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.

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 $X_{e_1} \dots X_{e_k}$ are observed, and the output is an assignment to all the variables $X_1 \dots X_n$, 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 $\text{Parents}(X_i)$ are variables X_j for $j < i$.

States: each partial assignment to a prefix of the variables, of the form $\{X_1 = x_1, X_2 = x_2, \dots X_k = x_k\}$
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.

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

(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.*

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

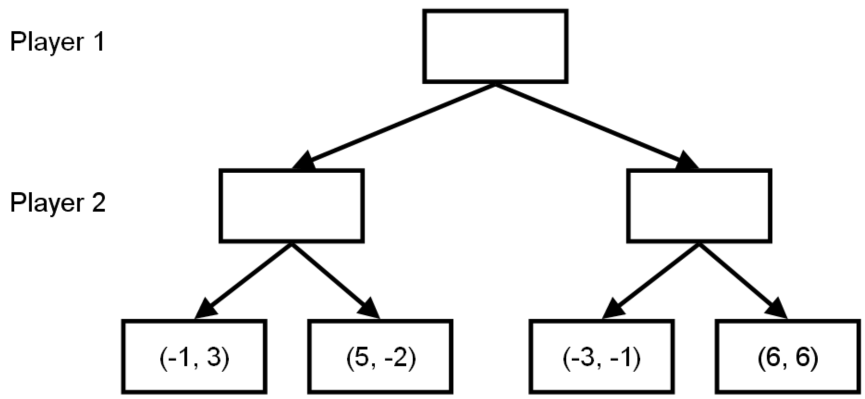
(e) **(6 pts)** Give a non-trivial admissible heuristic for this problem. You heuristic should be efficient to compute. Justify the admissibility of your heuristic briefly.

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

3. (16 points.) Game Trees

In a two-player *non-zero-sum* game, players 1 and 2 alternate moves, just as in a minimax game. However, terminal states are not labeled with a single value $V(s)$, but rather with a pair of values $V(s) = (V_1(s), V_2(s))$ representing the utility of that terminal outcome to players 1 and 2, respectively. Each player tries to maximize their own utility, under the assumption that the other player is playing optimally (again, similar to minimax).

(a) **(4 pts)** Label each node in the following search tree with its value pair. 1-nodes represent player 1's move, while 2-nodes represent player 2's move.



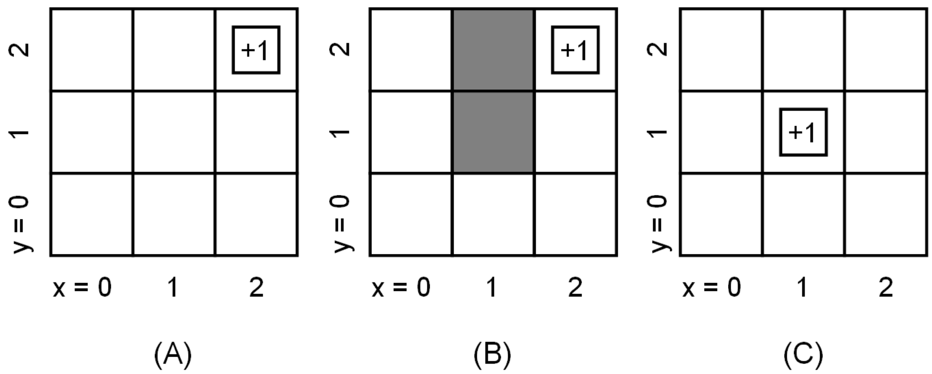
(b) **(4 pts)** Describe formally how to compute the value pair $V(s) = (V_1(s), V_2(s))$ of a node at state s under the control of player 1 (the analog of a max node).

(c) **(4 pts)** Is it possible to prune the search in a manner similar to α-β pruning? Either describe a pruning algorithm, or describe why such pruning is not possible.

(d) **(4 pts)** Would the knowledge that the game is nearly zero-sum, such as knowing that $|V_1(s) + V_2(s)| \leq k$ for all terminal states s allow you to improve your pruning algorithm or enable pruning (depending on your answer to (c))? Describe why or why not. Do not write more than a few sentences at most!

4. (16 points.) Reinforcement Learning

For the following gridworld problems, the agent can take the actions *N, S, E, W*, which move the agent one square in the respective directions. There is no noise, so these actions always take the agent in the direction attempted, unless that direction would lead off the grid or into a blocked (grey) square, in which case the action does nothing. The boxed +1 squares also permit the action X which causes the agent to exits the grid and enter the terminal state. The reward for all transitions are zero, except the exit transition, which has reward +1. Assume a discount of 0.5.



(a) **(4 pts)** Fill in the optimal values for grid (A) (*hint: this should require very little calculation*).

(b) **(3 pts)** Specify the optimal policy for grid (B) by placing an arrow in each empty square.

Imagine we have a set of real-valued features $f_i(s)$ for each non-terminal state $s = (x, y)$, and we wish to approximate the optimal utility values $V^*(s)$ by $V(s) = \sum_i w_i \cdot f_i(s)$ (linear feature-based approximation).

(c) **(3 pts)** If our features are $f_1(x, y) = x$ and $f_2(x, y) = y$, give values of w_1 and w_2 for which a one-step look-ahead policy extracted from V will be optimal in grid (A).

(d) **(2 pts)** Can we represent the actual optimal values V^* for grid (A) using these two features? Why or why not?

(e) **(4 pts)** For each of the feature sets listed below, state which (if any) of the grid MDPs above can be 'solved', in the sense that we can express some (possibly non-optimal) values which produce optimal one-step look-ahead policies.

i. $f_1(x, y) = x$ and $f_2(x, y) = y$.

ii. For each (i, j), a feature $f_{i,j}(x, y) = 1$ if $(x, y) = (i, j)$, 0 otherwise.

iii. $f_1(x, y) = (x-1)^2$, $f_2(x, y) = (y-1)^2$, and $f_3(x, y) = 1$.

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.

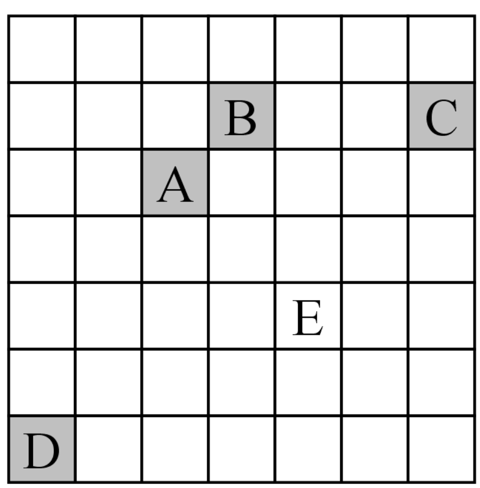
(a) **(3 pts)** Draw a Bayes' net for a one-dimensional 4x1 Minesweeper grid, showing all eight variables ($B_1 \ldots B_4$ and $N_1 \ldots N_4$). Show the minimal set of arcs needed to correctly model the domain above.

(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.

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

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

(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.



(f) **(3 pts)** On a 2x1 grid, imagine that you must take an action by declaring which squares have bombs and which do not, so there are four possible actions on the 2x1 grid (again, note that there is no fixed number of bombs, unlike in your project or in classic Minesweeper). The utility of correctly declaring a bomb is +1, the utility of correctly declaring a clear square is +1, the utility of overlooking a bomb is -10 and the utility of declaring a bomb where there is none is -1. If the initial probability of a bomb is 0.5, what is the MEU action, and what is its EU?

(g) **(6 pts)** On the same 2x1 grid, what is the value of information about N_1?

(h) **(4 pts)** How would you modify the network in (a) if you knew that there were exactly two bombs? Draw a new network below and briefly describe/justify any new nodes you introduce.

6. (20 points.) Classification

Consider a traffic-monitoring agent trying to decide whether the stoplights at an intersection are working or not ($W = w$ or $\neg w$. The agent observes two variables, the East-West light's color EW, which is either *red* or *green*, and the North-South light's color NS, which is also either red or green. When the lights are functioning, exactly one of the lights is green, while when the lights are broken, both are red (and flashing, but the agent cannot perceive this distinction). The agent has several observations as training data:

NS	EW	W
r	g	w
g	r	w
r	r	$\neg w$
g	r	w
r	g	w
r	g	w
g	r	w

(a) **(4 pts)** Assume the agent uses a naive Bayes model to make its predictions. Based on the training data above, fill in the CPTs below (use the maximum likelihood estimates, i.e. no smoothing).

(b) **(6 pts)** Assume the agent observes that both lights are red. What is the posterior probability that the lights are working *according to the agent's model*.

(c) **(3 pts)** Note that in the data, all red/red data points are from broken lights. Explain what critical conditional dependence the model fails to capture.

(d) **(3 pts)** Draw a Bayes' net which would allow the agent to draw the correct inference in the case of two red lights (as well as the working light cases). You should not need to introduce any new variables.

(e) **(4 pts)** Give a minimal set of features which would be sufficient for a perceptron to correctly predict W on all training examples. State your features precisely as functions from inputs $x = (ew, ns)$ to real numbers.

7. (22 points.) HMMs

You sometimes get colds, which make you sneeze. You also get allergies, which make you sneeze. Sometimes you are well, which doesn't make you sneeze (much). You decide to model the process using the following HMM, with hidden states $X \in \{well, allergy, cold\}$ and observations $E \in \{sneeze, quiet\}$:

Transitions

$P(X_1)$

well	1
allergy	0
cold	0

$P(X_t \mid X_{t-1} = well)$

well	0.7
allergy	0.2
cold	0.1

$P(X_t \mid X_{t-1} = allergy)$

well	0.6
allergy	0.3
cold	0.1

$P(X_t \mid X_{t-1} = cold)$

well	0.2
allergy	0.2
cold	0.6

Emissions

$P(E_t \mid X_t = well)$

quiet	1.0
sneeze	0.0

$P(E_t \mid X_t = allergy)$

quiet	0.0
sneeze	1.0

$P(E_t \mid X_t = cold)$

quiet	0.0
sneeze	1.0

Note that colds are "stickier" in that you tend to have them for multiple days, while allergies come and go on a quicker time scale. However, allergies are more frequent. Assume that on the first day, you are well.

(a) **(2 pts)** Imagine you observe the sequence *quiet, sneeze, sneeze.* What is the probability that you were well all three days and observed these effects?

(b) **(4 pts)** What is the posterior distribution over your state on day 2 (X_2) if $E_1 = quiet$, $E_2 = sneeze$?

(c) **(4 pts)** What is the posterior distribution over your state on day 3 (X_3) if $E_1 = quiet$, $E_2 = sneeze$, $E_3 = sneeze$?

(d) **(4 pts)** What is the Viterbi (most likely) sequence for the observation sequence *quiet, sneeze, sneeze, sneeze, quiet, quiet, sneeze, quiet, quiet*? *Hint: you should not have to do extensive calculations.*

Imagine you are monitoring your state using the particle filtering algorithm, and on a given day you have 5 particles on *well*, 2 on *cold*, and 3 on *allergy* before making an observation on that day.

(e) **(4 pts)** If you observe *sneeze*, what weight will each of your particles have?

(f) **(4 pts)** After resampling, what is the expected number of particles you will have on *cold*?

8. (18 points.) Short answer

Each question should answered by no more than one or two sentences! **(3 pts each)**

(a) In robot motion planning, why might we not prefer to find shortest paths through configuration space?

(b) Why do we smooth probability estimates in naive Bayes' classifiers?

(c) Name two advantages to a particle filtering approach to tracking / monitoring in an HMM (compared to the forward algorithm).

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

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

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

End of Exam