MDP question.

Consider an MDP (S, A, T, gamma, R). When your transition s,a,s' would standardly give you a reward R(s,a,s'), you don't get to collect this reward right away. Rather it can be thought of as a certificate, and to cash it in for real reward you need to visit a special state s*. When you land in the special state s*, you automatically cash in all your reward certificates in exchange for a reward corresponding to the total value of the reward certificates.

Formulate an MDP such that the optimal policy in this new MDP maximizes the expected sum of (discounted) cashed in rewards.

Application: this could be relevant when, let's say, you are collecting resources that need to be brought back to a base to be valuable and if you get intercepted along the way back to the base, no value was obtained from finding these resources.

Let the new MDP be $(\bar{S}, \bar{A}, \bar{T}, \bar{\gamma}, \bar{R})$

$\bar{A} = A$

$\bar{S} = S \times \mathbb{R}$ here the additional state variable corresponds to total of reward certificates the agent has

$\bar{\gamma} = \gamma$

$$\bar{T}(\bar{s}, \bar{a}, \bar{s}') = \bar{T}((s, r), a, (s', r'))$$

$$= \begin{cases} T(s,a,s') & \text{if } r' = r + R(s,a,s') \text{ and } s' \neq s^* \\ T(s,a,s^*) & \text{if } r' = 0 \text{ and } s' = s^* \\ 0 & \text{otherwise} \end{cases}$$

NAME:_____ SID#:_____ Login:_____ Sec:_____

CS 188 Fall 2006 Introduction to Artificial Intelligence Final Exam

You have 180 minutes. The exam is closed-book (except for your 3 pages of notes), no electronics (other than basic calculators). 160 points total. Don't panic!

Mark your answers ON THE EXAM ITSELF. Write your name, SID, login, and section number at the top of each page.

If you are not sure of your answer you may wish to provide a *brief* explanation. All short answer sections can be successfully answered in a few sentences *at most.*

For official use only

Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Q. 7	Q. 8	Total
/14	/24	/16	/16	/30	/20	/22	/18	/160

1. (14 points.) True/False

Each problem is worth 2 points. Incorrect answers are worth 0 points. Skipped questions are worth 1 point.

(a) *True*/*False*: All fringe-based graph search strategies are complete for finite state spaces.

(b) *True*/*False*: If a tree search method is optimal, then the corresponding graph search is also optimal.

(c) *True*/*False*: In establishing arc consistency, some arcs may have to be processed (made consistent) multiple times.

(d) *True*/*False*: Reflex agents can be rational.

(e) *True*/*False*: Given its parents, a variable X in a Bayes' net is conditionally independent of all variables Y which are not descendents of X (i.e. not X's children, not its children's children, etc.).

(f) *True*/*False*: A reinforcement learning agent can learn an optimal policy even if it executes only random actions.

(g) *True*/*False*: A reinforcement learning agent's behavior can be altered simply by altering the reward function.

MDP question.

Consider an MDP (S, A, T, gamma, R). When your transition s,a,s' would standardly give you a reward R(s,a,s'), you don't get to collect this reward right away. Rather it can be thought of as a certificate, and to cash it in for real reward you need to visit a special state s*. When you visit the special state s* at time t, you can cash in all your reward certificates in exchange for a reward corresponding to the total value of the reward certificates, multiplied by gamma^t.

Formulate an MDP (S', A', T', gamma', R') such that the optimal policy in this new MDP maximizes the expected sum of (discounted) cashed in rewards.

Application: this could be relevant when, let's say, you are collecting resources that need to be brought back to a base to be valuable and if you get intercepted along the way back to the base, no value was obtained from finding these resources.

2. (24 points.) Search and Bayes' Nets

Consider the problem of finding the *most likely explanation* in a general Bayes' net. The input is a network G in which some variables $X_{e_1} \ldots X_{e_k}$ are observed, and the output is an assignment to all the variables $X_1 \ldots X_n$, consistent with the observations, which has maximum probability. You will formulate this problem as a state space search problem. Assume that the network is constructed such that for any variable X_i, its parents Parents(X_i) are variables X_j for $j < i$.

States: each partial assignment to a prefix of the variables, of the form $\{X_1 = x_1, X_2 = x_2, \ldots X_k = x_k\}$
Initial state: the empty assignment {}
Successor function: ??
Goal test: the assignment is complete (i.e. assigns all variables)
Step cost: ??

(a) **(3 pts)** Give an expression for the size of the state space if each variable X_i has D_i elements in its domain.

Let $E = \{e_1, \ldots, e_k\}$

$$\prod_{\substack{i=1 \\ i \notin E}}^{n} (D_i + 1)$$

$+1$ corresponds to variable X_i not having been assigned yet

(b) **(3 pts)** What is the successor function for this search problem?

Make assignment to 1 variable

(c) **(4 pts)** What is the cost function for this search problem? *Hint: Recall that* $\log ab = \log a + \log b$ *and that search minimizes total cost.*

For a complete assignment we have

$$\text{cost} = -\log P(\text{complete assignment}) = -\sum_{i=1}^{n} \log P(x_i \mid Pa(X_i))$$

as we only consider partial assignments from 1 through k (as opposed to arbitrary subsets), we can associate (and easily compute) a cost $-\log P(x_i \mid Pa(x_i))$ for each successor/action.

(d) **(4 pts)** Give two reasons why BFS would be a poor choice for solving this problem.

(1) ~~Logical Search~~ BFS is not trying to optimize a cost

(2) Goal states are only reached at the bottom of the search tree → no benefit from BFS (similar to CSP solving)

(e) **(6 pts)** Give a non-trivial admissible heuristic for this problem. You heuristic should be efficient to compute. Justify the admissibility of your heuristic briefly.

assume ~~assign~~ partial assignment $x_1, x_2, \ldots, x_k$

$$\rightarrow h(x_1, \ldots, x_k) = \sum_{i=k+1}^{n} \min_{x_i, Pa(x_i) \text{ such that } x_i \text{ and } Pa(x_i) \text{ consistent w/ partial assignment } x_1 \ldots x_k} -\log P(x_i \mid Pa(x_i))$$

(f) **(4 pts)** Briefly describe how we might use local search to solve this problem.

Instantiate all variables.

For all non evidence variables, cycle through them and ~~to~~ set to best value assuming all others are fixed. (= greedy local search)

ϵ-Admissible Heuristics: Suppose that we have a heuristic function which is not admissible, but *ϵ-admissible*, meaning for some known $\epsilon > 0$,

$$h(n) \leq h^*(n) + \epsilon \qquad \text{for all nodes } n$$

where $h^*(n)$ is the optimal completion cost. In other words, h is never more than ϵ from being optimal.

(e) (1 point) Is using A* with an ϵ-admissible heuristic complete? Briefly justify.

Yes. No successors are overlooked until the goal state has been expanded.

(f) (2 points) Assuming we utilize an ϵ admissible heuristic in standard A* search, how much worse than the optimal solution can we get? I.e., if c^* is the optimal cost for a search problem, what is the worst cost solution an ϵ admissible heuristic would yield? Justify your answer.

$C^* + \epsilon$ [I assume tree search as question does not say tree or graph]

When the A* search with ϵ-admissible heuristic finishes with path of cost $\bar{c}$ there would still be partial plans on fringe with $g+h = \bar{c}$. Such partial plan could, potentially be extended in a path of cost $\bar{c} - \epsilon$.

(g) (2 points) Suggest a modification to the A* algorithm which will be guaranteed to yield an optimal solution using an ϵ-admissible heuristic with fixed, known ϵ. Justify your answer.

(1) Keep running the search until all partial plans remaining on fringe have cost at least $\bar{c} + \epsilon$

(2) use $\tilde{h}(a) = \max\{0, h(a) - \epsilon\}$

(d) Describe an effective heuristic for deciding which variable to assign next in a backtracking CSP solver.

Minimum remaining values

max degree

(e) Describe an effective heuristic for deciding which value of a variable to assign next in a backtracking CSP solver.

least constraining value

(f) Why does money not generally work well as a utility scale?

End of Exam

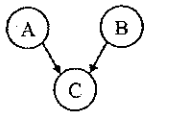
3. (19 points.) Bayes' Nets

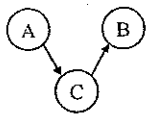
Consider the following pairs of Bayes' nets. If the two networks have identical conditional independences, write *same*, along with writing one of their shared independence (or *none* if they assert none). If the two networds have different conditional independences, write *different*, along with writing an independence that one has but not the other. For example, in the following case you would answer as shown:

different, right has $A \perp\!\!\!\perp B|\{\}$.

L R

(a) (1 pt)



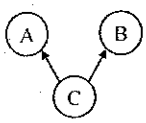


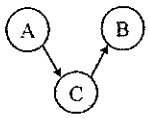
different

L R

$A \perp\!\!\!\perp B$ $A \not\perp\!\!\!\perp B$

(b) (1 pt)

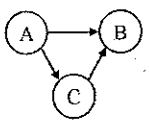


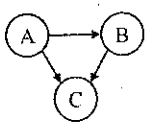


same

$A \perp\!\!\!\perp B|C$

(c) (1 pt)

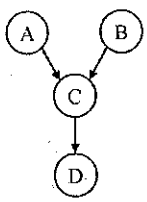


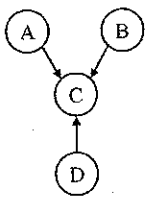


same

none

(d) (1 pt)

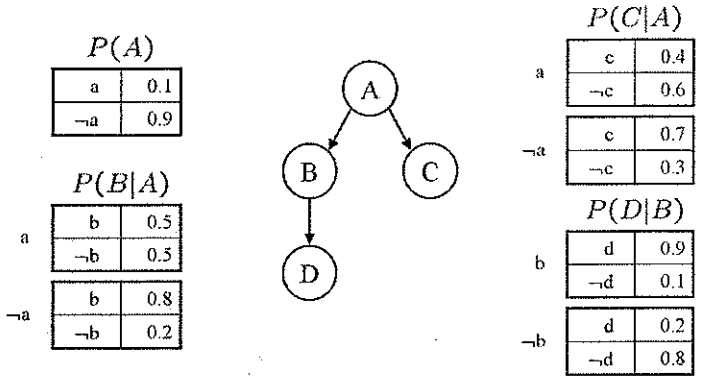




different

$A \perp\!\!\!\perp D|C$ $A \not\perp\!\!\!\perp D|C$

The next parts involve computing various quantities in the network below. These questions are designed so that they can be answered with a minimum of computation. If you find yourself doing copious amount of computation for each part, step back and consider whether there is simpler way to deduce the answer.



(e) (2 pts) $P(a, \neg b, c, \neg d)$ $P(a)\,P(\neg b|a)\,P(c|a)\,P(\neg d|b)$

$0.1 * 0.5 * 0.4 * 0.1$

(f) (2 pts) $P(b)$ $\sum_a P(a)\,P(b|a) = 0.1 * 0.5 + 0.9 * 0.9$

(g) (2 pts) $P(a|b)$ $\frac{P(a,b)}{P(b)} = \frac{0.1 * 0.5}{0.1 * 0.5 + 0.9 * 0.8}$

(h) (2 pts) $P(d|a)$ $\sum_b P(d|b)\,P(b|a) = 0.9 * 0.5 + 0.2 * 0.5$

(i) (2 pts) $P(d|a, c)$ $= P(d|a)$

Consider computing the following quantities in the above network using various methods:

(i) $P(A|b, c, d)$ (ii) $P(C|d)$ (iii) $P(D|a)$ (iv) $P(D)$

(j) (2 pts) Which query is least expensive using inference by enumeration?

$P(A|b, c, d)$ no need to sum out

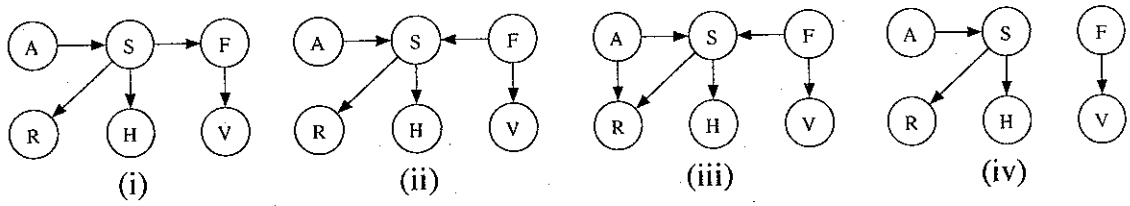
(k) (2 pts) Which query is most improved by using likelihood weighting instead of rejection sampling (in terms of number of samples required)?

$P(A|b, c, d)$ → 3 chances for rejection

4. (15 points.) Bayes Nets: Snuffles

Assume there are two types of conditions: (S)inus congestion and (F)lu. Sinus congestion is is caused by (A)llergy or the flu.

There are three observed symptoms for these conditions: (H)eadache, (R)unny nose, and fe(V)er. Runny nose and headaches are directly caused by sinus congestion (only), while fever comes from having the flu (only). For example, allergies only cause runny noses indirectly. Assume each variable is boolean.



(a) (2 points) Consider the four Bayes Nets shown. Circle the one which models the domain (as described above) best.

(ii)

(b) (3 points) For each network, if it models the domain exactly as above, write *correct*. If it has too many conditional independence properties, write *extra independence* and state one that it has but should not have. If it has too few conditional independence properties, write *missing independence* and state one that it should have but does not have.

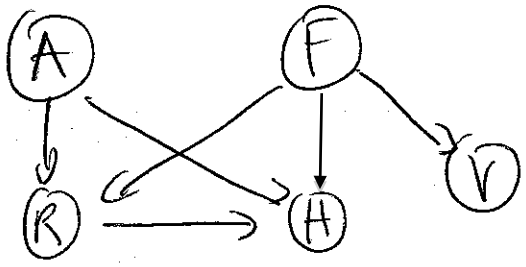
(i) missing $A \perp F$, extra $A \not\perp F \mid S$

(ii) correct

(iii) missing $A \perp R \mid S$

(iv) extra $F \not\perp S$

(c) (3 points) Assume we wanted to remove the Sinus congestion (S) node. Draw the minimal Bayes Net over the remaining variables which can encode the original model's marginal distribution over the remaining variables.



(d) (2 points) In the original network you chose, which query is more efficient to compute using variable elimination: $P(F|r, v, h, a, s)$ or $P(F)$? Briefly justify.

If strictly following variable elimination procedure, then $P(F|r, v, h, a, s)$

Of course, for this particular query reading $P(F)$ out from F's CPT would make most sense

Assume the following samples were drawn from prior sampling:

$a, s, r, \neg h, \neg f, \neg v$
$a, s, \neg r, h, f, \neg v$
$a, \neg s, \neg r, \neg h, \neg f, \neg v$
$a, \neg s, \neg r, h, f, \neg v$
$a, s, \neg r, h, \neg f, \neg v$

(e) (1 point) Give the sample estimate of $P(f)$ or state why it cannot be computed.

$\frac{2}{5}$

(f) (1 point) Give the sample estimate of $P(f|h)$ or state why it cannot be computed.

$\frac{2}{3}$

(g) (1 point) Give the sample estimate of $P(f|v)$ or state why it cannot be computed.

$\frac{0}{0}$

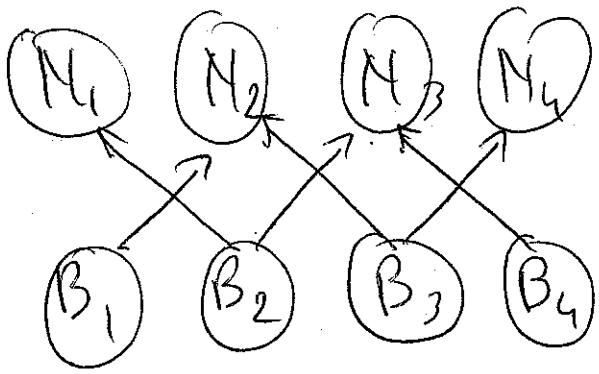
(h) (2 points) For rejection sampling in general (not necessarily on these samples), which query will require more samples to compute to a certain degree of accuracy, $P(f|h)$ or $P(f|h, a)$? Justify your answer in general terms.

$P(f|h)$ cheaper than $P(f|h,a)$ as fewer samples would be rejected

5. (30 points.) Bayes' Nets

In the game of Minesweeper, there are bombs placed on a grid; you do not know where or how many. Assume that each square (i, j) independently has a bomb ($B_{i,j} = true$) with probability b. What you can observe for a given square is a reading $N_{i,j}$ of the number of bombs in adjacent squares (i.e. the eight closest squares *not including the square itself*). The variables $N_{i,j}$ can therefore take the values 0 through 8, plus a special value *bomb* if the square itself has a bomb (at which point the adjacent bomb count has no effect on the reading). If a square has less than 8 neighbors, such as on the boundaries, its N has an appropriately limited domain. In classic Minesweeper, you lose if you try to reveal a square with a bomb; you will ignore that complication in this problem.

(a) **(3 pts)** Draw a Bayes' net for a one-dimensional 4x1 Minesweeper grid, showing all eight variables ($B_1 \dots B_4$ and $N_1 \dots N_4$). Show the minimal set of arcs needed to correctly model the domain above.



(b) **(3 pts)** Fully specify the CPTs for B_1 and N_1, assuming that there is no noise in the readings (i.e. that the number of adjacent bombs (or *bomb*) is reported exactly, deterministically). Your answers may use the bomb rate b if needed.

P(B1)

B_1	$P(B_1)$
true	b
false	$1-b$

N_1	B_2	P
1	true	1
0	false	1
other		0

(c) **(3 pts)** What are the posterior probabilities of bombs in each of the four squares, given no information?

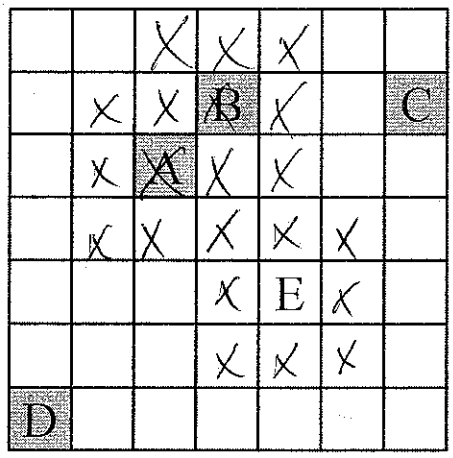
$\forall i \in \{1,2,3,4\} \quad P(B_i = \text{true}) = \frac{1}{6}$

(d) **(4 pts)** If we observe $N_2 = 1$, what are the posterior probabilities of bombs in each square?

$P(B_1 = \text{true}) = P(B_3 = \text{true}) = \frac{1}{2}$

$P(B_2 = \text{true}) = P(B_4 = \text{true}) = 0$

(e) **(4 pts)** On the following two-dimensional grid, assume we know the value of N_A, N_B, N_C, and N_D, and we are about to observe N_E. Shade in the squares whose posterior bomb probabilities can change as a result of this new observation.



5. (12 points.) HMMs: Forward and Backward Algorithms

Recall that HMMs model hidden variables $X_{1:T} = X_1, \ldots X_T$ and evidence variables $E_{1:T} = E_1 \ldots E_T$. The forward algorithm incrementally computes $P(X_t, e_{1:t})$ for increasing t for the purpose of calculating $P(X_t|e_{1:t})$, the posterior belief over X_t given current evidence e_t and past evidence $e_{1:t-1}$ in an HMM. A more general query is to condition on *all* evidence, past, present, and future: $P(X_t|e_{1:N})$. In this problem, you will work out a method of doing so.

(a) (4 points) Use the laws of probability and the conditional independence properties of an HMM to give an expression for $P(e_{t+1:N}|x_t)$ in terms of $P(e_{t+2:N}|x_{t+1})$ and the basic HMM quantities ($P(X|X')$ and $P(E|X)$). You do not need to worry about the base case.

$$P(e_{t+1:N}|x_t) = \sum_{x_{t+1}} P(e_{t+1:N}, x_{t+1}|x_t) = \sum_{x_{t+1}} P(e_{t+1:N}|x_t, x_{t+1}) \cdot P(x_{t+1}|x_t)$$

$$= \sum_{x_{t+1}} P(e_{t+1:N}|x_{t+1}) \; P(x_{t+1}|x_t)$$

This computation is called the backward algorithm.

(b) (3 points) Give an expression for the posterior distribution at a single time step, $P(x_t|e_{1:N})$, in terms of basic HMM quantities and / or quantities computed by the forward and backward algorithms. *Hint: use the chain rule along with the conditional independence properties of HMMs.*

$$P(x_t|e_{1:N}) \propto P(x_t, e_{1:N}) = P(e_{1:t}) \cdot P(x_t|e_{1:t}) \cdot P(e_{t+1:N}|x_t, e_{1:t})$$

$$\propto P(x_t|e_{1:t}) \cdot P(e_{t+1:N}|x_t)$$

$$\rightarrow P(x_t|e_{1:N}) = \frac{P(x_t|e_{1:t})\, P(e_{t+1:N}|x_t)}{\sum_{x_t} P(x_t|e_{1:t})\, P(e_{t+1:N}|x_t)}$$

(c) (2 points) Give an expression for the posterior distribution over two time steps, $P(x_t, x_{t-1}|e_{1:N})$, in terms of basic HMM quantities and / or quantities computed by the forward and backward algorithms.

$$P(x_t, x_{t-1}|e_{1:N}) \propto P(e_{1:t-1}) \cdot P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1}, e_{1:t-1}) \cdot P(e_t|x_{t-1}, x_t, e_{1:t-1})$$

$$\cdot P(e_{t+1:N}|x_{t-1}, x_t, e_{1:t-1})$$

$$\propto P(x_{t-1}|e_{1:t-1})\, P(x_t|x_{t-1})\, P(e_t|x_t)\, P(e_{t+1:N}|x_t)$$

In a second-order HMM, the transition function depends on the past two states: $P(x_t|x_{1:t-1}) = P(x_t|x_{t-1}, x_{t-2})$. Emissions still depend only on the current state.

(d) (3 points) Give the second-order generalization of the forward recurrence. Again, you may disregard the base case. *Hint: you should think about both the left and right hand sides.*

$$P(x_t, x_{t-1}, e_{1:t}) = \sum_{x_{t-2}} P(x_{t-1}, x_{t-2}, e_{1:t-1}) \cdot P(x_t|x_{t-1}, x_{t-2}) \cdot P(e_t|x_t)$$