $[y \wedge z]$ respectively. Then, continue to derive that $([x \vee y]) \vee [z], [\mathrm{x}] \vee ([y \vee z])$, $([x \wedge y]) \wedge [z]$, $[x] \wedge ([y \wedge z])$ to be $[x \vee y \vee z]$, $[x \vee y \vee z]$, $[x \wedge y \wedge z]$, $[x \wedge y \wedge z]$ respectively. From the Associative law of Boolean Algebra, we know $(x \vee y) \vee z \equiv x \vee (y \vee z)$, $(x \wedge y) \wedge z \equiv x \wedge (y \wedge z)$, then $((x \vee y) \vee z, x \vee (y \vee z)) \in R$, $((x \wedge y) \wedge z, x \wedge (y \wedge z)) \in R$. Hence, $[(x \vee y) \vee z] = [x \vee (y \vee z)]$, $[(x \wedge y) \wedge z] = [x \wedge (y \wedge z)]$, so there has $([x] \vee [y]) \vee [z] = [x] \vee ([y] \vee [z]), ([x] \wedge [y]) \wedge [z], [x] \wedge ([y] \wedge [z])$.

Therefore, the associative law holds for $F_{\equiv}$.

(3) Distributive

We can suppose that $[x], [y], [z] \in F_{\equiv}, R$ is the logical equivalence relation Then, from the defined operations, there has $[x] \vee ([y] \wedge [z])$, $[x] \wedge ([y] \vee [z])$ to be $[x] \vee ([y \wedge z])$, $[x] \wedge ([y \vee z])$, keeping use the operations, then we can get $[x \vee (y \wedge z)]$, $[x \wedge (y \vee z)]$ respectively. According to the distributive of Boolean Algebra for x, y, z, $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$,

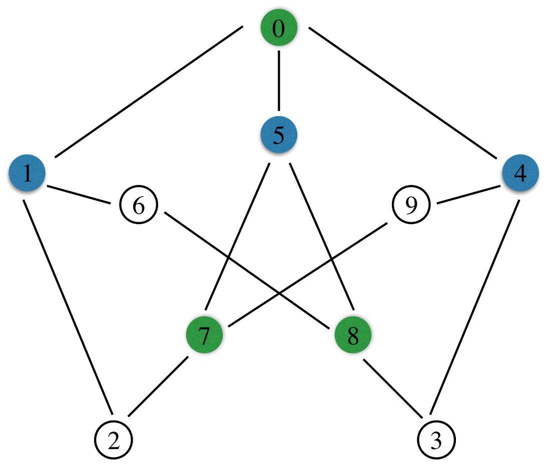
$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$. So, $(x \vee (y \wedge z), (x \vee y) \wedge (x \vee z)) \in R$, $(x \wedge (y \vee z), (x \wedge y) \vee (x \wedge z)) \in R$. Hence, $[(x \wedge y) \vee z] = [(x \vee z) \wedge (y \vee z)]$, $[(x \vee y) \wedge z] = [(x \wedge z) \wedge (y \vee z)]$, then we have $[x] \vee ([y] \wedge [z]) = [(x \vee y)] \wedge [(x \vee z)]$ and $[x] \wedge ([y] \vee [z]) = [(x \wedge y)] \vee [(x \wedge z)]$.
Therefore, the distributive law holds for $F_{\equiv}$.
(4) Identity
Suppose $[x] \in F_{\equiv}$, R is the logical equivalence relation. Because $x \vee\bot \equiv x$, $x \wedge \top \equiv x$ according to the identify of Boolean Algebra, $(x \vee\bot, x) \in R$, $(x \wedge\top, x) \in R$, there has $[x] \vee [\bot] = [x \vee\bot] = [x]$, $[x] \wedge$ $[\top] = [x \wedge\top] = [x]$. Since $[\bot], [\top]$ is defined to be 0 and 1, $[x] \vee 0 = [x], [x] \wedge 1 = [x]$
Therefore, the identity law holds for $F_{\equiv}$.

(5) Complementation
Because complementation in Boolean Algebra, $x \vee x' = 1, x \wedge x' = 0$. $[\varphi]'$ is defined to be $[\neg\varphi]$, $[\varphi] \vee[\neg\varphi] =[\varphi \vee \neg\varphi]$, $[\varphi] \wedge[\neg\varphi] = [\varphi \wedge \neg\varphi]$ and $\varphi \vee \neg\varphi \equiv\top$, $\varphi \wedge \neg\varphi \equiv\bot$, there has $(\varphi \vee \neg\varphi , \top) \in R$, $(\varphi \wedge \neg\varphi, \bot) \in R$
Then $[\varphi \vee \neg\varphi] = [\top]$, $[\varphi \wedge \neg\varphi] =\bot$, and$[\bot] \in F_{\equiv}$, $[\top] \in F_{\equiv}$. Because $[\bot]$, $[\top]$ is defined to be 0 and 1 by $[\varphi \vee \neg\varphi] = [\top]$, $[\varphi \wedge \neg\varphi] =\bot$, so $[\varphi] \vee[\varphi]' = 1, [\varphi] \wedge[\varphi]' = 0$.
Therefore, the complementation law holds.

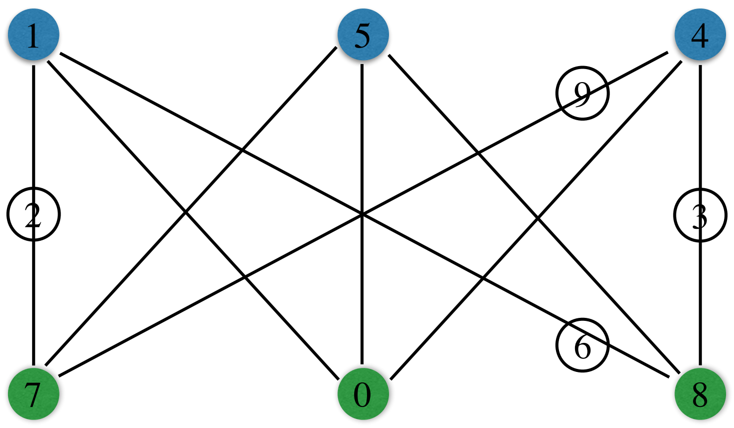
Problem 2:

(a) We cannot show that the Petersen graph contains a subdivision of K_5 cause it's impossible. Though Petersen graph has a K_5 minor, since we could require every vertex of degree 4, it does not contain a subdivision of K_5.

(b) Firstly, we can remove the edge of 2,3 and 6,9, then we can get following graph:



Secondly, we make the graph a visual standard $K_{3,3}$ one:



Above, it's obvious that the Petersen graph contains a subdivision of $K_{3,3}$.

Problem 3:

(a) Give alphas to represent the subjects as following:

A: Defense against the Dark Arts

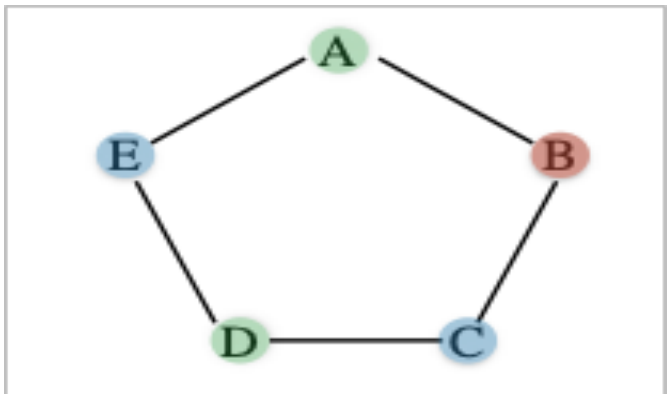
B: Potions

C: Herbology

D: Transfiguration

E: Charms

Then we can model this graph:



(i) In above graph, the vertices show the different subjects, and the edges between them show they are clashed classes.

(ii) We can regard this problem as a coloring graph problem, we get the connected vertices into different colors, which means the clashed subjects have different colors, once we color the graph in the minimum number of colors, the number of vertices in the same color is the maximum number of subjects Harry can take.

(b) From the above colored graph, we use three colors to color it, and the maximum number of vertices in the same color is 2.

Therefore, Harry can mostly take 2 classes.

Problem 4:

(a) T(0) = 1 (The binary tree is an empty tree)

T(1) = 1

T(2) = 2

T(3) = T(0)* T(2) + T(1)* T(1) + T(2)* T(0) = 1*2 + 1*1 +2*1 = 5

Suppose $T(i-1)$ represents number of nodes on the left-sub-tree, $T(n-i-1)$ represents number of nodes on the right-sub-tree.

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

Therefore, according to the recursive definition:

```
$T(n)$:
[B] if n = 0:    (where n = 0 means the binary tree is an empty tree)
        return 0
[R] else:
        return   $\sum_{i=0}^{n-1} T(i)T(n-i-1)$
```

(b) We can use the inductive reasoning to explain.

When $n = 1$, a binary tree only has a root, it's obvious that the full binary tree has odd number 1 of node.

When $n \geq 1$, we can suppose all full binary tree with k nodes has $1 \leq k < n$, where k is odd. In this situation, we're supposed to prove that all results hold for a full binary tree T. For a full binary tree, n must be at least 3, because there has no full binary tree with 2 nodes. Besides, for all full binary tree with 3 or more nodes, they must have a root with non-empty left and non-empty right subtrees. So, T must have 3 or more nodes, non-empty left subtrees T_L and non-empty right subtrees T_R, each of them has $< n$ nodes. Because T is a full binary tree, each node of T_L and T_R has zero or two children. By the induction hypothesis each has an odd number of nodes, let $2k+1$ and $2m+1$ $(k, m \geq 0)$ be the number of nodes of T_L and T_R respectively, then the sum nodes of T is $2(k+m)+3$, which is an odd number.

Therefore, it's proved that a full binary tree must have an odd number of nodes.

(c) $B(n)$:

```
[B] if n = 1:        (where n = 1 means the binary tree has only 1 root)
        return 1
[R] else:
        return $T(\frac{n-1}{2})$
        (where $T(\frac{n-1}{2}) = T(\frac{n-1}{2})T(0)+ T\left(\frac{n-5}{2}\right)T(1)+......+T(0)\,T\left(\frac{n-3}{2}\right)$
        for $n = 3,5,7,9,11......$)
```

(d) The structure of well-formed formula in Negated normal form can be shown as a full binary tree. According to the expression in (c), we can show $F(n)$ as:

$F(n)$=$[T(n-2)T(0) + T(n-3)T(1) +......+T(0)T(n-2)]$*$2^{2n-1} * n!$

Problem 5:

(a) Since $p_1(0) = 1$, $p_2(0) = 0$, $p_3(0) = 0$, $p_4(0) = 0$, initially start at v_1, and v_1, v_3 have two branches, v_2, v_4 have three branches, we can find out that:

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

(b) We set $p_i(n+1) = p_i(n)$, and we've already know the relationship from question (a), so there has:

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

the sum probability of these four is 1, we can show the probability as:

$p_1(n) + p_2(n) + p_3(n) + p_4(n) = 1$,

So, steady state probability is $p_1(n) = p_3(n) = \frac{1}{5}$, $p_2(n) = p_4(n) = \frac{3}{10}$.

(c) We can suppose that the distance between v_1 and v_2 is v_1v_2, so does v_1v_3 and v_1v_4, because the distance between v_1 and v_1 is 0, we don't need to consider about it. According to the result we get from (b), the expected distance is $p_2(n)*v_1v_2 + p_3(n)*v_1v_3 + p_4(n)*v_1v_4$, which means the result is $\frac{3}{10}v_1v_2 + \frac{1}{5}v_1v_3 + \frac{3}{10}v_1v_4$.