Solutions to Midterm 1

1. (a) A and $\neg A$ cannot simultaneously be true, so the proposition is false for all models. Hence, not valid; not satisfiable; unsatisfiable.

   (b) $A = B = C = F$ makes this proposition true. $A = T, B = C = F$ makes this proposition false. So, it is true for some models and false for some models. Hence, not valid; satisfiable; not unsatisfiable.

   (c) Rewrite the proposition as $(B \vee \neg A) \vee (A \vee \neg B)$ to see it is true for all models. Hence, valid; satisfiable; not unsatisfiable.

   Remark: These problems can also be answered using full truth tables, but as shown above, this can be avoided in each case.

2. (a) `A` is at depth 1, `B` and `C` are at depth 2.

   (b) $d+1$. (Informally: The root of t_1 in the tree $t_1 \bullet t_2$ is at depth 1. The number of hops from l to the root of $t_1 \bullet t_2$ is the number of hops from l to the root of t_1—that is, d—plus the number of hops from t_1's root to the root of $t_1 \bullet t_2$—which is 1.)

   (c) We have:

$$\begin{aligned} L(t) &= \frac{1}{2^{depth(\mathtt{A},t)}} + \frac{1}{2^{depth(\mathtt{B},t)}} + \frac{1}{2^{depth(\mathtt{C},t)}} \\ &= \frac{1}{2} + \frac{1}{4} + \frac{1}{4} \\ &= 1 \end{aligned}$$

   (d) Proof by induction. Base case ($t = \mathtt{A}$ is an atom): $L(t) = 1/2^0 = 1$. Inductive case: Assume $L(t_1) = L(t_2) = 1$. We need to show $L(t_1 \bullet t_2) = 1$.

$$\begin{aligned} L(t_1 \bullet t_2) &= \sum_{l: l \text{ is a leaf of } t_1 \bullet t_2} \frac{1}{2^{depth(l, t_1 \bullet t_2)}} \\ &= \sum_{l: l \text{ is a leaf of } t_1} \frac{1}{2^{depth(l, t_1 \bullet t_2)}} + \sum_{l: l \text{ is a leaf of } t_2} \frac{1}{2^{depth(l, t_1 \bullet t_2)}} \\ &= \sum_{l: l \text{ is a leaf of } t_1} \frac{1}{2^{depth(l, t_1)+1}} + \sum_{l: l \text{ is a leaf of } t_2} \frac{1}{2^{depth(l, t_2)+1}} \\ &= \frac{1}{2} \sum_{l: l \text{ is a leaf of } t_1} \frac{1}{2^{depth(l, t_1)}} + \frac{1}{2} \sum_{l: l \text{ is a leaf of } t_2} \frac{1}{2^{depth(l, t_2)}} \\ &= \frac{1}{2} L(t_1) + \frac{1}{2} L(t_2) \\ &= 1 \end{aligned}$$

3. (a) Suppose that there is some pair i, j, such that $L_i \equiv \neg L_j$. Then $C \equiv (L_1 \vee \ldots \vee L_i \vee \ldots \vee L_j \vee \ldots \vee L_k) \equiv (L_1 \vee \ldots \vee L_i \vee \ldots \vee \neg L_i \vee \ldots \vee L_k)$ would be true under any model since $L_i \vee \neg L_i$ is valid. Now suppose that there is *no* pair i, j such that $L_i \equiv \neg L_j$ (i.e. there are not literals of the forms A and $\neg A$). Then we can always find an assignment that makes every literal false: Simply observe that each variable in C appears only in negated form or in non-negated form, and so consider the model in which each variable appearing in negated form in C is true, and each variable appearing in non-negated form in C is false. Under this model, C is false. So, C is not valid.

(b) The algorithm is as follows:

- Examine each clause in the CNF expression to determine if there is a variable that appears in both negated and in non-negated form. (i.e. if there are literals of the form $\neg A$ and A in the clause.)
- If such variables appear in all clauses, conclude the CNF expression is valid. Otherwise, conclude it is not valid.

To show the algorithm is correct, note that $C_1 \wedge \cdots \wedge C_n$ is valid if and only if $C_1, C_2, \ldots, C_n$ are all valid. That is, if $C_1, \ldots, C_n$ are all true for all models, then so is $C_1 \wedge C_2 \wedge \ldots \wedge C_n$; if, on the other hand, $C_1, \ldots, C_n$ are not all true for all models, then there is some model under which some C_i is false, in which case the same model makes $C_1 \wedge \cdots \wedge C_n$ false. Using our result from part (a), we therefore know that a CNF expression is valid if and only if, in each of its clauses, there is a variable that appears in both negated and non-negated form.

(c) We have:

$$\begin{aligned}((A \vee B) \implies C) \implies (A \implies C) &\equiv \neg(\neg(A \vee B) \vee C) \vee (\neg A \vee C) \\ &\equiv ((A \vee B) \wedge \neg C) \vee \neg A \vee C \\ &\equiv (A \vee B \vee \neg A \vee C) \wedge (\neg C \vee \neg A \vee C)\end{aligned}$$

The first clause has A and $\neg A$; the second clause has $\neg C$ and C. Hence, the proposition is valid.

Remark: A common error in part (a) was to prove only one direction of the implication. A correct solution requires a proof of both the "if" and the "only if" parts!

4. (a) Call the two colors 0 and 1. For each $i = 1, \ldots, n$, let X_i be true iff country j is colored with color 1. Then for a 2-coloring to be feasible, any two adjacent countries i and j must have different colors, so one of X_i and X_j is true and one is false. Hence, the following proposition is satisfiable if and only if there is a feasible 2-coloring;

$$P \equiv \bigwedge_{(i,j):C_i,\ C_j \text{ adjacent}} (X_i \wedge \neg X_j) \vee (\neg X_i \wedge X_j)$$

To convert it to CNF, we use the distributivity to get

$$\begin{aligned}P &\equiv \bigwedge_{(i,j):C_i,\ C_j \text{ adjacent}} (X_i \vee \neg X_i) \wedge (X_i \vee X_j) \wedge (\neg X_j \vee \neg X_i)\,(\neg X_j \vee X_j) \\ &\equiv \bigwedge_{(i,j):C_i,\ C_j \text{ adjacent}} (X_i \vee X_j) \wedge (\neg X_j \vee \neg X_i)\end{aligned}$$

(b) We may, without loss of generality, assume country C_1 is colored with color 1, so $X_1 = T$. We wish to see if C_2 must then also be always colored with color 1. In other words, we wish to test if $X_1 \wedge P \implies X_2$ is valid, or, equivalently, if $\neg(X_1 \wedge P \implies X_2)$ is unsatisfiable. We have:

$$\begin{aligned} \neg(X_1 \wedge P \implies X_2) &\equiv \neg(\neg(X_1 \wedge P) \vee X_2) \\ &\equiv X_1 \wedge P \wedge \neg X_2 \end{aligned}$$

which is in CNF since P is in CNF. This CNF expression is unsatisfiable if and only if C_1 and C_2 must have the same color in any feasible 2-coloring.

Remark: For part (a), many people tried to construct variables X_{ij} which were true iff countries i and j are adjacent. But these are not *variables*! For any given map, the adjacency relations are fixed; they determine the structure of the logical constraints on the colors.

5. (a) In the inductive step, the proof for $P(n+1)$ appeals to $P(n)$ and $P(n-1)$, which fails for $n+1=2$ because $P(n-1)$ is $P(0)$ which is unproved (and false).

(b) The inductive step of this proof starts by assuming $P(n+1)$ and then proceeds to show $P(n)$, rather than the reverse. Note that writing all the lines of the inductive part of the proof in reverse order also would not make this a correct proof: The step going from the second-to-last inequality to the last inequality of the proof uses a one-way implication. (i.e. $(A \geq B) \wedge (C \geq D) \implies A+C \geq B+D$, but $A+C \geq B+D \not\Rightarrow (A \geq B) \wedge (C \geq D)$).

Remark: For part (a), many people essentially pointed out that $5n-5$ is not equal to 0 for all n, or found nonexistent errors in the first line of the inductive step. The inductive step is perfectly correct given P(n) and P(n-1).

For part (b), many people failed to notice that the proof was in the wrong direction, but focussed instead on the final step, claiming it to be an incorrect deduction. But consider this: If $x + 0.8 > y + 1$ then $x > y$! (If you're not convinced, consider the intermediate step $x > y + (1 - 0.8) = y + 0.2$).