
Problem 1 (12 marks)

Consider the following two algorithms that naïvely compute the sum and product of two $n \times n$ matrices.

```
sum($A$,$B$):
  for $i \in [0, n)$:
    for $j \in [0, n)$:
      $C[i,j] = A[i,j] + B[i,j]$
    end for
  end for
  return $C$
```

```
product($A$,$B$):
  for $i \in [0, n)$:
    for $j \in [0, n)$:
      $C[i,j] = \mathsf{add}\{A[i,k] * B[k,j] : k \in [0, n)\}$
    end for
  end for
  return $C$
```

Assuming that adding and multiplying matrix elements can be carried out in $O(1)$ time, and add will add the elements of a set S in $O(|S|)$ time:

(a) Give an asymptotic upper bound, in terms of n, for the running time of sum. (3 marks)

(b) Give an asymptotic upper bound, in terms of n, for the running time of product. (3 marks)

When n is even, we can define a recursive procedure for multiplying two $n \times n$ matrices as follows. First, break the matrices into smaller submatrices:

$$A = \begin{pmatrix} S & T \\ U & V \end{pmatrix} \qquad B = \begin{pmatrix} W & X \\ Y & Z \end{pmatrix}$$

where S, T, U, V, W, X, Y, Z are $\frac{n}{2} \times \frac{n}{2}$ matrices. Then it is possible to show:

$$AB = \begin{pmatrix} SW + TY & SX + TZ \\ UW + VY & UX + VZ \end{pmatrix}$$

where $SW + TY$, $SX + TZ$, etc. are sums of products of the smaller matrices. If n is a power of 2, each smaller product (SW, TY, etc) can be computed recursively, until the product of 1×1 matrices needs to be computed – which is nothing more than a simple multiplication, taking $O(1)$ time.

Assume n is a power of 2, and let $T(n)$ be the worst-case running time for computing the product of two $n \times n$ matrices using this method.

(c) With justification, give a recurrence equation for $T(n)$. (4 marks)

(d) Find an asymptotic upper bound for $T(n)$. (2 marks)

Solution

(a) Counting elementary operations we have:

```
sum($A$,$B$) :
    for $i \in [0,n)$ :                                              $O(1)$
        for $j \in [0,n)$ :                         $O(1)$
            $C[i,j] = A[i,j] + B[i,j]$              $O(1)$  } $n$ times   $O(n)$  } $n$ times   $O(n^2)$
        end for
    end for
  return $C$                                                                                   $O(1)$
```

So the running time for sum is $O(n^2) + O(1) = O(n^2)$.

(b) Counting elementary operations we have:

```
product($A$,$B$) :
    for $i \in [0,n)$:                                                          $O(1)$
        for $j \in [0,n)$ :                                    $O(1)$
          $C[i,j] = $ add$\{A[i,k] * B[k,j] : k \in [0,n)\}$   $O(n)$  } $n$ times   $O(n^2)$  } $n$ times   $O(n^3)$
        end for
    end for
  return $C$                                                                                              $O(1)$
```

So the running time for product is $O(n^3) + O(1) = O(n^3)$

(c) (i) For the base case we have $T(1) \in O(1)$. For the recursive case, we have 8 products of $\frac{n}{2} \times \frac{n}{2}$ matrices, taking $T(\frac{n}{2})$ time each, together with 4 sums of $\frac{n}{2} \times \frac{n}{2}$ matrices. Using sum from the previous question, each of these will take $O(\frac{n^2}{4}) = O(n^2)$ time. This yields the following recurrence:

$$T(1) = O(1) \quad \text{and} \quad T(n) = 8T(\frac{n}{2}) + O(n^2).$$

(ii) Applying the Master Theorem, we have $a = 8$, $b = 2$, $c = 2$, $d = \log_b(a) = 3$; so we are in Case I and we have $T(n) \in O(n^3)$. So asymptotically there is no improvement over the naïve approach.

Remark

The Strassen algorithm for matrix multiplication is built upon this recursive procedure, but improves on the running time by computing the smaller submatrices with 7 rather than 8 recursive calls.

Discussion

- Minor errors include: Correct with limited justification (i.e 1 or 2 justification steps omitted)
- Major errors include: Correct with no justification; incorrect answers with good justification
- Shows progress includes: Incorrect answer with limited justification

---

Problem 2 (18 marks)

Recall from Assignment 2 the neighbourhood of eight houses:

As before, each house wants to set up its own wi-fi network, but the wireless networks of neighbouring houses – that is, houses that are either next to each other (ignoring trees) or over the road from one another (directly opposite) – can interfere, and must therefore be on different channels. Houses that are sufficiently far away may use the same wi-fi channel. Again we would like to solve the problem of finding the minimum number of channels needed, but this time we will solve it using techniques from logic and from probability. Rather than directly asking for the minimum number of channels required, we ask if it is possible to solve it with just 2 channels. So suppose each wi-fi network can either be on channel hi or on channel lo. Is it possible to assign channels to networks so that there is no interference?

(a) Formulate this problem as a problem in propositional logic. Specifically:

- (i) Define your propositional variables (4 marks)
- (ii) Define any propositional formulas that are appropriate and indicate what propositions they represent. (4 marks)
- (iii) Indicate how you would solve the problem (or show that it cannot be done) using propositional logic. It is sufficient to explain the method, you do not need to provide a solution. (2 marks)
- (iv)* Explain how to modify your answer(s) to (i) and (ii) if the goal was to see if it is possible to solve with 3 channels rather than 2. (4 marks)

(b) Suppose each house chooses, uniformly at random, one of the two network channels. What is the probability that there will be no interference? (4 marks)

Problem 3 (12 marks)

Prove the following results hold in all Boolean Algebras:

(a) For all x: $(x \wedge 1') \vee (x' \wedge 1) = x'$ (4 marks)

(b) For all x, y: $(x \wedge y) \vee x = x$ (4 marks)

(c) For all x, y: $y' \wedge ((x \vee y) \wedge x') = 0$ (4 marks)

Proof assistant

https://cgi.cse.unsw.edu.au/∼cs9020/cgi-bin/logic/22T2/boolean_algebra/assignment3a

Solution

(a)

$$\begin{aligned}
(x \wedge 1') \vee (x' \wedge 1) &= (x \wedge 0) \vee (x' \wedge 1) && \text{(Complement of 1)} \\
&= 0 \vee (x' \wedge 1) && \text{(Annihilation of } \wedge\text{)} \\
&= 0 \vee x' && \text{(Identity of } \wedge\text{)} \\
&= x' \vee 0 && \text{(Commutatitivity of } \vee\text{)} \\
&= x' && \text{(Identity of } \vee\text{)}
\end{aligned}$$

(b)

$$\begin{aligned}
(x \wedge y) \vee x &= (x \wedge y) \vee (x \wedge 1) && \text{(Identity of } \wedge\text{)} \\
&= x \wedge (y \vee 1) && \text{(Distributivity of } \wedge \text{ over } \vee\text{)} \\
&= x \wedge 1 && \text{(Annihilation of } \vee\text{)} \\
&= x && \text{(Identity of } \wedge\text{)}
\end{aligned}$$

(c)

$$\begin{aligned}
y' \wedge ((x \vee y) \wedge x') &= y' \wedge (x' \wedge (x \vee y)) && \text{(Commutatitivity of } \wedge\text{)} \\
&= (y' \wedge x') \wedge (x \vee y) && \text{(Associativity of } \wedge\text{)} \\
&= (x' \wedge y') \wedge (x \vee y) && \text{(Commutatitivity of } \wedge\text{)} \\
&= (x \vee y)' \wedge (x \vee y) && \text{(De Morgan's, ' over } \vee\text{)} \\
&= (x \vee y)' \wedge (x \vee y)'' && \text{(Double complement)} \\
&= 0 && \text{(Complement with } \wedge\text{)}
\end{aligned}$$

Discussion

- For top marks each rule should be on its own line, but multiple applications of the same rule on one line are ok.
- Minor errors include: one or two incorrect rule names (not counting multiple occurrences) ignoring small typos; one or two rule omissions (name or logical step – i.e. two rules on the same line). Double complementation is a commonly omitted rule.
- Major errors include: Two+ minor errors; omitting all rule names; unfinished proofs.
- Good progress includes: One or two correct logical steps.

Problem 4* (4 marks)

Show that there are no three element Boolean Algebras.

Solution

Suppose $(T, \vee, \wedge, ', 0, 1)$ is a Boolean Algebra with $|T| = 3$. Let $x \in T$ be an element which is not equal to 0 or 1. Note that $0 \neq 1$, as otherwise, by Annihilation and Identity of $\vee$:

$$x = (x \vee 0) = (x \vee 1) = 1,$$

which contradicts the fact that $x \neq 1$. Therefore $T = \{0, 1, x\}$.
We know that $x' \in T$, so $x' = 0, 1,$ or x. If $x' = 0$, then by Identity of $\vee$ and Complement:

$$x = x \vee 0 = x \vee x' = 1,$$

which is a contradiction. If $x' = 1$ then by Identity of $\wedge$ and Complement:

$$x = x \wedge 1 = x \wedge x' = 0,$$

which is a contradiction. Finally, if $x' = x$, then by Idempotence of $\vee$ and Complement:

$$x = x \vee x = x \vee x' = 1,$$

which is also a contradiction. Therefore we cannot define the complement of x without violating a law of Boolean Algebra. So a three element Boolean Algebra cannot exist.

Discussion

- For full marks, a clear, logical argument must be made. Use of derived laws is acceptable.
- Minor errors include omitting rule names
- Major errors include 3+ minor errors; a lack of clarity; or a lack of logical reasoning
- Shows progress includes any logical argument that could potentially result in a proof.

Problem 5 (12 marks)

Prove or disprove the following logical equivalences:

(a) $\neg(p \to q) \equiv (\neg p \to \neg q)$ (4 marks)

(b) $((p \wedge q) \to r) \equiv (p \to (q \to r))$ (4 marks)

(c) $((p \vee (q \vee r)) \wedge (r \vee p)) \equiv ((p \wedge q) \vee (r \vee p))$ (4 marks)

Proof assistant

https://cgi.cse.unsw.edu.au/∼cs9020/cgi-bin/logic/22T2/prop_logic/assignment3b

Solution

(a) This is false. Consider the assignment $v(p) = v(q) = 1$ then $v(\neg(p \to q)) = 0$ whereas $v(\neg p \to \neg q) = 1$.

(b)

$$\begin{aligned}
((p \wedge q) \to r) &\equiv \neg(p \wedge q) \vee r && \text{(Implication)} \\
&\equiv (\neg p \vee \neg q) \vee r && \text{(De Morgan's, } \neg \text{ over } \wedge) \\
&\equiv \neg p \vee (\neg q \vee r) && \text{(Associativity of } \vee) \\
&\equiv p \to (\neg q \vee r) && \text{(Implication)} \\
&\equiv (p \to (q \to r)) && \text{(Implication)}
\end{aligned}$$

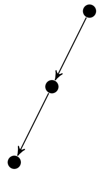
(c)

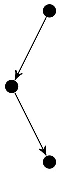
$$\begin{array}{rcll} ((p \lor (q \lor r)) \land (r \lor p)) & \equiv & ((p \lor q) \lor r) \land (r \lor p) & \text{(Associativity of } \lor\text{)} \\ & \equiv & (r \lor (p \lor q)) \land (r \lor p) & \text{(Commutatitivity of } \lor\text{)} \\ & \equiv & r \lor ((p \lor q) \land p) & \text{(Distributivity of } \lor \text{ over } \land\text{)} \\ & \equiv & r \lor (p \land (p \lor q)) & \text{(Commutatitivity of } \land\text{)} \\ & \equiv & r \lor ((p \land p) \lor (p \land q)) & \text{(Distributivity of } \land \text{ over } \lor\text{)} \\ & \equiv & r \lor (p \lor (p \land q)) & \text{(Idempotence of } \land\text{)} \\ & \equiv & (r \lor p) \lor (p \land q) & \text{(Associativity of } \lor\text{)} \\ & \equiv & ((p \land q) \lor (r \lor p)) & \text{(Commutatitivity of } \lor\text{)} \end{array}$$

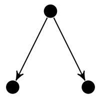
Problem 6 (16 marks)

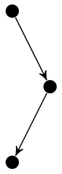
Recall from Assignment 2 the definition of a binary tree data structure: either an empty tree, or a node with two children that are trees.

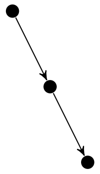
Let $T(n)$ denote the number of binary trees with n nodes. For example $T(3) = 5$ because there are five binary trees with three nodes:











(a) Using the recursive definition of a binary tree structure, or otherwise, derive a recurrence equation for $T(n)$. (6 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. (2 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 $T(n)$.* (4 marks)

A well-formed formula is in **Negated normal form** if it consists of just $\land$, $\lor$, and literals (i.e. propositional variables or negations of propositional variables). For example, $(p \lor (\neg q \land \neg r))$ is in negated normal form; but $(p \lor \neg(q \lor r))$ is not.

Let $F(n)$ denote the number of well-formed, negated normal form formulas[1] 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)

Remark

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

---

[1] Note: we do not assume $\land$ and $\lor$ are associative

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 $T(m).T(n - m - 1)$ 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,

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

(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 $B(n)$:

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

(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 $F(n)$:

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

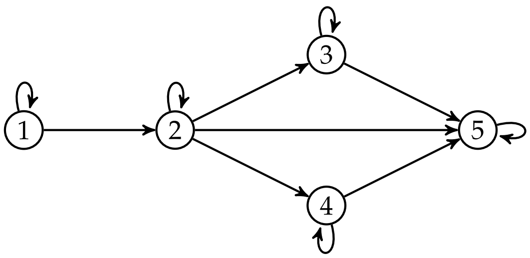
Discussion

- 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 7 (16 marks)

Consider the following directed graph:



and consider the following process:

- Initially, start at 1.
- At each time step, choose one of the outgoing edges from your current location uniformly at random, and follow it to the next location. For example, if your current location was 2, then with probability $\frac{1}{4}$ you would move to 3; with probability $\frac{1}{4}$ you would move to 4; with probability $\frac{1}{4}$ you would move to 5; and with probability $\frac{1}{4}$ you would stay at 2.

Let $p_1(n)$, $p_2(n)$, $p_3(n)$, $p_4(n)$, $p_5(n)$ be the probability your location after n time steps is 1, 2, 3, 4, or 5 respectively. So $p_1(0) = 1$ and $p_2(0) = p_3(0) = p_4(0) = p_5(0) = 0$.

(a) Express $p_1(n+1)$, $p_2(n+1)$, $p_3(n+1)$, $p_4(n+1)$, and $p_5(n+1)$ in terms of $p_1(n)$, $p_2(n)$, $p_3(n)$, $p_4(n)$, and $p_5(n)$. (5 marks)

(b) Prove ONE of the following:

(i) For all $n \in \mathbb{N}$: $p_1(n) = \frac{1}{2^n}$ (5 marks)

(ii) For all $n \in \mathbb{N}$: $p_2(n) = 2\left(\frac{1}{2^n} - \frac{1}{4^n}\right)$ (6 marks)

(iii) For all $n \in \mathbb{N}$: $p_3(n) = p_4(n) = (n-2)\frac{1}{2^n} + \frac{2}{4^n}$ (7 marks)

(iv) For all $n \in \mathbb{N}$: $p_5(n) = 1 - (2n-1)\frac{1}{2^n} - \frac{2}{4^n}$ (8 marks)

Note

Clearly state which identity you are proving. A maximum of 8 marks is available for this question and marks will be awarded based on level of technical ability demonstrated. You may assume the identities which you are not proving.

(c) For each $n \in \mathbb{N}$ let X_n be the random variable that has value:

- 0 if your location at time n is 1;
- 1 if your location at time n is 2;
- 2 if your location at time n is 3 or 4; and
- 3 if your location at time n is 5

(i. e. X_n is the length of the longest path from 1 to your location at time n).

What is the expected value of X_3? (3 marks)

Remark

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

Solution

(a) Consider first $p_5(n+1)$. The event that we are in state 5 at time $n+1$ is dependent on the following events:

- We were in state 5 at time n (probability $p_5(n)$) and remained there (probability 1)
- We were in state 4 at time n (probability $p_4(n)$) and moved to state 5 (probability $\frac{1}{2}$)
- We were in state 3 at time n (probability $p_3(n)$) and moved to state 5 (probability $\frac{1}{2}$)
- We were in state 2 at time n (probability $p_2(n)$) and moved to state 5 (probability $\frac{1}{4}$)

Therefore,

$$p_5(n+1) = p_5(n) + \frac{1}{2} \cdot p_4(n) + \frac{1}{2} \cdot p_3(n) + \frac{1}{4} \cdot p_2(n).$$

We can establish similar equations for the other values as:

$$\begin{aligned}
p_1(n+1) &= \frac{1}{2} \cdot p_1(n) \\
p_2(n+1) &= \frac{1}{2} \cdot p_1(n) + \frac{1}{4} \cdot p_2(n) \\
p_3(n+1) &= \frac{1}{4} \cdot p_2(n) + \frac{1}{2} \cdot p_3(n) \\
p_4(n+1) &= \frac{1}{4} \cdot p_2(n) + \frac{1}{2} \cdot p_4(n) \\
p_5(n+1) &= \frac{1}{4} \cdot p_2(n) + \frac{1}{2} \cdot p_3(n) + \frac{1}{2} \cdot p_4(n) + p_5(n)
\end{aligned}$$

(b) It is possible to prove all results simultaneously by induction, however the question states that we may assume the identities not being proved. We will prove (iv).

Let $P(n)$ be the proposition that $p_5(n) = 1 - (2n-1)\frac{1}{2^n} - \frac{2}{4^n}$. We will prove that $P(n)$ holds for all $n \geq 0$.

Base case: $n = 0$. We have $p_5(0) = 0$ (given) and when $n = 0$,

$$1 - (2n-1)\frac{1}{2^n} - \frac{2}{4^n} = 1 + \frac{1}{1} - \frac{2}{1} = 0.$$

So $P(0)$ holds.

Inductive case. Assume $P(n)$ holds, that is $p_5(n) = 1 - (2n-1)\frac{1}{2^n} - \frac{2}{4^n}$.

From (a), we have

$$\begin{aligned}
p_5(n+1) &= \tfrac{1}{4} \cdot p_2(n) + \tfrac{1}{2} \cdot p_3(n) + \tfrac{1}{2} \cdot p_4(n) + p_5(n) \\
&= \tfrac{1}{4} \cdot 2\left(\tfrac{1}{2^n} - \tfrac{1}{4^n}\right) + \tfrac{1}{2}\left(\cdot(n-2)\tfrac{1}{2^n} + \tfrac{2}{4^n}\right) + \tfrac{1}{2}\left(\cdot(n-2)\tfrac{1}{2^n} + \tfrac{2}{4^n}\right) + p_5(n) && \text{(Given)} \\
&= \left(\tfrac{1}{2^{n+1}} - \tfrac{2}{4^{n+1}}\right) + \left((n-2)\tfrac{1}{2^n} + \tfrac{2}{4^n}\right) + p_5(n) \\
&= (2n-3)\tfrac{1}{2^{n+1}} + \tfrac{6}{4^{n+1}} + p_5(n) \\
&= (2n-3)\tfrac{1}{2^{n+1}} + \tfrac{6}{4^{n+1}} + 1 - (2n-1)\tfrac{1}{2^n} - \tfrac{2}{4^n} && \text{(IH)} \\
&= 1 + ((2n-3) - (4n-2))\tfrac{1}{2^{n+1}} + \tfrac{6-8}{4^{n+1}} \\
&= 1 - (2(n+1)-1)\tfrac{1}{2^{n+1}} - \tfrac{2}{4^{n+1}}
\end{aligned}$$

So $P(n+1)$ holds. Therefore, by induction, $P(n)$ holds for all $n \in \mathbb{N}$.

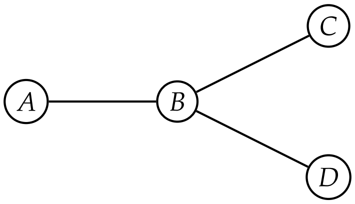
(c) The expected value of X_3 is given by:

$$\begin{aligned}
E(X_3) &= 0 \cdot p_1(3) + 1 \cdot p_2(3) + 2 \cdot (p_3(3) + p_4(3)) + 3 \cdot p_5(3) \\
&= 0\tfrac{1}{8} + 1 \cdot \tfrac{7}{32} + 2 \cdot \tfrac{10}{32} + 3 \cdot \tfrac{11}{32} \\
&= \tfrac{60}{32} = \tfrac{15}{8}
\end{aligned}$$

Problem 8 (10 marks)

In this question we are going to look at modelling the spread of a virus in a network (or news in a social network).

Consider the following graph:



and consider the following process:

- Initially, at time $n = 0$, A is infected and no other vertices are infected.
- At each time step, each infected vertex does the following:
  - for each uninfected neighbour, spread the infection to that vertex with probability $\frac{1}{2}$.

So if A and B were infected and C and D were not, then in one time step, the virus would spread to both C and D with probability $\frac{1}{4}$; spread to C only with probability $\frac{1}{4}$; spread to D only with probability $\frac{1}{4}$; and not spread any further with probability $\frac{1}{4}$.

Let $p_A(n)$, $p_B(n)$, $p_C(n)$, $p_D(n)$ be the probability that A, B, C, D (respectively) are infected after n time steps. So $p_A(0) = 1$ and $p_B(0) = p_C(0) = p_D(0) = 0$.

(a) Express $p_D(n+1)$ in terms of $p_A(n)$, $p_B(n)$, $p_C(n)$ and $p_D(n)$. (4 marks)

(b) Find an expression for $p_D(n)$ in terms of n only. You do not need to prove the result, but you should briefly justify your answer. *Hint: Try to relate this system with Question 7* (4 marks)

(c)* What is the expected number of infected vertices after $n = 3$ time steps? (2 marks)

Solution

(a) If D is infected at time $n+1$ there are two (disjoint) cases:

- Either D was infected at time n: with probability $p_D(n)$, or
- D was not infected at time n, and D became infected from one of his infected neighbours (i.e. B): with probability $(1 - p_D(n))(\frac{1}{2} \cdot p_B(n))$

These events are disjoint, so we can add their probabilities to obtain:

$$p_D(n+1) = p_D(n) + \frac{1}{2}(1 - p_D(n)) \cdot p_B(n)$$

(b) For each time n, consider the sample space of sets of vertices where an outcome indicates which vertices are infected at time n. We note that the current set-up gives rise to only five outcomes (i.e. states of the system) that possibly have non-zero probability at any time:

1. $\{A\}$: i.e. only A is infected;
2. $\{A, B\}$: i.e. only A and B are infected;
3. $\{A, B, C\}$;
4. $\{A, B, D\}$; and
5. $\{A, B, C, D\}$

We note that the event "D is infected at time n" corresponds to the set of outcomes $\{4, 5\}$.

With each of these outcomes we can assign a probability which will change as n changes, but is entirely dependent on the state of the system at the previous time step. For example, if the system is in the state corresponding to outcome 2 (only A and B are infected) at time n, then at time $n+1$ the system will be in one of four states:

- Outcome 2 (B infects nobody): Probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
- Outcome 3 (B infects C only): Probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
- Outcome 4 (B infects D only): Probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$
- Outcome 5 (B infects both C and D): Probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

We can model these transitions between outcomes as time increases using a Markov Chain – indeed, the Markov Chain from Question 7 is the result. Thus the probability that the system will be in state 4 at time n is given by $p_4(n)$ from Question 7.

Therefore the probability that D is infected at time n is:

$$p_D(n) = p_4(n) + p_5(n) = (n-2)\frac{1}{2^n} + \frac{2}{4^n} + 1 - (2n-1)\frac{1}{2^n} - \frac{2}{4^n} = 1 - (n+1)\frac{1}{2^n}$$

(c) For each n, let Y_n be the random variable that indicates the number of infected vertices at time n. In other words $Y_n(S) = |S|$ for each outcome S in the previously defined sample space. We note that $Y_n = 1 + X_n$ where X_n is the random variable defined in Question 7. Therefore, by the linearity of expectation:

$$E(Y_3) = E(1 + X_3) = 1 + E(X_3) = 1 + \frac{15}{8} = \frac{23}{8}.$$

So, after 3 time steps, we expect $\frac{23}{8} = 2.875$ vertices to be infected.