

Problem 1

(40 marks)

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, \equiv , is an equivalence relation on F . (12 marks)
- (b) List four elements in $[\perp]$, the equivalence class of \perp . (4 marks)
- (c) For all $\varphi, \varphi', \psi, \psi' \in F$ with $\varphi \equiv \varphi'$ and $\psi \equiv \psi'$; show that:
- (i) $\neg\varphi \equiv \neg\varphi'$ (4 marks)
 - (ii) $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$ (4 marks)
 - (iii) $\varphi \vee \psi \equiv \varphi' \vee \psi'$ (4 marks)

Let us define F_{\equiv} to be the set of **equivalence classes** of F under \equiv . That is,

$$F_{\equiv} := \{[\varphi] : \varphi \in F\}.$$

Part (c) above shows that the following operations are *well-defined*¹ on F_{\equiv} :

- $[\varphi] \wedge [\psi]$ defined to be $[\varphi \wedge \psi]$
- $[\varphi] \vee [\psi]$ defined to be $[\varphi \vee \psi]$
- $[\varphi]'$ defined to be $[\neg\varphi]$

- (d) Show that F_{\equiv} together with the operations defined above forms a Boolean Algebra. Note: you will have to give a suitable definition of a zero element and a one element in F_{\equiv} . (12 marks)

¹well-defined means that the output is not dependent on different representations of the same input.

Solution

(a) We will show that \equiv satisfies Reflexivity, Symmetry, and Transitivity.

- For any formula φ and any valuation v , we have $v(\varphi) = v(\varphi)$, so $\varphi \equiv \varphi$, and \equiv is reflexive.
- Suppose $\varphi \equiv \psi$. Then for any valuation v : $v(\varphi) = v(\psi)$. But then $v(\psi) = v(\varphi)$, so $\psi \equiv \varphi$ and \equiv is symmetric.
- Suppose $\varphi \equiv \psi$ and $\psi \equiv \theta$. Then, for any valuation v : $v(\varphi) = v(\psi)$ and $v(\psi) = v(\theta)$. But then $v(\varphi) = v(\theta)$, so $\varphi \equiv \theta$; and \equiv is transitive.

(b) $[\perp] = \{\perp, \perp \vee \perp, \perp \wedge \perp, \top \rightarrow \perp, p \wedge \neg p, p \wedge \perp, \dots\}$

(c) For all valuations v , we have $v(\varphi) = v(\varphi')$ and $v(\psi) = v(\psi')$. Therefore, for all valuations v :

- (i) $v(\neg\varphi) = \neg v(\varphi) = \neg v(\varphi') = v(\neg\varphi')$. So $\neg\varphi \equiv \neg\varphi'$.
- (ii) $v(\varphi \wedge \psi) = v(\varphi) \wedge v(\psi) = v(\varphi') \wedge v(\psi') = v(\varphi' \wedge \psi')$. So $\varphi \wedge \psi \equiv \varphi' \wedge \psi'$.
- (iii) For an approach that makes use of (i) and (ii):

$$\begin{aligned}
 \varphi \vee \psi &\equiv \neg\neg\varphi \vee \neg\neg\psi && \text{(Double negation)} \\
 &\equiv \neg(\neg\varphi \wedge \neg\psi) && \text{(De Morgan's)} \\
 &\equiv \neg(\neg\varphi' \wedge \neg\psi') && \text{From (i) and (ii)} \\
 &\equiv \neg\neg\varphi' \vee \neg\neg\psi' && \text{(De Morgan's)} \\
 &\equiv \varphi' \vee \psi' && \text{(Double negation)}
 \end{aligned}$$

Discussion

Each property of an equivalence relation is out of 4 marks. 1 mark per correctly identified element of $[\perp]$. At least two of (c) must be shown through first principles. For the proofs ((a) and (c)):

- Minor errors include small omissions in logical reasoning; but otherwise correct.
- Major errors include proving the result for specific examples, not in full generality.
- Shows progress includes identifying reflexivity, symmetry and transitivity.

Solution (ctd)

- (d) Suppose we define zero to be $[\perp] \in F_{\equiv}$ and one to be $[\top] \in F_{\equiv}$. We claim that F_{\equiv} with the given operations and constants defines a Boolean Algebra. From the lectures, we identified several relevant properties identifying logical equivalent propositions, namely:

Commutativity:	$\varphi \wedge \psi \equiv \psi \wedge \varphi$	$\varphi \vee \psi \equiv \psi \vee \varphi$
Associativity:	$\varphi \wedge (\psi \wedge \theta) \equiv (\varphi \wedge \psi) \wedge \theta$	$\varphi \vee (\psi \vee \theta) \equiv (\varphi \vee \psi) \vee \theta$
Distributivity:	$\varphi \wedge (\psi \vee \theta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \theta)$	$\varphi \vee (\psi \wedge \theta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \theta)$
Identity:	$\varphi \wedge \top \equiv \varphi$	$\varphi \vee \perp \equiv \varphi$
Complement:	$\varphi \wedge \neg \varphi \equiv \perp$	$\varphi \vee \neg \varphi \equiv \top$

And these properties hold for all $\varphi, \psi, \theta \in F$.

From these results, and using the fact that if $\varphi \equiv \psi$ then $[\varphi] = [\psi]$, showing that F_{\equiv} is a Boolean algebra is straightforward. For all $[\varphi], [\psi], [\theta] \in F_{\equiv}$:

Commutativity:	$[\varphi] \wedge [\psi] = [\varphi \wedge \psi] = [\psi \wedge \varphi] = [\psi] \wedge [\varphi]$ $[\varphi] \vee [\psi] = [\varphi \vee \psi] = [\psi \vee \varphi] = [\psi] \vee [\varphi]$	
Associativity:	$ \begin{aligned} [\varphi] \wedge ([\psi] \wedge [\theta]) &= [\varphi] \wedge [\psi \wedge \theta] & [\varphi] \vee ([\psi] \vee [\theta]) &= [\varphi] \vee [\psi \vee \theta] \\ &= [\varphi \wedge (\psi \wedge \theta)] & &= [\varphi \vee (\psi \vee \theta)] \\ &= [\varphi \wedge (\psi \wedge \theta)] & &= [\varphi \vee (\psi \vee \theta)] \\ &= [(\varphi \wedge \psi) \wedge \theta] & &= [(\varphi \vee \psi) \vee \theta] \\ &= [\varphi \wedge \psi] \wedge [\theta] & &= [\varphi \vee \psi] \vee [\theta] \\ &= ([\varphi] \wedge [\psi]) \wedge [\theta] & &= ([\varphi] \vee [\psi]) \vee [\theta] \end{aligned} $	
Distributivity :	$ \begin{aligned} [\varphi] \wedge ([\psi] \vee [\theta]) &= [\varphi] \wedge [\psi \vee \theta] & [\varphi] \vee ([\psi] \wedge [\theta]) &= [\varphi] \vee [\psi \wedge \theta] \\ &= [\varphi \wedge (\psi \vee \theta)] & &= [\varphi \vee (\psi \wedge \theta)] \\ &= [(\varphi \wedge \psi) \vee (\varphi \wedge \theta)] & &= [(\varphi \vee \psi) \wedge (\varphi \vee \theta)] \\ &= [\varphi \wedge \psi] \vee [\varphi \wedge \theta] & &= [\varphi \vee \psi] \wedge [\varphi \vee \theta] \\ &= ([\varphi] \wedge [\psi]) \vee ([\varphi] \wedge [\theta]) & &= ([\varphi] \vee [\psi]) \wedge ([\varphi] \vee [\theta]) \end{aligned} $	
Identity:	$[\varphi] \wedge [\top] = [\varphi \wedge \top] = [\varphi]$ $[\varphi] \vee [\perp] = [\varphi \vee \perp] = [\varphi]$	
Complement:	$[\varphi] \wedge [\varphi]' = [\varphi] \wedge [\neg \varphi] = [\varphi \wedge \neg \varphi] = [\perp]$ $[\varphi] \vee [\varphi]' = [\varphi] \vee [\neg \varphi] = [\varphi \vee \neg \varphi] = [\top]$	

Discussion

For full marks students should:

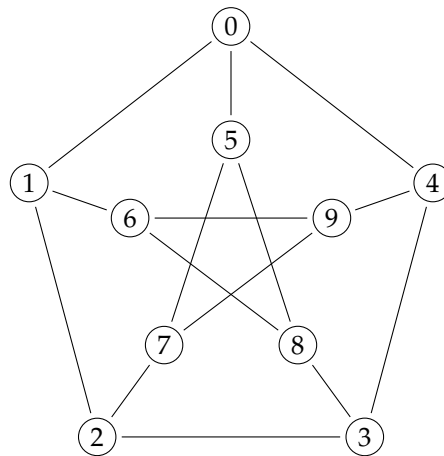
- Identify a suitable zero and one
- Establish all the properties of a Boolean Algebra
- Have a clear argument for correctness

Minor errors include small logical omissions. Major errors include omitting one of the above points. Shows progress includes having one of the above points.

Problem 2

(10 marks)

This is the Petersen graph:

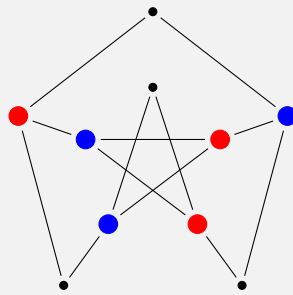


- (a) Give an argument to show that the Petersen graph does not contain a subdivision of K_5 . (5 marks)
- (b) Show that the Petersen graph contains a subdivision of $K_{3,3}$. (5 marks)

Solution

To show that a graph G contains a subdivision of H we can do the following.

- Starting with G .
 - Repeat any of the following operations as many times as necessary:
 - Delete edges
 - Delete vertices
 - Contract a vertex of degree 2 (delete the vertex and connect its two neighbours with an edge)
 - Obtain H .
- (a) We observe that in each of the above operations we will never increase the degree of a vertex. Each of the vertices in the Petersen graph has degree 3, but each vertex in K_5 has degree 4. Therefore the Petersen graph does not contain a subdivision of K_5 .
- (b) Here is a subdivision of $K_{3,3}$ which is clearly a subgraph of the Petersen graph. The two partitions are indicated by red and blue vertices. The black vertices subdivide edges.



Discussion

- Full marks for clear logical arguments
- Minor errors (4 marks) for small omissions such as not indicating the approach for finding a subdivision (two different approaches were given in the lecture, the students should indicate which one they are using).
- Major errors (2 marks) for a clear, but incorrect proof
- Shows progress (1 mark) for demonstrating some understanding

Problem 3

(10 marks)

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. (4 marks)
- (ii) State the associated graph problem that you need to solve. (2 marks)

(b) Give the solution to the graph problem corresponding to this scenario; and solve Harry's problem. (4 marks)

Solution

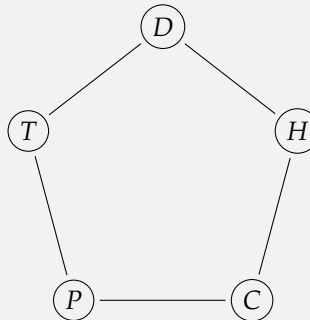
(a) We model this problem as the following graph problem.

(i) We define an (undirected) graph G as follows:

- Vertices: Set of subjects Harry wants to take. In this case: $\{D, P, H, T, C\}$
- Edges: An edge between two subjects if they **do not** have a clash. In this case: $\{\{D, T\}, \{D, H\}, \{P, T\}, \{H, C\}, \{T, D\}\}$

(ii) A clique in this graph represents a set of mutually-compatible subjects, so we want to find the largest clique that appears in this graph. That is, we are interested in the clique-number of the graph, $\kappa(G)$.

(b) In this case we want the clique number of this graph:



This graph does not contain K_3 , so the clique number is at most 2. It does contain a clique of size 2 (e.g. $\{D, T\}$), so its clique number is 2. Therefore the most number of subjects that Harry can take without conflict is 2.

Discussion

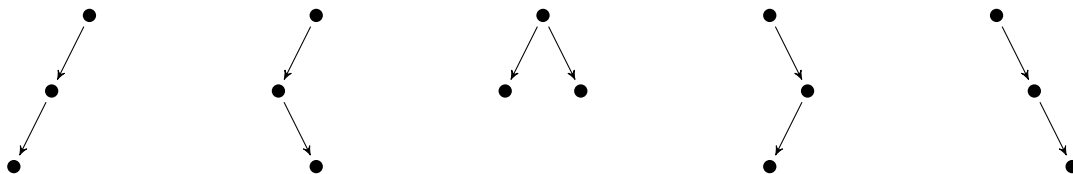
- For (a) (i): Any clearly defined graph will do. A definition different to the sample solution could be correct with the right question in (ii).
 - Full marks for defining the graph without reference to the specific case (e.g. “The subjects Harry wants to take”)
 - 2 marks for defining a specific graph based on the question
 - 1 mark for drawing a graph without indicating where/how the elements were obtained.
- For (a) (ii): Full marks for a correct and clearly defined question. 1 mark for a correct but unclear question. 0 marks for a question that does not correctly give the answer.
- For (b)
 - +1 mark for giving the correct graph based on the question and the definitions in (a)
 - +2 marks for correctly identifying the correct parameter (+1 for upper bound, +1 for lower bound)
 - +1 mark for correctly translating it back to the original problem.

Problem 4

(22 marks)

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 $T(n)$. (8 marks)

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. (4 marks)
- (c) Let $B(n)$ denote the number of full binary trees with n nodes. Derive an expression for $B(n)$, involving $T(n')$ where $n' \leq n$. Hint: Relate the internal nodes of a full binary tree to $T(n)$. (6 marks)

A well-formed formula is in **Negated normal form** if it consists of just \wedge , \vee , and literals (i.e. propositional variables or negations of propositional variables). That is, a formula that results after two steps of the process for transforming a formula into a logically equivalent one. For example, $p \vee (\neg q \wedge \neg r)$ is in negated normal form; but $p \vee \neg(q \vee r)$ is not.

Let $F(n)$ denote the number of well-formed, negated normal form formulas² there are that use precisely n propositional variables exactly one time each. So $F(1) = 2$, $F(2) = 16$, and $F(4) = 15360$.

(d) Using your answer for part (c), give an expression for $F(n)$. (4 marks)

Solution

(a) A binary tree is either empty (has no nodes) or contains a single node with two trees as children. Therefore there is only one tree that has 0 nodes, so $T(0) = 1$.

For trees with $n \geq 1$ nodes, we must be in the recursive case, so we have 1 node at the root, $m \in [0, n-1]$ nodes, say, in the left subtree, and $n-m-1$ nodes in the right subtree. For each value of m , we know that there are $T(m)$ different trees that can be the left subtree, and $T(n-m-1)$ trees that can be the right subtree, giving $T(m) \cdot T(n-m-1)$ possible trees when there are m nodes in the left subtree. As different values of m give different trees, the total number of trees is obtained by summing this value over all m . That is,

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

(b) Let l be the number of leaves, i be the number of fully-internal nodes, and n be the number of nodes. From Assignment 2: $l = i + 1$. In a full binary tree, we have $n = l + i = 2i + 1$. As i is an integer, n is odd.

(c) There is a 1-1 correspondence between an arbitrary binary tree, and the fully-internal nodes of a full binary tree given as follows:

- Given a full binary tree with i fully-internal nodes, delete all the leaves. The result is a binary tree with i nodes.
- Given a binary tree with n nodes, replace all “empty” subtrees with a tree with one node. Each newly added node is now a leaf, and any original node is now a fully-internal node; so the result is a full binary tree with n fully-internal nodes.

It is easy to see that the above operations are inverses of one another, so the connection is a bijection. From (b) we have that a full binary tree with n nodes has $\frac{n-1}{2}$ fully-internal nodes if n is odd, and if n is even, there are no full binary trees with n nodes. This gives us the following formula for $B(n)$:

$$B(n) = 0 \text{ if } n \text{ is even} \quad B(n) = T\left(\frac{n-1}{2}\right) \text{ if } n \text{ is odd.}$$

²Note: we do not assume \wedge and \vee are associative

Solution (ctd)

(d) Consider the parse tree of a negated normal form formula where n propositional variables are used exactly one time each. It will be a full binary tree with n leaves where the fully-internal nodes are either \wedge or \vee and the leaves are literals (i.e. propositional variables or their negations). As we are not assuming associativity or commutativity, different choices for each of the nodes gives different formulas, so we just need to count the number of ways there are of assigning symbols to the nodes of a full binary tree with n leaves.

- From (b), a full binary tree with n leaves has $2n - 1$ nodes, so there are $B(2n - 1) = T(n - 1)$ such trees.
- From Assignment 2, a full binary tree with n leaves has $n - 1$ fully-internal nodes. Each of these can be either \wedge or \vee , giving 2^{n-1} assignments of symbols to the fully-internal nodes
- Each leaf is a literal, so can either be negated or not, giving 2^n possible assignments of \neg .
- The order of propositional variables is significant, and there are $n!$ of arranging the n propositional variables to assign them to the leaves.

Putting everything together, we obtain the following formula for $F(n)$:

$$F(n) = 2^{n-1} \cdot 2^n \cdot n! \cdot T(n - 1) = 2^{2n-1} \cdot n! \cdot T(n - 1)$$

Discussion

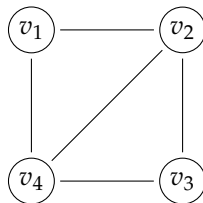
The $T(n)$ are known as the Catalan numbers. As this question demonstrates they are very useful for counting various tree-like structures.

- Minor errors include: Omitting the base case in (a), even case in (c), one bullet point in (d)
- Major errors include: Correct answer in (a) or (c) or (d) without justification or by appealing to external resources; omitting two or more of the bullet points in (d)
- Shows progress includes: A recurrence formula in (a), (c), or (d) that is incorrect

Problem 5

(18 marks)

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 $p_4(n)$. (6 marks)
- (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. (8 marks)
- (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? (4 marks)

Solution

- (a) To be at v_1 at time $n+1$ we must have been in either v_2 or v_4 at time n and selected v_1 as the next location (probability $\frac{1}{3}$ at each vertex). So $p_1(n+1) = \frac{1}{3}p_2(n) + \frac{1}{3}p_4(n)$. The other probabilities can be derived similarly giving:

$$\begin{aligned} p_1(n+1) &= \frac{1}{3}p_2(n) + \frac{1}{3}p_4(n) \\ p_2(n+1) &= \frac{1}{2}p_1(n) + \frac{1}{2}p_3(n) + \frac{1}{3}p_4(n) \\ p_3(n+1) &= \frac{1}{3}p_2(n) + \frac{1}{3}p_4(n) \\ p_4(n+1) &= \frac{1}{2}p_1(n) + \frac{1}{3}p_2(n) + \frac{1}{2}p_3(n). \end{aligned}$$

Note, we also have the following constraint, holding for all n , indicating that we must be somewhere on the graph:

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

If we let $\mathbf{p}(n) = (p_1(n), p_2(n), p_3(n), p_4(n))^T$ be the column vector of probabilities the above set of equations can be represented succinctly with a matrix equation:

$$\mathbf{p}(n+1) = A \cdot \mathbf{p}(n)$$

where

$$A = \begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{2} & 0 \end{pmatrix}.$$

Note that A can be derived from the adjacency matrix for the graph by normalizing each column (i.e. dividing each column by its sum so that each column sums to 1).

Solution (ctd)

(b) In the steady state we have $p_i(n+1) = p_i(n)$ for all i , so let p_i be this probability; that is, p_i is the probability that we are in vertex v_i in the steady state. From above we have the following system of equations:

$$\begin{aligned}p_1 &= \frac{1}{3}p_2 + \frac{1}{3}p_4 \\p_2 &= \frac{1}{2}p_1 + \frac{1}{2}p_3 + \frac{1}{3}p_4 \\p_3 &= \frac{1}{3}p_2 + \frac{1}{3}p_4 \\p_4 &= \frac{1}{2}p_1 + \frac{1}{3}p_2 + \frac{1}{2}p_3 \\1 &= p_1 + p_2 + p_3 + p_4.\end{aligned}$$

Solving for p_i , using e.g. <https://www.wolframalpha.com>, gives

$$p_1 = p_3 = \frac{1}{5} \quad p_2 = p_4 = \frac{3}{10}.$$

(c) If D is the random variable indicating the distance from v_1 in the steady state, then, since v_1 is distance 0 from v_1 , v_2 and v_4 are distance 1, and v_3 is distance 2:

$$\begin{aligned}E(D) &= 0 \cdot p_1 + 1 \cdot p_2 + 2 \cdot p_3 + 1 \cdot p_4 \\&= 0 \cdot \frac{1}{5} + 1 \cdot \frac{3}{10} + 2 \cdot \frac{1}{5} + 1 \cdot \frac{3}{10} \\&= 1.\end{aligned}$$

Discussion

This is an example of a Markov chain – a very useful model for stochastic processes.

- Minor errors include: One or two arithmetical errors; Not clearly defining the variables
- Major errors include: Three+ arithmetical errors; Correct answer without justification
- Shows progress includes: Any indication of understanding.