CS 188
Spring 2009

Introduction to
Artificial Intelligence

Midterm Exam

INSTRUCTIONS

- You have 3 hours.
- The exam is closed book, closed notes except a one-page crib sheet.
- Please use non-programmable calculators only.
- Mark your answers ON THE EXAM ITSELF. 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.

Last Name	
First Name	
SID	
Login	
GSI	
Section Time	
All the work on this exam is my own. **(please sign)**	

For staff use only

Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Total
/16	/15	/20	/10	/21	/18	/100

1. (16 points) True/False

For the following questions, a correct answer is worth 2 points, no answer is worth 1 point, and an incorrect answer is worth 0 points. Circle *true* or *false* to indicate your answer.

a) (*true* or *false*) If $g(s)$ and $h(s)$ are two admissible A* heuristics, then their average $f(s) = \frac{1}{2}g(s) + \frac{1}{2}h(s)$ must also be admissible.

b) (*true* or *false*) For a search problem, the path returned by uniform cost search may change if we add a positive constant C to every step cost.

c) (*true* or *false*) The running-time of an efficient solver for tree-structured constraint satisfaction problems is linear in the number of variables.

d) (*true* or *false*) If $h_1(s)$ is a consistent heuristic and $h_2(s)$ is an admissible heuristic, then $\min(h_1(s), h_2(s))$ must be consistent.

e) (*true* or *false*) The amount of memory required to run minimax with alpha-beta pruning is $O(b^d)$ for branching factor b and depth limit d.

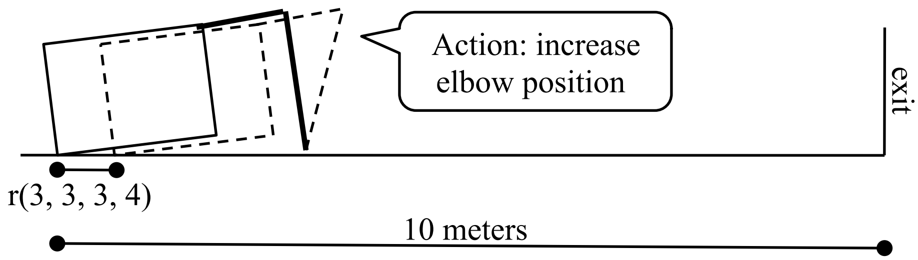
f) (*true* or *false*) In a Markov decision process with discount $\gamma = 1$, the difference in values for two adjacent states is bounded by the reward between them: $|V(s) - V(s')| \leq \max_a R(s, a, s')$.

g) (*true* or *false*) Value iteration and policy iteration must always converge to the same policy.

h) (*true* or *false*) In a Bayes' net, if $A \perp\!\!\!\perp B$, then $A \perp\!\!\!\perp B \mid C$ for some variable C other than A or B.

2. (15 points) Search: Crawler's Escape

Whilst Pacman was Q-learning, Crawler snuck into `mediumClassic` and stole all the dots. Now, it's trying to escape as quickly as possible. At each time step, Crawler can *either* move its *s*houlder or its *e*lbow one position up or down. Both joints s and e have five total positions each (1 through 5) and both begin in position 3. Upon changing arm positions from (s, e) to (s', e'), Crawler's body moves some distance $r(s, e, s', e')$, where $|r(s, e, s', e')| \leq 2$ meters (negative distance means the crawler moves backwards). Crawler must travel 10 meters to reach the exit.



In this problem, you will design a search problem for which the optimal solution will allow Crawler to escape in as few time steps as possible.

(a) (3 pt) Define a state space, start state and goal test to represent this problem.

(b) (3 pt) Define the successor function for the start state by listing its (successor state, action, step cost) triples. Use the actions s_+ and s_- for increasing and decreasing the shoulder position and e_+ and e_- for the elbow.

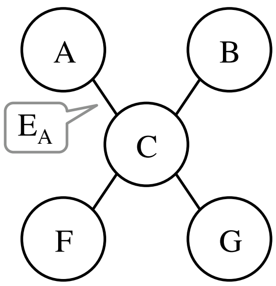
(c) (2 pt) Would depth-first graph search be complete for the search problem you defined? Explain.

(d) (3 pt) Design a non-trivial, admissible heuristic for this problem.

(e) (4 pt) Crawler's shoulder overheats if it switches direction more than once in any three consecutive time steps, making progress impossible. Describe how you would change your search problem so that an optimal search algorithm would only return solutions that avoid overheating.

3. (20 points) CSPs: Constraining Graph Structure

Hired as a consultant, you built a Bayes' net model of Cheeseboard Pizza customers. Last night, a corporate spy from Zachary's Pizza deleted the *directions* from all your arcs. You still have the undirected graph of your model and a list of dependencies. You now have to recover the direction of the arrows to build a graph that allows for these dependencies. *Note: $X \not\perp\!\!\!\perp Y$ means X is not independent of Y.*



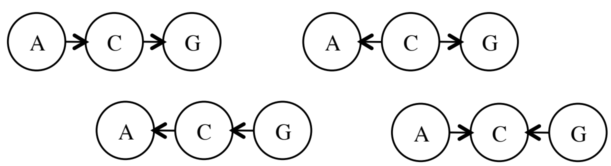
Distribution properties (dependencies):

$A \not\perp\!\!\!\perp G$

$B \not\perp\!\!\!\perp F$

$B \not\perp\!\!\!\perp G \mid C$

a) (1 pt) Given the first constraint only, circle all the topologies that are allowed for the triple (A, C, G).



b) (3 pt) Formulate direction-recovery for this graph as a CSP with *explicit binary constraints* only. The variables and values are provided for you. The variable E_A stands for the direction of the arc between A and the center C, where *out* means the arrow points outward (toward A), and *in* means the arrow points inward (toward C).

Variables: E_A, E_B, E_F, E_G

Values: *out*, *in*

Constraints:

c) (1 pt) Draw the constraint graph for this CSP.

d) (2 pt) After selecting $E_A = in$, cross out all values eliminated from E_B, E_F and E_G by forward checking.

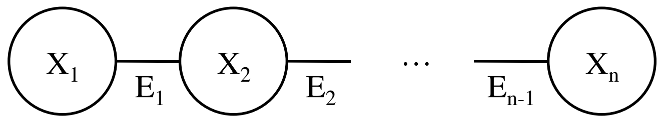
E_A	E_B	E_F	E_G
in	*out in*	*out in*	*out in*

e) (2 pt) Cross out all values eliminated by arc consistency applied before any backtracking search.

E_A	E_B	E_F	E_G
out in	*out in*	*out in*	*out in*

f) (1 pt) Solve this CSP, then add the correct directions to the arcs of the graph at the top of the page.

Now consider a chain structured graph over variables $X_1, \ldots, X_n$. Edges $E_1, \ldots, E_{n-1}$ connect adjacent variables, but their directions are again unknown.



g) **(4 pt)** Using only *binary* constraints and *two-valued* variables, formulate a CSP that is satisfied by *only and all* networks that can represent a distribution where $X_1 \not\perp\!\!\!\perp X_n$. Describe what your variables mean in terms of direction of the arrows in the network.

Variables:

Constraints:

h) **(6 pt)** Using only *unary, binary and ternary* (3 variable) constraints and *two-valued* variables, formulate a CSP that is satisfied by *only and all* networks that enforce $X_1 \perp\!\!\!\perp X_n$. Describe what your variables mean in terms of direction of the arrows in the network. *Hint: You will have to introduce additional variables.*

Variables:

Constraints:

4. (10 points) Multi-agent Search: Connect-3

In the game of Connect-3, players X and O alternate moves, dropping their symbols into columns 1, 2, 3, or 4. Three-in-a-row wins the game, horizontally, vertically or diagonally. X plays first.

X	O	O	O
O	X	X	O
X	X	O	X
1	2	3	4

X	O		
O	X		O
X	X	O	X
1	2	3	4

(a) (1 pt) What is the maximum branching factor of minimax search for Connect-3?

(b) (1 pt) What is the maximum tree depth in plies?

(c) (1 pt) Give a reasonably tight upper bound on the number of terminal states.

(d) (2 pt) Draw the game tree starting from the board shown above right (with O to play next). You may abbreviate states by drawing only the upper-right region circled. The root node is drawn for you.

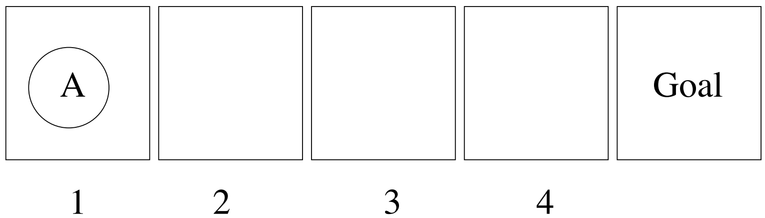
	O

(e) (2 pt) X is the maximizer, while O is the minimizer. X's utility for terminal states is k when X wins and $-k$ when O wins, where k is 1 for a horizontal 3-in-a-row win (as in above left), 2 for a vertical win, and 3 for a diagonal win. A tie has value 0. Label each node of your tree with its minimax value.

(f) (3 pt) Circle all nodes of your tree that will *not* be explored when using alpha-beta pruning and a move ordering that maximizes the number of nodes pruned.

5. (21 points) MDPs: Robot Soccer

A soccer robot **A** is on a fast break toward the goal, starting in position 1. From positions 1 through 3, it can either shoot (S) or dribble the ball forward (D); from 4 it can only shoot. If it shoots, it either scores a goal (state G) or misses (state M). If it dribbles, it either advances a square or loses the ball, ending up in M.



In this MDP, the states are 1, 2, 3, 4, G and M, where G and M are terminal states. The transition model depends on the parameter y, which is the probability of dribbling success. Assume a discount of $\gamma = 1$.

$T(k, S, G) = \frac{k}{6}$ $\quad T(k, S, M) = 1 - \frac{k}{6}$ $\quad$ for $k \in \{1, 2, 3, 4\}$
$T(k, D, k+1) = y$ $\quad T(k, D, M) = 1 - y$ $\quad$ for $k \in \{1, 2, 3\}$
$R(k, S, G) = 1$ $\quad$ for $k \in \{1, 2, 3, 4\}$, $\quad$ and rewards are 0 for all other transitions

(a) (2 pt) What is $V^\pi(1)$ for the policy π that always shoots?

(b) (2 pt) What is $Q^*(3, D)$ in terms of y?

(c) (2 pt) Using $y = \frac{3}{4}$, complete the first two iterations of value iteration.

i	$V_i^*(1)$	$V_i^*(2)$	$V_i^*(3)$	$V_i^*(4)$
0	0	0	0	0
1				
2				

(d) (2 pt) After how many iterations will value iteration compute the optimal values for all states?

(e) (2 pt) For what range of values of y is $Q^*(3, S) \geq Q^*(3, D)$?

The dribble success probability y in fact depend on the presence or absence of a defending robot, **D**. **A** has no way of detecting whether **D** is present, but does know some statistical properties of its environment. **D** is present $\frac{2}{3}$ of the time. When **D** is absent, $y = \frac{3}{4}$. When **D** is present, $y = \frac{1}{4}$.

(f) **(4 pt)** What is the posterior probability that **D** is present, given that **A** Dribbles twice successfully from 1 to 3, then Shoots from state 3 and scores.

(g) **(3 pt)** What transition model should **A** use in order to correctly compute its maximum expected reward when it doesn't know whether or not **D** is present?

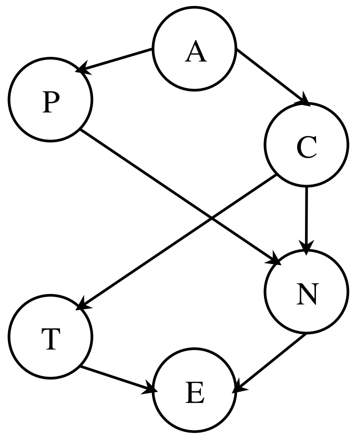
$T(k, S, G) =$ $\qquad$ $T(k, S, M) =$ $\qquad$ for $k \in \{1, 2, 3, 4\}$

$T(k, D, k+1) =$ $\qquad$ $T(k, D, M) =$ $\qquad$ for $k \in \{1, 2, 3\}$

(h) **(4 pt)** What is the optimal policy π^* when **A** doesn't know whether or not **D** is present?

6. (18 points) Bayes' Nets: The Mind of Pacman

Pacman doesn't just eat dots. He also enjoys pears, apples, carrots, nuts, eggs and toast. Each morning, he chooses to eat some subset of these foods. His preferences are modeled by a Bayes' net with this structure.



a) (2 pt) Factor the probability that Pacman chooses to eat only apples and nuts in terms of conditional probabilities from this Bayes' net.

$P(\neg p, a, \neg c, n, \neg e, \neg t) =$

b) (4 pt) For each of the following properties, circle whether they are *true*, *false* or *unknown* of the distribution $P(P, A, C, N, E, T)$ for this Bayes' net.

$N \perp\!\!\!\perp T$	(*true* , *false* , *unknown*)
$P \perp\!\!\!\perp E \mid N$	(*true* , *false* , *unknown*)
$P \perp\!\!\!\perp N \mid A, C$	(*true* , *false* , *unknown*)
$E \perp\!\!\!\perp A \mid C, N$	(*true* , *false* , *unknown*)

c) (2 pt) If the arrow from T to E were reversed, list two conditional independence properties that would be true under the new graph that were not guaranteed under the original graph.

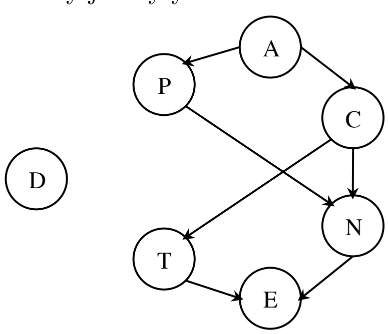
d) (2 pt) You discover that $P(a|\neg c, \neg t) > P(a|\neg c)$. Suggest how you might change the network in response.

e) (4 pt) You now want to compute the distribution $P(A)$ using variable elimination. List the factors that remain before and after eliminating the variable N.

Before:

After:

f) (2 pt) Pacman's new diet allows only fruit (P and A) to be eaten, but Pacman only follows the diet occasionally. Add the new variable D (for whether he follows the diet) to the network below by adding arcs. Briefly justify your answer.



g) (2 pt) Given the Bayes' net and the factors below, fill in the table for $P(D|\neg a)$ or state that there is not enough information.

D	$P(D)$
d	0.4
$\neg d$	0.6

D	A	$P(A\|D)$
d	a	0.8
d	$\neg a$	0.2
$\neg d$	a	0.5
$\neg d$	$\neg a$	0.5

D	A	$P(D\|A = \neg a)$
d	$\neg a$	
$\neg d$	$\neg a$	