CS 188 Spring 2021 — Introduction to Artificial Intelligence — Midterm

- You have approximately 110 minutes.
- The exam is open book, open calculator, and open notes.
- In the interest of fairness, we want everyone to have access to the same information. To that end, we will not be answering questions about the content or making clarifications.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ○ means mark a **single choice**

First name	
Last name	
SID	

For staff use only:

Total	**/??**

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [15 pts] Potpourri

(a) Decide whether the following statements are *True/False*.

(i) [1 pt] Consider a search problem in which the state space includes variable x. Then, any quantities that are a deterministic function of x don't need to be included in the state space.

◯ True ◯ False

(ii) [1 pt] Without any constraints on the utility value, it's impossible to perform pruning on an Expectimax tree.

◯ True ◯ False

(iii) [1 pt] If $N_a > N_b$, then running Monte Carlo Tree Search with N_a rollouts will always yield better decisions than with N_b rollouts.

◯ True ◯ False

(iv) [1 pt] If one fixes temperature T to be 0, simulated annealing will reduce to standard (steepest-ascent) hill climbing.

◯ True ◯ False

(v) [1 pt] Given that a Bayes Net expresses a joint distribution over its variables, there can be conditional independence relationships implied by the joint distribution that are not asserted by the Bayes net topology.

◯ True ◯ False

(vi) [1 pt] Assume that we have $P(X_t|e_{1:t})$ for a standard HMM model. Then the cost of running the forward algorithm to calculate $P(X_{t+1}|e_{1:t+1})$ is $\mathcal{O}(t|X|)$, where $|X|$ is the number of states.

◯ True ◯ False

(b) [4 pts] Below is a list of task environments. For each of the sub-parts, choose all the environments in the list that falls into the specified type.

A: The tic-tac-toe game
B: Self driving cars
C: The "N-queens" search problem (discussed in the local search lecture)
D: The "Weather forecast" problem (predict weather from measurements of wind, pressure, humidity, etc.)

Which of the environments are *dynamic*? ☐ A ☐ B ☐ C ☐ D

Which of the environments are *fully observable*? ☐ A ☐ B ☐ C ☐ D

Which of the environments are *stochastic*? ☐ A ☐ B ☐ C ☐ D

Which of the environments have *known physics*? ☐ A ☐ B ☐ C ☐ D

(c) [2 pts] Consider a search problem with a set of goal states S. Define $h^*(x)$ to be the shortest(optimal) distance from state x to a goal state $s \in S$. Using heuristic function h, which of the following statements are true? Select all that apply. Assume that $h(x) \geq 0$ for all x.

☐ Given $h(x) > h^*(x)$ for some state x, A* graph search cannot find the optimal solution.

☐ Given h is an admissible heuristic, A* graph search cannot find the optimal solution.

☐ Given h is a consistent heuristic, running A* graph search will find the optimal path to every goal state.

☐ Given h is an admissible heuristic, there exists a constant $c > 0$ such that $c * h$ becomes a consistent heuristic.

◯ None of the Above

(d) [3 pts] We propose a variant of the depth-first graph search algorithm as follows: we record the number of times we have visited each state in the state space, and we will not expand a state if we have visited the state at least k times, where k is a positive integer greater than 1. (When $k = 1$, this is equivalent to standard depth-first graph search). Which of the following statements are true? Select all that apply.

☐ The new algorithm is complete, while standard depth-first tree search is not.

☐ Given that there are no cycles in the state-space graph, the worst-case time complexity of the new algorithm is the same as that of standard depth-first tree search.

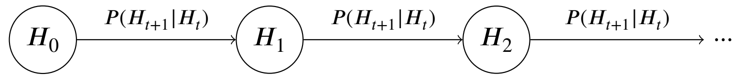
☐ The worst-case space complexity of the new algorithm is the same as that of standard depth-first tree search.

☐ The worst-case space complexity of the new algorithm is the same as that of standard breadth-first tree search.

○ None of the Above

Q2. [15 pts] MangoBot Human Detector

Your startup company MangoBot wants to build robots that delivers packages on the road. One core module of the robot's software is to detect whether a human is standing in front of it. We model the presence of humans with a Markov model:



where $H_t \in \{0, 1\}$ corresponds to a human being absent or present respectively. The initial distribution and the transition probabilities are given as follows:

H_0	$P(H_0)$
0	p
1	$1 - p$

H_t	H_{t+1}	$P(H_{t+1}\|H_t)$
0	0	0.9
0	1	0.1
1	0	0.8
1	1	0.2

(a) Express the following quantities in terms of p:

(i) [1 pt] $P(H_1 = 1) =$

(ii) [1 pt] $\lim_{t\to\infty} P(H_t = 0) =$

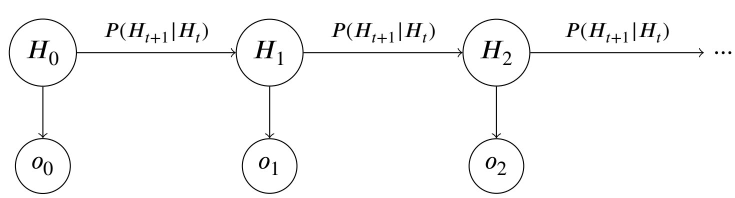
(b) The first-order Markov assumption in the model above can be inaccurate in real-world situations. Some potential ways to improve the model are listed below. For each option, determine whether it is possible to rewrite the process as a first-order Markov process, potentially with a different state representation.

(i) [2 pts] H_t depends not only on H_{t-1} but also on H_{t-2}. ○ Yes ○ No

(ii) [2 pts] H_t depends not only on H_{t-1} but also on $H_{t-2}, H_{t-3}, \cdots, H_{t-k}$ for some fixed $k \geq 3$. ○ Yes ○ No

(iii) [2 pts] H_t depends not only on H_{t-1} but also on $H_{t-2}, H_{t-3}, \cdots, H_1, H_0$. ○ Yes ○ No

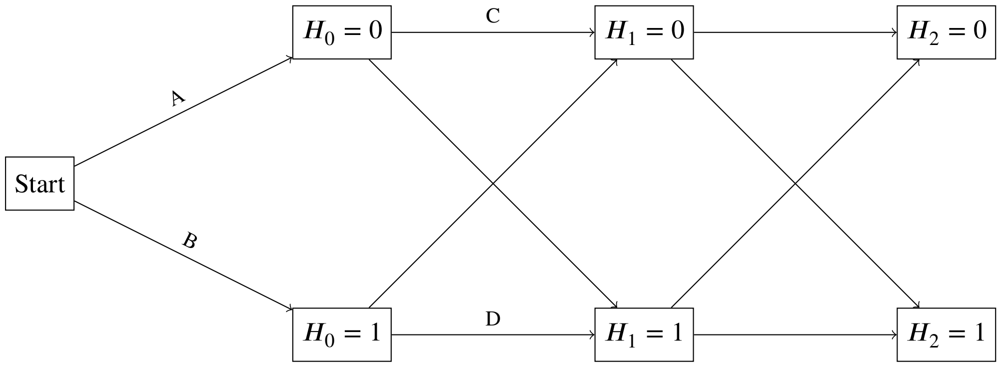
To make things simple, we stick to the original first-order Markov chain formulation. To make the detection more accurate, the company built a sensor that returns an observation O_t each time step as a noisy measurement of the unknown H_t. The new model is illustrated in the figure, and the relationship between H_t and O_t is provided in the table below.



H_t	O_t	$P(O_t\|H_t)$
0	0	0.8
0	1	0.2
1	0	0.3
1	1	0.7

(c) Based on the observed sensor values $o_0, o_1, \cdots, o_t$, we now want the robot to find the most likely sequence $H_0, H_1, \cdots, H_t$ indicating the presence/absence of a human up to the current time.

(i) [2 pts] Suppose that $[o_0, o_1, o_2] = [0, 1, 1]$ are observed. The following "trellis diagram" shows the possible state transitions. Fill in the values for the arcs labeled A, B, C, and D with the product of the transition probability and the observation likelihood for the destination state. The values may depend on p.



(ii) [3 pts] There are two possible most likely state sequences, depending on the value of p. Complete the following (Write the sequence as "x,y,z" (without quotes), where x, y, z are either 0 or 1):
Hint: it might be helpful to complete the labelling of the trellis diagram above.

- When $p <$ ☐, the most likely sequence H_0, H_1, H_2 is ☐.

- Otherwise, the most likely sequence H_0, H_1, H_2 is ☐.

(d) [2 pts] True or False: For a fixed p value and observations $\{o_0, o_1, o_2\}$ in general, H_1^*, the most likely value for H_1, is always the same as the value of H_1 in the most likely sequence H_0, H_1, H_2. ◯ True ◯ False

Q3. [14 pts] Adventurers Assemble

Chuck and Nancy are exploring a cave off the coast of Maine when Chuck gets separated from Nancy. He decides to model his situation as a search problem to try to find Nancy.

Chuck draws out the cave as an X by Y grid on his magical map, and keeps track of his current location as Z. He also marks the locations of N slime monsters within the cave as $S_1, \cdots, S_N$. The slimes are stationary but very dangerous, so Chuck must make sure to avoid them as he explores.

Chuck observes Nancy's current location on his map, and marks her location as $T(s)$ for any given state s. On each turn Chuck chooses an action from the action set {*left, right, up, down*} and moves one square in the direction he chooses, although any action that would bump into a wall results in no movement. Similarly, Nancy chooses an action from the same action set and moves in that direction, which can be different from the action that Chuck takes. Once Chuck has reached the same location as Nancy on a given timestep, the search problem ends. Consider the following heuristic function, where s is the input state:

$$h(n) = \begin{cases} 0 \text{ where } Z(s) = T(s) \\ h_i(s) \text{ otherwise} \end{cases}$$

For the condition in each subpart, determine whether the following options for $h_i(s)$ would allow $h(s)$ to guarantee optimality for A* tree search, A* graph search, neither of them, or both of them given the condition. We will look at different proposals for what $h_i(s)$ could be. $M(A, B)$ is a function that returns the Manhattan distance between two locations in the grid A and B.

(a) Assume the cost of taking actions within a single timestep is 1 (i.e.: for a given timestep, Chuck moving one square and Nancy moving another square costs 1 in total).

(i) [2 pts] $h_1 = M(Z(s), T(s))$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

(ii) [2 pts] $h_2 = \left\lfloor \frac{M(Z(s),T(s))}{2} \right\rfloor$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

(b) Chuck gets more and more tired with each subsequent action. At timestep t, the cost of Chuck's action a_t is 2 times the cost of taking action a_{t-1}, and the cost of taking his first action a_1 is 1. The function $C(a_t)$ returns the cost of Chuck taking action a_t. Nancy isn't tired, so the cost of Nancy's actions is always 1. (i.e.: for a given timestep t, Chuck moving one square and Nancy moving another square costs $C(a_t) + 1$ in total.)

(i) [2 pts] $h_3 = \sum_{j=k}^{n} C(a_j)$ where k = the current timestep, $n = k + \left\lfloor \frac{M(Z(s),T(s))}{2} \right\rfloor$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

(ii) [2 pts] $h_4 = n + \sum_{j=k}^{n} C(a_j)$ where k = the current timestep, $n = k + \left\lfloor \frac{M(Z(s),T(s))}{2} \right\rfloor$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

(c) Assume the cost of taking actions within a single timestep is 1, as in the very first subpart. Chuck must now destroy all N slimes in the cave before reuniting with Nancy. He destroys a slime S_i by moving onto the same square as it, i.e., $Z(s) = S_i$ for $i \in [1, \cdots, N]$. Chuck records the index of slimes that he destroys by adding the index i to a list R. Only Chuck can destroy slimes; Nancy does not have that power. Even if he and Nancy move onto the same square, the search problem will not end until all slimes in the cave are eliminated, when R contains all integers from $1, \cdots, N$. Note that $len(R)$ returns the number of elements currently in the list R.

(i) [3 pts] $h_5(s) = \begin{cases} 0 & \text{if } len(R) \geq N-1 \\ \min_{i \notin R}\{M(Z(s), S_i)\} + \max_{j \notin R, j \neq i^\star}\{M(S_{i^\star}, S_j)\} & \text{otherwise} \end{cases}$

for $i, j \in [1, \cdots, N]$ and $i^\star = \arg\min_i\{M(Z(s), S_i)\}$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

(ii) [3 pts] $h_6(s) = \begin{cases} \left\lfloor \frac{M(Z(s),T(s))}{2} \right\rfloor & \text{if } len(R) \geq N \\ \max_i \left\{M(S_i, T(s))\right\} & \text{otherwise} \end{cases}$

for $i \in [1, \cdots, N]$ and $i \notin R$

- ◯ optimality for A* tree search only
- ◯ optimality for A* graph search only
- ◯ optimality for neither
- ◯ optimality for both

Q4. [20 pts] It's So Logical!

(a) [4 pts] Inspired by learning about propositional logic, you decided to write down some sentences about your two hobbies: **playing go and collecting kettlebells**, using symbols to stand for propositions. You now have this list of logic sentences, and you remember what the English meanings of the sentences were, but you forget the meaning of each symbol!

For each English sentence on the left, there is a corresponding logical sentence on the right, **but it is not necessarily the sentence next to it**. Your goal is to recover the meaning for each symbol. Please write down the English sentence that each logic symbol represents below.

English		Propositional logic
I will not buy a 24 kg kettlebell.		$\neg P \vee S$
I will buy an 8 kg kettlebell or a 24 kg kettlebell, but not both.		$(Q \vee R) \wedge (\neg Q \vee \neg R)$
If I play more go, then I will get better at it.		$(\neg Q \wedge \neg R) \vee S$
If I buy an 8 kg kettlebell or a 24 kg kettlebell, then I will get better at go.		$\neg R$

(i) [1 pt] P:

(ii) [1 pt] Q:

(iii) [1 pt] R:

(iv) [1 pt] S:

(b) [8 pts] Winston, Xuan, Yvonne, and Zeke just had an 80's exercise party, where they **did some exercise, wore sweatbands, and listened and danced to some music.** They have the following predicates in their vocabulary:

- $Suggested(p, e)$: Person p suggested exercise e.
- $Tried(p, e)$: Person p tried exercise e.
- $Danced(p, s)$: Person p danced to song s.
- $Likes(p, s)$: Person p likes song s.

They also abbreviate their names by the first letter.

After the party, they each tried to write down some English sentences in first-order logic, but they didn't always succeed. Below, you have the English sentences as well as the attendees' attempts to write them in first-order logic.

For each first-order logic sentence, choose whether it is invalid, valid but not equivalent to the English sentence, or equivalent to the English sentence.

(i) [4 pts] Xuan tried every exercise that Zeke suggested.

$\exists e\ Suggested(Z, e) \wedge (\forall e' \vee Tried(X, e'))$	○ Invalid	○ Not equiv.	○ Equiv.
$\forall e, e'\ \neg(e = e') \vee (\neg Suggested(Z, e')) \vee Tried(X, e)$	○ Invalid	○ Not equiv.	○ Equiv.
$\forall e\ Suggested(Z, e) \Rightarrow Tried(X, e)$	○ Invalid	○ Not equiv.	○ Equiv.
$(\exists e\ Suggested(Z, e)) \wedge (\forall e\ Tried(X, e))$	○ Invalid	○ Not equiv.	○ Equiv.

(ii) [4 pts] Every song that anybody liked was danced to by someone who tried an exercise.

$\neg\exists s, p\ (Likes(p, s) \wedge Danced(p, s)) \vee (\forall e\ Tried(p, e))$	○ Invalid	○ Not equiv.	○ Equiv.
$\forall s\ \exists p, e\ Likes(p, s) \Rightarrow\ Danced(p, s) \wedge Tried(p, e)$	○ Invalid	○ Not equiv.	○ Equiv.
$\forall s\ (\exists p\ Likes(p, s)) \Rightarrow (\exists e, p'\ Danced(p', s) \wedge Tried(p', e))$	○ Invalid	○ Not equiv.	○ Equiv.
$\forall s, p, p'\ Likes(p \wedge p', s) \Rightarrow ((Danced(p, s) \wedge \exists e\ Tried(p, e)) \wedge (Danced(p', s) \wedge \exists e\ Tried(p', e))$	○ Invalid	○ Not equiv.	○ Equiv.

(c) [8 pts] You have the following sentence, and you want to know whether or not it is satisfiable:

$$S:\quad (A \vee \neg B \vee D) \wedge (\neg A \vee B \vee E) \wedge (A \vee \neg C \vee \neg F) \wedge (B \vee C \vee F) \wedge (\neg C \vee \neg D \vee \neg E) \wedge (D \vee E \vee F).$$

First, you must choose an algorithm to use.

(i) [1 pt] Which algorithm is best suited for your task?
◯ DPLL ◯ Propositionalization ◯ Forward chaining

Now that you've picked your SAT-solving algorithm, you have to run it. Luckily, you know how to do this, because it's essentially a variant of another algorithm you've studied earlier in the course.

(ii) [1 pt] Which other algorithm that we've studied is your SAT-solving algorithm a variant of?
◯ Alpha-beta pruning ◯ Simulated annealing ◯ Depth-first search
◯ Breadth-first search ◯ A* search

During the execution of the algorithm, you have set variable A to be False and variable B to be True. Answer the following questions:

(iii) [1 pt] Which variable is now a pure literal? ◯ C ◯ D ◯ E ◯ F

(iv) [1 pt] What value do you set that variable to in this case? ◯ True ◯ False

(v) [1 pt] Which of the original clauses is now a unit clause? (Note that it was already a unit clause before you dealt with the pure literal)
◯ $A \vee \neg B \vee D$
◯ $\neg A \vee B \vee E$
◯ $A \vee \neg C \vee \neg F$
◯ $B \vee C \vee F$
◯ $\neg C \vee \neg D \vee \neg E$
◯ $D \vee E \vee F$

(vi) [1 pt] What do we do once we find this unit clause?
◯ Terminate, since sentence is unsatisfiable
◯ Terminate, return satisfying assignment for the sentence
◯ Satisfy the clause by setting the remaining variable **in the clause** to True
◯ Satisfy the clause by setting the remaining variable **in the clause** to False

(vii) [1 pt] Is there an assignment to the variables that satisfies S where A is set to False and B is set to True?
◯ Yes ◯ No

(viii) [1 pt] Reflecting on your journey, you notice that resolving the pure literal didn't add any more unit clauses. Can resolving a pure literal ever add a new unit clause? ◯ Yes ◯ No

Q5. [16 pts] Ball Games

Alice is playing a ball game with Bob. There are 3 boxes in front of them, each containing 3 balls, and each ball has a score. Alice first selects a box from the 3 boxes, and then Bob takes a ball from the box selected by Alice. In Figure 1, nodes B_1, B_2 or B_3 are boxes, and nodes labeled C through K represent individual balls and their scores. Unless otherwise specified, Alice's objective is to maximize the score of the ball that is eventually chosen, and Bob's objective is to minimize Alice's score. Assume both players always act optimally for their goals.

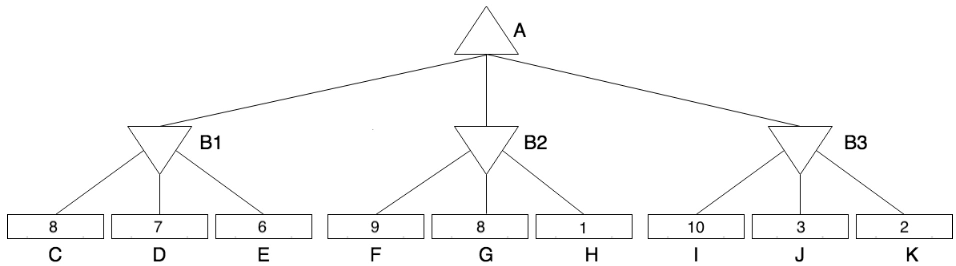


Figure 1

(a) **(i)** [2 pts] In the blanks below, fill in the **labels** (not the numerical values) of the balls selected for nodes A, B_1, B_2 and B_3. For example, if Alice's optimal move is to select the box B_1, and Bob selects the ball C, then A=C, B_1=C.

A= ☐, B_1= ☐, B_2= ☐, B_3= ☐.

(ii) [1 pt] Let's apply alpha-beta pruning for the tree. Please choose all the **child nodes** of the branches that could be pruned in the search tree. ☐ B_1 ☐ B_2 ☐ B_3 ☐ C ☐ D ☐ E ☐ F ☐ G ☐ H ☐ I ☐ J ☐ K

(iii) [3 pts] Assume that you can re-allocate all the balls to different boxes as long as in the end, each box has exactly 3 balls in it. Could you find a way to re-allocate the balls so that when we apply alpha-beta pruning, the right branch of B_2 as well as the middle and the right branch of B_3 will be pruned?

Please list all balls in the order of their new positions in your solution. For example, if your solution is not to move any ball, then your answer is "C,D,E,F,G,H,I,J,K". When multiple solutions exist, break ties alphabetically. For example, if there are two solutions starting with "C,D,..." and "D,C,...", then your answer should be "C,D,...". If there is no solution, please write down "None".

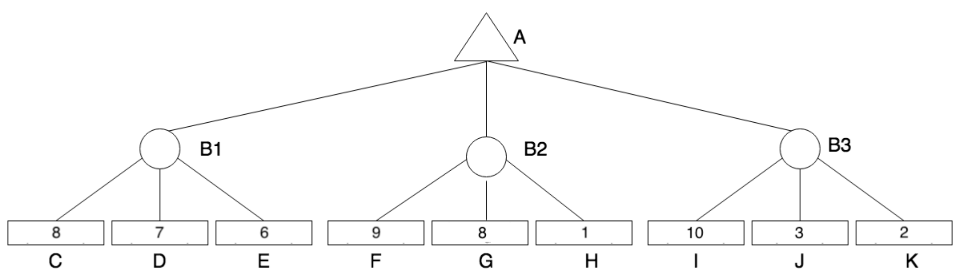


Figure 2

(b) In this part, Alice still chooses a box of balls, but now Bob only gets to choose a ball uniformly at random from that box, as shown in Figure 2. After Alice chooses a box and Bob randomly picks out a ball out of the box, Alice now has two options: (1) keep that score, and the game ends normally, or (2) abandon that score and start over from choosing a box (she may choose the same box as the one she previously chose). The latter score will always override the previous score, even if it is lower.

(i) [2 pts] If Alice first chooses box B_3 and get a score of 3, should Alice keep that score? If she choose to abandon the score, what is the optimal action for the next choice? If there are multiple optimal actions with the same expected score, choose all of them.

☐ Alice should keep the score.

☐ Alice should abandon the score and choose B_1.

☐ Alice should abandon the score and choose B_2.
☐ Alice should abandon the score and choose B_3.

(ii) [2 pts] What is Alice's optimal action for the first choice she makes, given that she is aware of the potential second chance? If there are multiple optimal actions with the same expected score, choose all of them.

☐ B_1 ☐ B_2 ☐ B_3

(iii) [5 pts] Suppose now that Alice has the chance to get the previous score overridden twice (if she is not satisfied with her second score, she start over a third time). Given that she is aware of this rule, what is her optimal policy? A description of the optimal policy is given below. Fill in the blanks. Ties are broken with the order B_1, B_2, B_3.

First, choose box ○ B_1 ○ B_2 ○ B_3.

If the score $\geq$ ______, then keep the score and end the game.

Otherwise, abandon the previous score and choose box ○ B_1 ○ B_2 ○ B_3.

If the score $\geq$ ______, then keep the score and end the game.

Otherwise, abandon the previous score again and choose box ○ B_1 ○ B_2 ○ B_3 at the final chance.

(iv) [1 pt] What is the expected score Alice will get if she follows the sequence of choices described in the previous problem? ______

SID: ____________________________

Q6. [?? pts] Bayes Nets and Inference

"The greatest progress that the human race has made lies in learning how to make correct inferences",
Friedrich Nietzsche.

And after such a pompous quote let's dig into our problem.

(a) **(i)** [4 pts] Assume $P(C|B,A) = P(C|B)$ for random variables A, B and C. Which of the following expressions hold?

☐ $P(A|B,C) = P(A|B)$

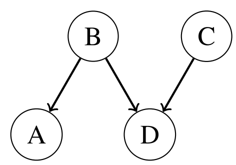
☐ $P(A,C|B) = P(A|B)P(C|B)$

☐ $\frac{P(B,C|A)P(A)}{P(B)} = \frac{\sum_b P(A,b,C)}{P(B)}$

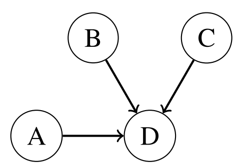
☐ $P(B|A,C) = \frac{P(B)P(A|B)P(C|B)}{P(A)P(A|C)}$

(ii) [4 pts] For which of the following Bayes Nets is $P(A)P(C|A) = P(C) * \sum_c P(A,c|B)$ guaranteed to hold?

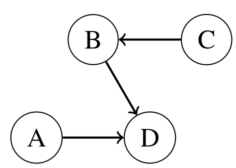
☐



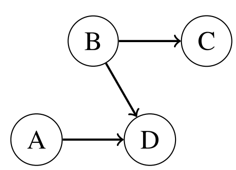
☐



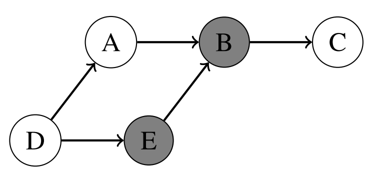
☐



☐



(iii) [2 pts]



Now consider the Bayes' net depicted above. We want to infer exactly $P(A|b,e)$, where A is the query variable, B, E the evidence variables, C, D the hidden variables and α a normalizing constant which value you can choose. Which of the following expressions is the correct value of the query $P(A|b,e)$?

○ $\alpha \sum_c \sum_d P(c|b)P(b|A)P(b|e)P(A|d)P(e|d)P(d)$

○ $\alpha \sum_c \sum_d P(c|b)P(b|A,e)P(A|d)P(e|d)$

○ $\alpha \sum_c \sum_d P(c|b)P(b|A,e)P(A|d)P(e|d)P(d)$

○ $\alpha \sum_b \sum_e P(c|b)P(A|b,e)P(A|d)P(e|d)P(d)$

(iv) [2 pts] For the same network as in (iii) which of the following expressions is the most computationally efficient expression for the query $P(A|b,e)$?

- ◯ $\alpha P(b|A,e) \sum_d \sum_c P(A|d)P(e|d)P(c|b)P(d)$
- ◯ $\alpha P(b|A,e)P(A|d)P(e|d) \sum_c P(c|b) \sum_d P(d)$
- ◯ $\alpha P(b|A,e) \sum_d P(A|d)P(e|d)P(d) \sum_c P(c|b)$
- ◯ $\alpha P(b|A,e) \sum_c P(A|d)P(e|d) \sum_d P(c|b)P(d)$

(v) [2 pts] We call a variable irrelevant if its value has no effect on the value of the query we want to compute. For the same network as in (iii) which of the following variables, if any, is irrelevant to the query $P(A|b,e)$?

- ◯ A
- ◯ C
- ◯ E
- ◯ B
- ◯ D
- ◯ None

(vi) [2 pts] If you were to perform variable elimination on the the Bayes Net depicted above to estimate $P(A|b,e)$, what would the size of the largest generated probability table be under the most efficient elimination ordering? (We define "most efficient elimination ordering" as the order that yields the minimally-sized largest probability table.) Assume that all variables A, B, C, D, E are binary.

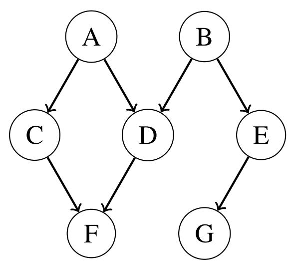
- ◯ 2
- ◯ 8
- ◯ 4
- ◯ 16

(vii) [1 pt] This question is unrelated to the previous questions. Let A, B, C, D, E , F and G be binary random variables. What is the size of the probability table generated by the pointwise product of the factors $P(A,B,C) \times P(A,F,G)$?

- ◯ 2^3
- ◯ 2^5
- ◯ 2^4
- ◯ 2^6

(b) In large networks exact inference can become intractable. For this reason we frequently employ sampling methods for inference.

(i) [1 pt]



Consider the seven-node Bayesian network above. Which set of variables comprise the Markov blanket of variable D?

- ◯ F
- ◯ A, B, F
- ◯ A, B
- ◯ A, B, F, C

(ii) [**??** pts] Assume for convenience that the variables in the above network are binary and that we have observed $A = 1$, $B = 0$, $F = 0$. We will be using Gibbs sampling to estimate the probability $P(D|A = 1, B = 0, F = 0)$. The initial values of the non-evidence variables are $C = D = E = G = 1$. We give you four sets of the first three updates performed by Gibbs sampling. The notation $X, P(X|Y), X = x$ means that at this update step, variable X was chosen, it was sampled from distribution $P(X|Y)$, and the realized value of the variable was x. Which of these sets of updates could NOT have resulted from a correct implementation of Gibbs sampling?

- ☐ E, $P(E|B = 0, G = 1)$, $E = 1$
E, $P(E|B = 0, G = 1)$, $E = 0$
E, $P(E|B = 0, G = 1)$, $E = 0$
- ☐ C, $P(C|A = 1, F = 0)$, $C = 1$
G, $P(G|E = 1)$, $G = 1$
C, $P(C|A = 1, F = 0)$, $C = 0$
- ☐ E, $P(E|B = 0, G = 1)$, $E = 1$
G, $P(G|E = 1)$, $G = 0$
E, $P(E|B = 0, G = 1)$, $E = 0$
- ☐ C, $P(C|A = 1, D = 1, F = 0)$, $C = 1$
G, $P(G|E = 1)$, $G = 0$
E, $P(E|B = 0, G = 0)$, $E = 1$