NATIONAL UNIVERSITY OF SINGAPORE

CS3243: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

(Semester 2, AY2022/2023)

SOLUTIONS

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. Please write your Student Number only. Do not write your name.
2. This assessment paper contains **FIVE** questions and comprises **SIXTEEN** printed pages.
3. Students are required to answer **ALL** questions.
4. This is a CLOSED BOOK assessment. However, you may reference a SINGLE DOUBLE-SIDED A4 CHEAT SHEET.
5. You are allowed to use NUS APPROVED CALCULATORS.
6. If something is unclear, solve the question under reasonable assumptions. State your assumptions clearly in the answer. If you must seek a clarification, the invigilators will only answer questions with Yes/No/No Comment answers.
7. You may not communicate with anyone other than the invigilators in any way.

STUDENT NUMBER: ____________________________________

EXAMINER'S USE ONLY		
Question	Marks	Remarks
1		
2		
3		
4		
5		
TOTAL		

1a. [2 marks] There are n search and rescue personnel that are conducting a search in a dense forest. The goal of the rescuers is to search the forest until the k people that are lost in it (denoted as "lost") are rescued. Note that these k lost are at different locations in the forest.

Assume that this forest is modelled as a 2-dimensional maze-like grid that spans r rows and c columns. The n rescuers all initially start at the same coordinate on the grid. For each time unit, each rescuer may move to the north, south, east, or west (assuming that there isn't a barrier – i.e., a forest wall – blocking their path). The lost do not move.

This problem is modelled as a search problem where each state is defined by the following.

- The current locations of the n rescuers in the current time step, $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$, where each $\boldsymbol{p}_i$ corresponds to the coordinates of the i-th rescuer.
- The condition of the k lost in the current time step, $b_1, \ldots, b_k$, where each b_j is 1 if the j-th lost has been rescued (i.e., found by a rescuer) or 0 if not.

State a reasonable upper bound on the size of the state space for this search problem.

Solution:

$O(2^k(rc)^n)$. Each rescuer can be in rc locations in the grid and each lost can be found or not.

1b. [2 marks] For problems with finite depth and branching factor, Depth-First Search implemented with tree search is ***complete***. True or false? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

Solution:

True. Since the problems correspond to search trees with finite depth and branching factor, DFS will be able to traverse the entire tree. This means that if a solution exists, it will find it, and when one does not, it would be able to determine as much.

1c. [2 marks] Depth-First Search always expands at least as many nodes as A* Search with an admissible heuristic. True or false? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

Solution:

False. On a certain problem DFS might find a solution without having to perform any backtracking at all. However, A* would have to explore all paths with a total $f(n) = g(n) + h(n)$ value that is less than the actual cost of the optimal path.

1d. [2 marks] Consider the following implementation of the A* Search algorithm adopting a variant of graph search. Assume that a consistent heuristic, h, is used.

```
function A*-SEARCH(problem, frontier)
    reached ← empty set
    frontier ← INSERT(MAKE-NODE(problem.initial_state), frontier)
    loop
        if frontier is empty then
            return failure
        end if
        node ← POP-FRONT(frontier)
        if node.state is not in reached then
            reached ← ADD(node.state, reached)
            for successor in GET-SUCCESSORS(problem, node.state) do
                frontier ← INSERT(MAKE-NODE(successor), frontier)
                if GOAL-TEST(problem, successor) then
                    return successor
                end if
            end for
        end if
    end loop
end function
```

You may assume that all functions called in `A*-SEARCH(..)` will function correctly.

List any of the following options that are **True**.

Option **A**: Any `node.state` may be revisited.

Option **B**: The algorithm is no longer complete.

Option **C**: The algorithm could return an incorrect solution.

Option **D**: None of the above

*Note that you may pick **more than one** option, except in the case where you pick option **D**.*

Solution:

B. Notice that the goal test is never applied to the initial state.
C. The given algorithm expands fewer nodes to find a goal but does not always find the optimal path. Note that "incorrect" means that it is "not optimal" in this case.

1e. [2 marks] Assuming application to the A* Search algorithm, succinctly explain why a heuristic that expands more nodes may be preferred over one which expands fewer nodes.

Solution:

(A) An admissible heuristic might expand more nodes than an inadmissible heuristic but would be guaranteed to find the optimal answer.

OR

(B) Even among two admissible heuristics, we might prefer a less accurate heuristic that is easier to compute per state.

1f. Suppose that you are developing an intelligent agent that must reconstruct sentences. Each given sentence corresponds to a language that you know nothing about. Furthermore, the word order in the sentence has also been lost. The agent must reconstruct the sentences such that they have a sensible meaning.

For example, assuming a sentence based on the English language, the words {***final***, ***very***, ***exam***, ***enjoyable***, ***is***, ***this***} can be reconstructed into the valid ordering: "***this final exam is very enjoyable***".

However, as mentioned above, the language used is completely unknown to you. Thankfully, you have access to an oracle that assigns a score to every sequence you propose. This oracle scores proposed sentences based on how ridiculous the sentence is. Referring to the example above, the oracle would consider "***exam very enjoyable***" to be **less** ridiculous than "***enjoyable very exam***".

The developer of this agent has decided to solve this problem as a local search problem. More specifically, for this search problem, each state corresponds to a particular ordering of the given sentence (which is a complete state formulation). At each state, the possible actions correspond to the swapping of a one pair of words. Thus, at each state, assuming n unique words in the sentence, there are ${}^{n}C_2$ (i.e., n choose 2) possible actions.

Suppose that the developer wishes to apply the hill-climbing algorithm (with random restarts) to solve this problem. Advise the developer by answering the following questions.

Note that the questions in parts **(i)** *and* **(ii)** *below are based on the above context* (**Q1f**).

(i) [1 mark] Hill-Climbing Search with random restarts will always find a valid sentence in reasonable time. True or false? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

Solution:

False. Hill-climbing search is likely to converge to a local optimum.

(ii) [2 marks] Stochastic Hill-Climbing Search with random restarts will always find a valid sentence in reasonable time. True or false? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

Solution:

False. Stochastic hill-climbing search is less prone to converge to an arbitrary local optimum and is more likely to find a better solution. However, it is not guaranteed to find the global optimum. Theoretically, it should find the global optimum as running time reaches infinity (note, however, that this is also the case for brute force search). This is especially the case given the size of the search space, which is O(n!).

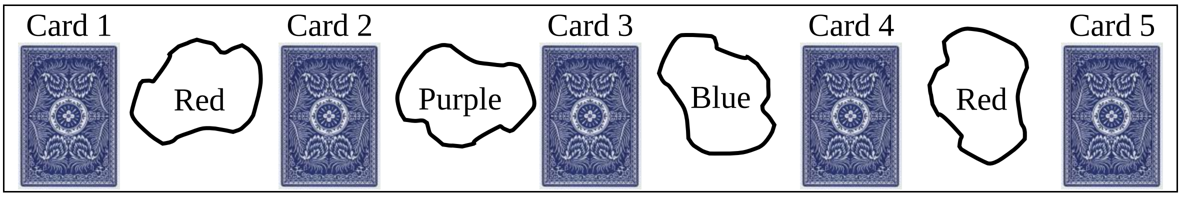
2. A magician's trick starts with 4 identical decks, each with 52 different cards. The reverse side of all the cards from all 4 decks have the same picture. This means that when only looking at the reverse side (i.e., not the front) of the cards, one cannot distinguish one card from another when the pair of cards are drawn randomly from all 4 decks. However, each card in each of the decks can be uniquely identified when looking at the front.

The magician calls for 3 volunteers from the audience. One of the volunteers must pick 1 card from one of the 4 decks (denote this as the **key deck**), while the other two volunteers must each pick 2 cards from any of the other decks (i.e., from any of the decks that are not the key deck). However, the volunteers must pick unique cards (i.e., when looking at the front of all 5 chosen cards, all the cards must be different).

Given that the magician has not seen which cards (i.e., the front of the cards) are chosen by each volunteer, the magician's goal is to identify the card that was chosen from the key deck.

In order to achieve this goal, the magician has marked the cards in each deck with a different chemical so that the cards from the different decks may be identified with customised glasses (i.e., 4 chemicals are used; each chemical is applied to exactly one deck). Essentially, by using the glasses, the magician can thus distinguish the cards from the different decks since each chemical is seen as a different colour.

However, due to a mishap from the magician's assistant, the chemicals on the cards were somehow smeared on the table. The colours from adjacent cards have been mixed! The magician sees the following on the table.



Thankfully, all is not lost since it is known what the superseding colour is when two chemicals are mixed. More specifically, the following is known.

Deck	Colour
Non-Key Deck 1	Red
Non-Key Deck 2	Purple
Key Deck	Blue
Non-Key Deck 3	Green

Superseding Colour Chart
Red > Purple, Blue, Green
Purple > Blue, Green
Blue > Green
Green > ∅

Note that given two colours, C_1 and C_2, $C_1 > C_2 \Rightarrow C_1$ will supersede C_2 when the chemicals corresponding to C_1 and C_2 are mixed.

Help the assistant (i.e., ensure the magician does not fire the assistant) by first formulating this problem as a constraint satisfaction problem (CSP), and then solving it. Let the variables in this CSP be X_1, X_2, X_3, X_4, and X_5, where X_i denotes the i-th card, and let the domain for each card be $\{R, P, B, G\}$, corresponding to Red, Purple, Blue, and Green, respectively.

Note that the questions in parts **(i)** *to* **(vi)** *below are based on the above context* (**Q2**). *No error carried forward (ECF) will be considered.*

(i) [4 marks] Define the binary and/or unary constraints implied by the cards and observed colours on the magician's table.

Solution:

Unary:

- $X_2 \neq R$
- $X_3 \neq R$
- $X_3 \neq P$
- $X_4 \neq R$
- $X_4 \neq P$

Binary:

- $X_1 = R \vee X_2 = R$
- $X_2 = P \vee X_3 = P$
- $X_3 = B \vee X_4 = B$
- $X_4 = R \vee X_5 = R$

Note that there are also several global constraints (**that need not be specified**), which constrain X_1, X_2, X_3, X_4, X_5 to only include only one blue card:

- $(X_1 = B) \Rightarrow (X_2 \neq B) \wedge (X_3 \neq B) \wedge (X_4 \neq B) \wedge (X_5 \neq B)$
- $(X_2 = B) \Rightarrow (X_1 \neq B) \wedge (X_3 \neq B) \wedge (X_4 \neq B) \wedge (X_5 \neq B)$
- $(X_3 = B) \Rightarrow (X_1 \neq B) \wedge (X_2 \neq B) \wedge (X_4 \neq B) \wedge (X_5 \neq B)$
- $(X_4 = B) \Rightarrow (X_1 \neq B) \wedge (X_2 \neq B) \wedge (X_3 \neq B) \wedge (X_5 \neq B)$
- $(X_5 = B) \Rightarrow (X_1 \neq B) \wedge (X_2 \neq B) \wedge (X_3 \neq B) \wedge (X_4 \neq B)$

(ii) [3 marks] Apply arc-consistency and node-consistency and to narrow down the domains. Define the remaining domains for each variable in the CSP after arc-consistency and node-consistency are enforced.

Solution:

X_1: $\{R\}$
X_2: $\{P\}$
X_3: $\{B, G\}$
X_4: $\{B, G\}$
X_5: $\{R\}$

Note that this solution does not require the application of the global constraints.

(iii) [1 mark] If you wanted to find a solution at this point, what variables would the minimum remaining values (MRV) ordering heuristic tell you to assign first?

Solution:

X_1, X_2, or X_5.

Note that this solution does not require the application of the global constraints.

(iv) [1 mark] List all the solutions to this CSP or state that none exist.

Solution:

X_1	X_2	X_3	X_4	X_5
R	P	B	G	R
R	P	B	B	R
R	P	G	B	R

Note that this solution does not require the application of the global constraints. However, answers that do, and thus omit the second assignment above, will also be accepted.

(v) [1 mark] Assuming that the magician cannot afford to be wrong, is it all right to pick Card 4 (i.e., X_4)? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

Solution:

No, as there is a chance that it could be from a deck that is not the key deck.

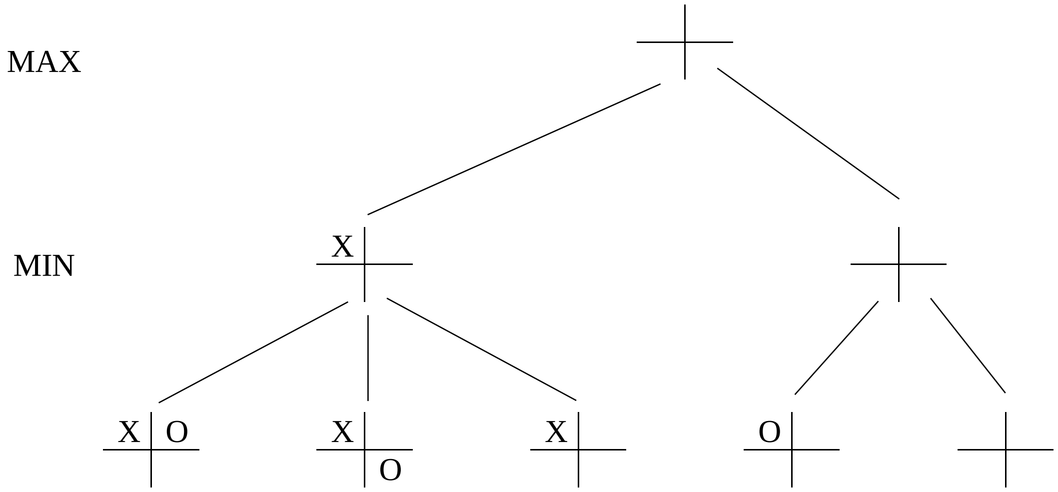
(vi) [2 marks] Imagine that there is a CSP solver, which, when given a formulation of a CSP (i.e., the variables, domains, and unary and binary constraints), will output true if at least one solution exists. Using such a CSP solver, describe how the assistant and the magician could prove that X_i is the card from the key deck.

Solution:

Add a constraint that $X_i \neq B$. If the CSP solver returns false, then X_i is the card from the key deck since it means that there is no way to satisfy the configuration of cards and colours except for the cases where X_i is from the key deck.

3a. Consider the zero-sum game of 2×2 tic-tac-toe where each player has the additional option of passing (i.e., marking no square). Assume that "X" goes first, and "O" goes second.

Below is the game tree specified to a depth of 2.



Note that the questions in parts **(i)** *to* **(iii)** *below are based on the above context* (**Q3a**). *No error carried forward (ECF) will be considered.*

(i) [2 marks] The given tree, specified up to depth 2, represents all possible moves within the first two turns. True or false? Succinctly justify your answer.

Note that you will not be awarded any marks without a valid justification.

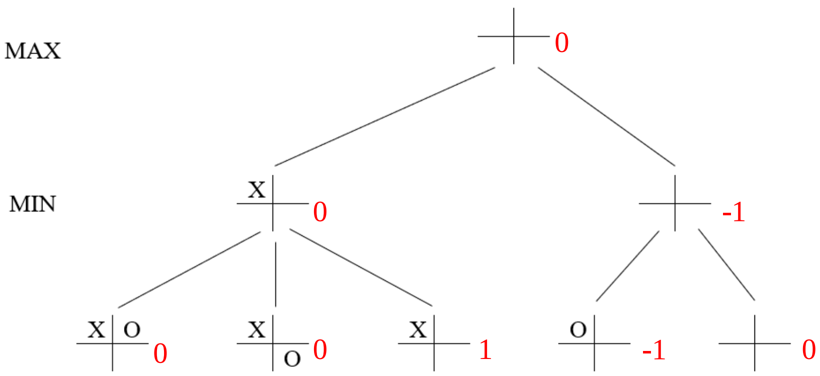
Solution:

True. Since all other positions are just reflections or rotations of the given board states.

(ii) [2 marks] Suppose that the evaluation function used is the number of Xs minus the number of Os. Define the utility values based on this evaluation function for the leaf nodes, and then apply backwards induction (i.e., the Minimax strategy) to define the utility values for every other node in the tree.

Note that you must clearly write the MAX utility value on the ***right*** *of each associated node.*

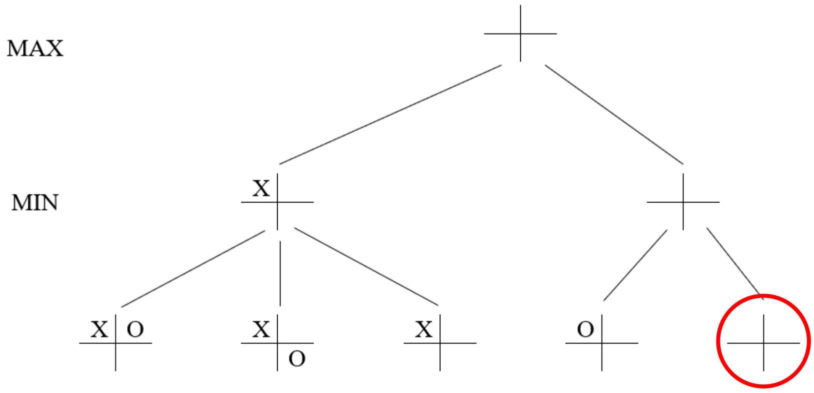
Solution:



(iii) [2 marks] Assume that α-β (alpha-beta) pruning is applied in a left-to-right exploration of the given game tree, with values given by Part **(ii)**. Circle the nodes that would be pruned.

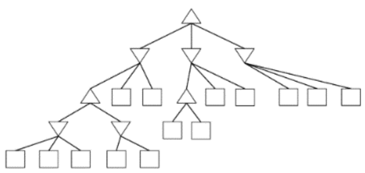
*Note that you should **not** circle any nodes if you feel that no node will be pruned.*

Solution:

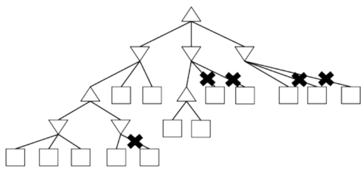


3b. [6 marks] Assume that α-β (alpha-beta) pruning is applied to each of the following game trees in a left-to-right exploration. Further, assume zero-sum games. Select the options where the given pruning **is possible** under some choice of pay-off values. Note that the crossed-off edges indicate where pruning has occurred.

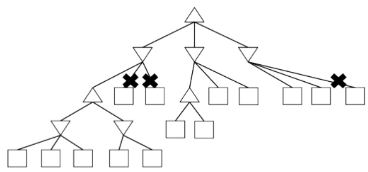
Option **A**:



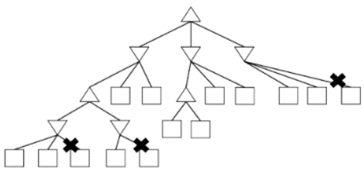
Option **B**:



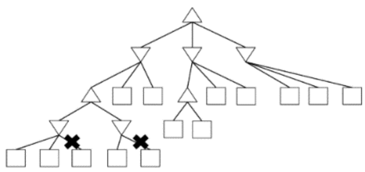
Option **C**:



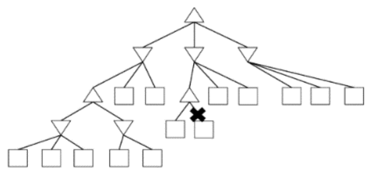
Option **D**:



Option **E**:



Option **F**:



Option **G**: None of the above

*Note that you may pick **more than one** option, except in the case where you pick option **G**.*

Further, note that negative marking will be applied to this question – i.e., your score will be based on the total number of options correctly specified as possible or impossible minus the total number of options incorrectly specified as possible or impossible.

Solution:

A, B

4. The Pigeonhole principle states that, given n items and m boxes, we must satisfy the following:

Sentence 1: Each of the n items must be assigned to exactly one of the m boxes.

Sentence 2: Each of the m boxes must contain at most one of the n items.

Note that the questions in parts **(i)** *and* **(ii)** *below are based on the above context* (**Q4**). *No error carried forward (ECF) will be considered.*

(i) [6 marks] Let $X_{i,j}$ denote that the i-th item (out of the n items) is placed in the j-th box (out of the m boxes). Using this notation, describe the Knowledge Base (KB) representing the Pigeonhole problem where n = 2 and m = 3. You must express this KB in conjunctive normal form (CNF).

Solution:

KB:

$$\left.\begin{array}{l} X_{1,1} \vee X_{1,2} \vee X_{1,3} \\ \neg X_{1,1} \vee \neg X_{1,2} \\ \neg X_{1,1} \vee \neg X_{1,3} \\ \neg X_{1,2} \vee \neg X_{1,3} \end{array}\right\} \text{Item 1 must be assigned to exactly one box}$$

$$\left.\begin{array}{l} X_{2,1} \vee X_{2,2} \vee X_{2,3} \\ \neg X_{2,1} \vee \neg X_{2,2} \\ \neg X_{2,1} \vee \neg X_{2,3} \\ \neg X_{2,2} \vee \neg X_{2,3} \end{array}\right\} \text{Item 2 must be assigned to exactly one box}$$

$$\left.\begin{array}{l} \neg X_{1,1} \vee \neg X_{2,1} \\ \neg X_{1,2} \vee \neg X_{2,2} \\ \neg X_{1,3} \vee \neg X_{2,3} \end{array}\right\} \text{Each box has at most one item}$$

Note that you can draw the truth tables corresponding to each item under Sentence 1, and each box under Sentence 2 to validate the above.

(ii) [6 marks] Using resolution, prove that Sentence 1 and Sentence 2 are **false** for all cases where $n = m + 1$.

Solution:

The given case may be exemplified with $n = 2$:

- Here, $m = 1$
- The resultant query, α, is given by $\alpha \equiv \neg((X_{1,1} \land X_{2,1}) \land (\neg X_{1,1} \lor \neg X_{2,1}))$
    - Sentence 1 gives $X_{1,1} \land X_{2,1}$ (since each item must be assigned to 1 box)
    - Sentence 2 gives $\neg X_{1,1} \lor \neg X_{2,1}$ (since each box must only contain 1 item)
    - We use ¬ over the query since we wish to prove it is false.
- During resolution, we want to show that KB ⊨ ¬α
    - $\neg\alpha \equiv (X_{1,1} \land X_{2,1}) \land (\neg X_{1,1} \lor \neg X_{2,1})$
      $\equiv X_{1,1} \land X_{2,1} \land (\neg X_{1,1} \lor \neg X_{2,1})$ // now in CNF
- Without even looking at the KB, we have the following:
    - Resolve $X_{1,1}$ and $(\neg X_{1,1} \lor \neg X_{2,1})$ to get the resolvent $\neg X_{2,1}$
    - Resolve $\neg X_{2,1}$ and $X_{2,1}$ to get the resolvent that is the empty clause ∅
- Thus, by contradiction we have proven that the statement that Sentence 1 and Sentence 2 are false for the case of $n = m + 1$, when $n = 2$.

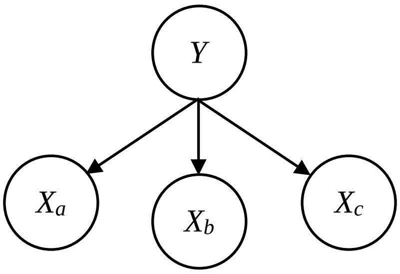
Intuitively, when $n = 3$, we notice that accordingly, $m = 2$, so we may conveniently assign the newly added item to the newly added box (satisfying both Sentence 1 and Sentence 2), and then we arrive at the base case above once more.

5a. Consider the following table, which contains the performance data for 10 students taking the CS9999 course. The data for the i-th student is given in column i. Each student is represented by 3 Boolean variables, X_a, X_b, and X_c, as well as one class label, Y.

- If $X_a = 1$, then the student did well in Assessment 1, else, $X_a = 0$.
- If $X_b = 1$, then the student did well in Assessment 2, else, $X_b = 0$.
- If $X_c = 1$, then the student did well in Assessment 3, else, $X_c = 0$.
- The class label $Y = 1$ if the student did well in the CS9999 course, else, $Y = 0$.

i	1	2	3	4	5	6	7	8	9	10
X_a	0	0	0	0	0	1	0	1	0	1
X_b	0	0	0	0	0	0	0	0	1	1
X_c	1	1	0	0	1	1	1	1	0	0
Y	0	0	0	0	1	1	1	1	1	1

Assume that a Naïve Bayes model is to be utilised. More specifically, assume that we are to adopt the following Bayesian Network.



Note that the questions in parts **(i)** *and* **(ii)** *below are based on the above context* (**Q5a**). *No error carried forward (ECF) will be considered.*

(i) [2 marks] Complete the following probability tables – i.e., fill in the last column in each of the tables given below.

Solution:

Y	P(Y)
0	2/5
1	3/5

Y	X_a	P(X_a|Y)
0	0	1
0	1	0
1	0	1/2
1	1	1/2

Y	X_b	P(X_b|Y)
0	0	1
0	1	0
1	0	2/3
1	1	1/3

Y	X_c	P(X_c|Y)
0	0	1/2
0	1	1/2
1	0	1/3
1	1	2/3

(ii) [2 mark] Assume that we observe the assessments for a new student, where $X_a = 0$, $X_b = 0$, and $X_c = 1$. Is this student more likely to do well in the CS9999 course (i.e., have $Y = 1$)?

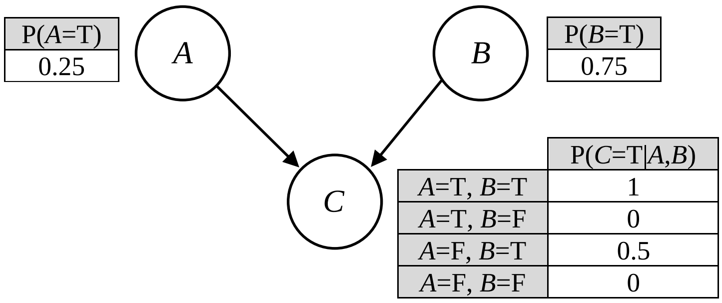
Option **A**: The new student is more likely to do well in CS9999 (i.e., have $Y = 1$)
Option **B**: The new student is more likely to not do well in CS9999 (i.e., have $Y = 0$)
Option **C**: Both $Y = 1$ and $Y = 0$ are equally likely
Option **D**: There is not enough information to determine which outcome is more likely

Note that you may only pick ***one*** *option.*

Solution:

B // $P[Y=1 \mid X_a=0, X_b=0, X_c=1] = 2/15$; $P[Y=0 \mid X_a=0, X_b=0, X_c=1] = 1/5$

5b. Consider the following Bayesian Network.



Note that the questions in parts **(i)** *to* **(iv)** *below are based on the above context* (**Q5b**). *No error carried forward (ECF) will be considered.*

(i) [1 mark] State the formula for the joint probability distribution induced by the above Bayesian Network – i.e., define the expression for $P(A, B, C)$.

Solution:

$P(A,B,C) = P(A)\cdot P(B)\cdot P(C|A,B)$, since the directed graph structure indicates that A and B are independent. Note that they are no longer independent if we condition on C.

(ii) [2 marks] Calculate the probability of $P(C = T)$.

Solution:

$$P(C = T) = \sum_{A,B} P(A)\cdot P(B)\cdot P(C = T \mid A, B)$$
$$= \left(\frac{1}{4}\cdot\frac{3}{4}\cdot 1\right) + \left(\frac{1}{4}\cdot\frac{1}{4}\cdot 0\right) + \left(\frac{3}{4}\cdot\frac{3}{4}\cdot\frac{1}{2}\right) + \left(\frac{3}{4}\cdot\frac{1}{4}\cdot 0\right)$$
$$= \frac{15}{32}$$

(iii) [2 marks] Calculate the probability of P(A = T, B = T).

Solution:

$P(A = T, B = T) = P(A = T) \cdot P(B = T) = \frac{3}{16}$

Since A and B are independent.

(iv) [2 marks] Calculate the probability of P(A = T, B = T | C = T).

Solution:

$$\begin{aligned} P(A = T, B = T \mid C = T) &= \frac{P(A = T, B = T, C = T)}{P(C = T)} \\ &= \frac{P(A = T) \cdot P(B = T) \cdot P(C = T \mid A = T, B = T)}{P(C = T)} \\ &= \left(\frac{1}{4} \cdot \frac{3}{4} \cdot 1\right) / \frac{15}{32} \\ &= \frac{2}{5} \end{aligned}$$

END OF PAPER

BLANK PAGE

BLANK PAGE