CS 188
Fall 2010

Introduction to
Artificial Intelligence

Final Exam

INSTRUCTIONS

- You have 3 hours.
- The exam is closed book, closed notes except a two-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	
Name of the person to your left	
Name of the person to your right	
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
/21	/5	/9	/11	/7	/6	
Q. 7	Q. 8	Q. 9	Q. 10	Q. 11		
/5	/10	/12	/7	/7		/100

THIS PAGE INTENTIONALLY LEFT BLANK

1. (21 points) Everything

(a) (1 pt) CS 188

Circle the best motto for AI.

i. Maximize your expected utilities.

(b) (2 pt) Search

Circle all of the following statements that are true, if any. **Ignore ties** in all cases.

i. Breadth-first search is a special case of depth-first search. (There is a way to get depth-first search to generate the same search order as breadth-first search).

ii. Depth-first search is a special case of uniform-cost search. (There is a way to get uniform-cost search to generate the same search order as depth-first search).

iii. Uniform-cost search is a special case of A* search. (There is a way to get A* search to generate the same search order as uniform-cost search).

iv. A* search can perform breadth-first search under some class of admissible heuristics and cost functions.

v. A* search can perform depth-first search under some class of admissible heuristics and cost functions.

(c) (2 pt) CSP

For each of the heuristics, circle the **single choice** that best describes what they're doing.

A. Minimum Remaining Values (MRV)

i. Focuses on the hard parts of the graph in order to fail quickly.

ii. Focuses on the easy parts of the graph in order to postpone failure.

iii. Maximizes the chance of the solution succeeding without backtracking.

iv. Minimizes the chance of the solution succeeding without backtracking.

B. Least Constraining Value

i. Focuses on the hard parts of the graph in order to fail quickly.

ii. Focuses on the easy parts of the graph in order to postpone failure.

iii. Maximizes the chance of the solution succeeding without backtracking.

iv. Minimizes the chance of the solution succeeding without backtracking.

(d) (3 pt) Games

Say we have player MAX and player MIN playing a game with a finite number of possible moves. MAX calculates the minimax value of the root to be M. You may assume that each player has at least 2 possible actions at every turn. Also, you may assume for all parts that a different sequence of moves will always lead to a different score (**no two sequences yield the same score**). Circle all of the following statements that are true, if any.

i. Assume MIN is playing suboptimally, and MAX **does not** know this. The outcome of the game can be better than M (i.e. higher for MAX).

ii. Assume MAX **knows** player MIN is playing randomly. There exists a policy for MAX such that MAX can guarantee a better outcome than M.

iii. Assume MAX **knows** MIN is playing suboptimally at all times and **knows** the policy π_{MIN} that MIN is using (MAX knows exactly how MIN will play). There exists a policy for MAX such that MAX can guarantee a better outcome than M.

iv. Assume MAX **knows** MIN is playing suboptimally at all times but **does not know** the policy π_{MIN} that MIN is using (MAX knows MIN will choose a suboptimal action at each turn, but does not know which suboptimal action). There exists a policy for MAX such that MAX can guarantee a better outcome than M.

(e) (2 pt) MDPs

Circle all of the following statements that are true, if any.

i. If one is using value iteration and the policy (the greedy policy with respect to the values) has converged, the values must have converged as well.

ii. If one is using value iteration and the values have converged, the policy must have converged as well.

iii. Expectimax will generally run in the same amount of time as value iteration on a given MDP.

iv. Policy iteration will converge to an optimal policy.

(f) (3 pt) Reinforcement Learning

We are given an MDP (S, A, T, γ, R), and a policy π (not necessarily the optimal policy). For each of the following Bellman-like update equations, **circle the single correct choice** to match the equations with the quantity being computed (V^π, Q^π, V^*, Q^*, π^*, or none of these).

i. $g_1(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s') \left[R(s, a, s') + \gamma \max_{a' \in A} Q^*(s', a')\right]$

(a) V^π (b) Q^π (c) V^* (d)Q^* (e)π^* (f) none of these.

ii. $g_2(s) = \arg\max_{a \in A} Q^*(s, a)$

(a) V^π (b) Q^π (c) V^* (d)Q^* (e)π^* (f) none of these.

iii. $g_3(s, a) = \sum_{s' \in S} T(s, a, s') \left[R(s, a, s') + \gamma g_3(s', \pi(s'))\right]$

(a) V^π (b) Q^π (c) V^* (d)Q^* (e)π^* (f) none of these.

iv. $g_4(s) = \sum_{s' \in S} T(s, \pi(s), s') \left[R(s, \pi(s), s') + \gamma g_4(s')\right]$

(a) V^π (b) Q^π (c) V^* (d)Q^* (e)π^* (f) none of these.

v. $g_5(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s') \left[R(s, a, s') + \gamma \max_{s'' \in S} g_5(s'')\right]$

(a) V^π (b) Q^π (c) V^* (d)Q^* (e)π^* (f) none of these.

(g) (2 pt) Probability

Circle all of the following equalities that are **always** true, if any.

i. $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$

ii. $\mathbf{P}(A|B) = \mathbf{P}(A)\mathbf{P}(B)$

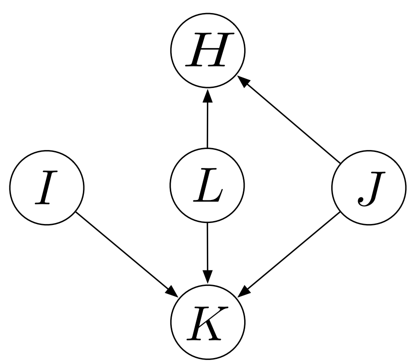
iii. $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B) - \mathbf{P}(A|B)$

iv. $\mathbf{P}(A, B, C) = \mathbf{P}(A|B, C)\mathbf{P}(B|C)\mathbf{P}(C)$

v. $\mathbf{P}(A, B) = \sum_{c \in C} \mathbf{P}(A|B, C = c)\mathbf{P}(B|C = c)\mathbf{P}(C = c)$

(h) (2 pt) Bayes' Nets

For the Bayes' Net shown below, you start with the factors $\mathbf{P}(I), \mathbf{P}(J), \mathbf{P}(L), \mathbf{P}(K|I, L, J)$, and $\mathbf{P}(H|L, J)$.



What are the factors after joining on and eliminating J?

(i) (2 pt) Particle Filtering

Circle all of the following statements that are true, if any.

i. It is possible to use particle filtering when the state space is continuous.

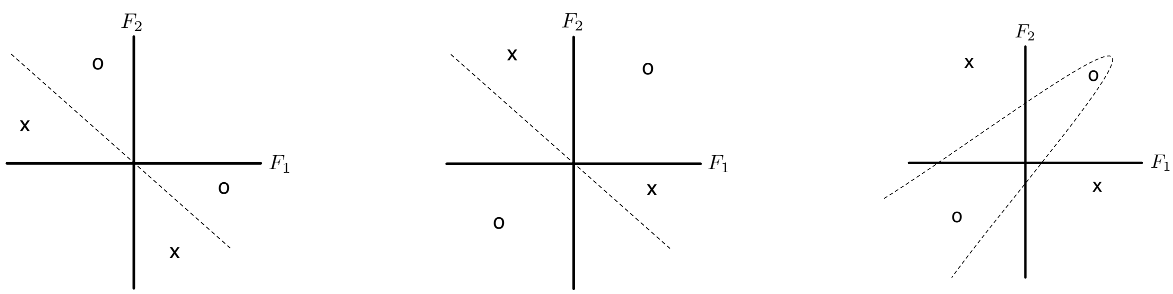
ii. It is possible to use particle filtering when the state space is discrete.

iii. As the number of particles goes to infinity, particle filtering will represent the same probability distribution that you'd get by using exact inference.

iv. Particle filtering can represent a flat distribution (i.e. uniform) with fewer particles than it would need for a more concentrated distribution (i.e. Gaussian).

(j) (1 pt) Perceptron

Suppose you have a classification problem with classes $Y = \text{X}, \text{O}$ and features F_1, F_2. You decide to use the perceptron algorithm to classify the data. Suppose you run the algorithm for each of the data sets shown below, stopping either after convergence or 1000 iterations (you may assume that if the algorithm will converge, it will within 1000 iterations). Circle all of the examples, if any, where the decision boundary could possibly be created by the perceptron algorithm.



(k) (1 pt) Inverse Reinforcement Learning

What quantity of an MDP is inverse reinforcement learning trying to estimate?

$Q(s,a)$ $\quad$ $R(s,a,s')$ $\quad$ $V(s)$ $\quad$ $T(s,a,s')$

2. (5 points) Probability: Monsters vs. Aliens

(a) (2 pt) Aliens

$$F \longrightarrow A$$

Aliens can be friendly or not; 75% are friendly ($F \in \{\text{friendly}, \text{unfriendly}\}$). Friendly aliens arrive during the day 90% of the time, while unfriendly ones always arrive at night ($A \in \{\text{day}, \text{night}\}$). What fraction of night-arriving aliens are friendly?

(b) (3 pt) Monsters

$$L \longrightarrow F$$

Half of all monsters live in attics, while the rest live in basements ($L \in \{\text{attic}, \text{basement}\}$). Attic-living monsters are always fuzzy, while a randomly chosen monster has only an 80% chance of being fuzzy ($F \in \{\text{fuzzy}, \text{unfuzzy}\}$). What fraction of basement-living monsters are fuzzy?

3. (9 points) VPI: Crack the Code

You are defusing a bomb constructed by the evil Dr. Xor. You know the shutdown sequence is three bits B_1, B_2, B_3, $B_i \in \{0, 1\}$ and you know that an odd number of the B_i are 1. Otherwise, all sequences are equally likely. You must pick a sequence, at which point the bomb either deactivates, or not. You get a utility of 100 if you guess the correct sequences, and 0 otherwise. (Hint: you should not need to do much calculation.)

(a) (1 pt) Draw a minimal (fewest arcs) Bayes' Net that can represent the joint distribution over these variables. You only need to include B_1, B_2, and B_3 (not the utility).

(b) (1 pt) What is the MEU given no evidence?

(c) (1 pt) What is the VPI of B_1 given no information?

(d) (1 pt) What is the VPI of B_2 given B_1?

(e) (1 pt) What is the VPI of B_3 given B_1 and B_2?

At the last second, you discover that the bomb was actually set by Xor's uncreative henchman, Repeato. Repeato always uses all 1's or all 0's in his code (and is not restricted to an odd number of 1's).

(f) (1 pt) Draw a minimal (fewest arcs) Bayes' Net that can represent the joint distribution over these variables. You only need to include B_1, B_2, and B_3 (not the utility).

(g) (1 pt) What is the VPI of B_1 given no information?

(h) (1 pt) What is the VPI of B_2 given B_1?

(i) (1 pt) What is the VPI of B_3 given B_1 and B_2?

4. (11 points) MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. Otherwise, you must Stop. When you Stop, your utility is equal to your total score (up to 5), or zero if you get a total of 6 or higher. When you Draw, you receive no utility. There is no discount ($\gamma = 1$).

(a) (2 pt) What is the state space for this MDP?

(b) (2 pt) What is the reward function for this MDP?

(c) (2 pt) Give the optimal policy for this MDP.

(d) (2 pt) What is the smallest number of rounds (k) of value iteration for which this MDP will have its exact values (if value iteration will never converge exactly, state so).

(e) (3 pt) Imagine that you run Q-learning instead of calculating values offline. You play many games, and frequently choose all actions from states that you visit. However, due to bizarre luck, each card is a 2. What will the final q-values approach in the limit if they are initialized to zero and you use a learning rate of 1/2?

5. (7 points) Search: Expanded Nodes

Consider tree search (i.e. no closed set) on an arbitrary search problem with max branching factor b. Each search node n has a backward (cumulative) cost of $g(n)$, an admissible heuristic of $h(n)$, and a depth of $d(n)$. Let c be a minimum-cost goal node, and let s be a shallowest goal node.

For each of the following, you will give an expression that characterizes the set of nodes that are expanded before the search terminates. For instance, if we asked for the set of nodes with positive heuristic value, you could say $h(n) \geq 0$. Don't worry about ties (so you won't need to worry about $>$ versus $\geq$). If there are no nodes for which the expression is true, you must write "none."

(a) (1 pt) Give an expression (i.e. an inequality in terms of the above quantities) for which nodes n will be expanded in a breadth-first search.

(b) (1 pt) Give an expression for which nodes n will be expanded in a uniform cost search.

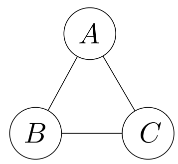
(c) (1 pt) Give an expression for which nodes n will be expanded in an A* search with heuristic $h(n)$.

(d) (2 pt) Let h_1 and h_2 be two admissible heuristics such that $\forall n, h_1(n) \geq h_2(n)$. Give an expression for the nodes which will be expanded in an A* search using h_1 but not when using h_2.

(e) (2 pt) Give an expression for the nodes which will be expanded in an A* search using h_2 but not when using h_1.

6. (6 points) CSPs: Arc Consistency

Consider the following CSP graph. Each variable is binary valued (0 or 1). For each of the following sets of constraints, circle all true statements, if any.



(a) (2 pt) $A = B$, $B = C$, $C = A$

(i) The CSP has no solutions, and enforcing arc consistency will expose it.

(ii) The CSP has no solutions, but enforcing arc consistency will not expose it.

(iii) The CSP has exactly one solution, and arc consistency will narrow domains to this solution.

(iv) The CSP has exactly one solution, but arc consistency will not narrow domains to this solution.

(v) The CSP has multiple solutions, and arc consistency will rule out all but one.

(vi) The CSP has multiple solutions, and arc consistency will leave domains so that all are possible.

(b) (2 pt) $A \neq B$, $B \neq C$, $C \neq A$

(i) The CSP has no solutions, and enforcing arc consistency will expose it.

(ii) The CSP has no solutions, but enforcing arc consistency will not expose it.

(iii) The CSP has exactly one solution, and arc consistency will narrow domains to this solution.

(iv) The CSP has exactly one solution, but arc consistency will not narrow domains to this solution.

(v) The CSP has multiple solutions, and arc consistency will rule out all but one.

(vi) The CSP has multiple solutions, and arc consistency will leave domains so that all are possible.

(c) (2 pt) $A < B$, $B < C$, $C < A$

(i) The CSP has no solutions, and enforcing arc consistency will expose it.

(ii) The CSP has no solutions, but enforcing arc consistency will not expose it.

(iii) The CSP has exactly one solution, and arc consistency will narrow domains to this solution.

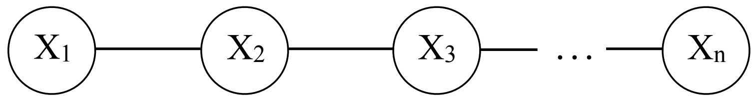
(iv) The CSP has exactly one solution, but arc consistency will not narrow domains to this solution.

(v) The CSP has multiple solutions, and arc consistency will rule out all but one.

(vi) The CSP has multiple solutions, and arc consistency will leave domains so that all are possible.

7. (5 points) Linear-Chain CSPs

You would like to solve a CSP with the following constraint graph over variables (X_t) with domain D for $t = 1 \dots n$.



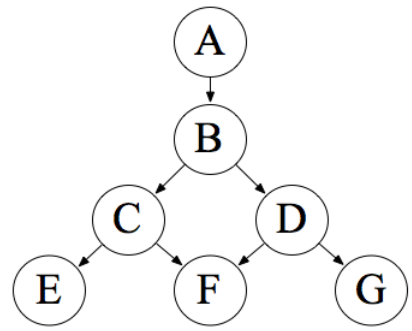
Let $C_t(X_t = x_t, X_{t-1} = x_{t-1})$ be a function which is 1 if $(X_t = x_t, X_{t-1} = x_{t-1})$ satisfies all binary constraints involving (X_t, X_{t-1}), and 0 otherwise. Let $F_t(X_t = x_t)$ be a function which is greater than zero if and only if there is an assignment of variables $(X_1, X_2, \dots, X_t)$ which satisfies all constraints on those variables, and assigns $X_t = x_t$. Let $F_1(X_1 = x_1) = 1$ for all $x_1 \in D$.

(a) (3 pt) Provide a recurrence that can be used to compute $F_t(X_t)$ in terms of $F_{t-1}(X_{t-1})$.

(b) (2 pt) Suppose a black box computes values of $F_t(X_t = x_t)$ for all t and x_t. How would you determine if there is a solution to the CSP?

8. (10 points) D-Separation as Search

(a) (2 pt) Consider the following Bayes' Net:



(i) Given B, what variable(s) is E guaranteed to be independent of? (Circle all that apply.)

A C D F G

(ii) Given B and F, what variable(s) is G guaranteed to be independent of? (Circle all that apply.)

A C D E

(b) (6 pt) Now we'd like to formulate d-separation as a search problem. Specifically, you're given a variable X and a variable Y, a Bayes' Net G, and a set of observed variables E. You're also given E^+, which is the set of variables that are the parents or ancestors of evidence variables. Given this information, define a search problem that finds Y if X and Y are not d-separated, and does not find a goal otherwise. You may find the notation $W \to U \in G$ meaning "an arc from W to U is in the Bayes' Net" helpful. A full credit solution will have a minimal state space.

State Space:

Initial State:

Successor Function:

Goal Test:

(c) (2 pt) Give a non-trivial consistent heuristic for this problem.

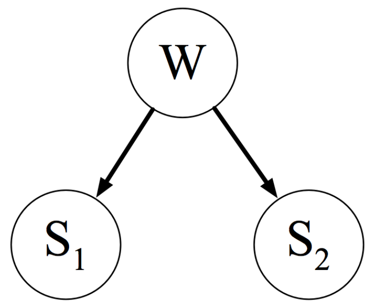
9. (12 points) Naive Bayes and Perceptron

Stoplights S_1 and S_2 can each be in one of two states: green (g) or red (r). Additionally, the machinery behind both stoplights (W) can be in one of two states: working (w) or broken (b). We collect data by observing the stoplights and the state of their machinery on seven different days. Here's a naive Bayes graphical model for the stoplights:

Data:

Day	S_1	S_2	W
1	g	r	w
2	g	r	w
3	g	r	w
4	r	g	w
5	r	g	w
6	r	g	w
7	r	r	b

Model:



(a) (1 pt) Fill in tables for $P(W)$, $P(S_1|W)$, $P(S_2|W)$ with probabilities that give the naive Bayes joint distribution that assigns highest probability to the data we observed.

W	$\mathbf{P}(W)$
w	
b	

S_1	W	$\mathbf{P}(S_1|W)$
g	w	
r	w	
g	b	
r	b	

S_2	W	$\mathbf{P}(S_2|W)$
g	w	
r	w	
g	b	
r	b	

(b) (2 pt) What's the posterior probability $P(W = b|S_1 = r, S_2 = r)$?

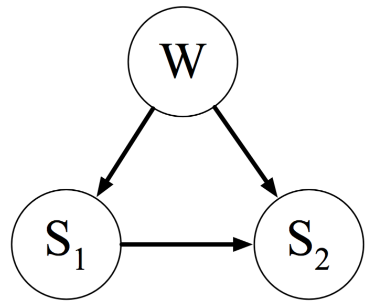
(c) (2 pt) Estimate each of $P(W)$, $P(S_1|W)$, and $P(S_2|W)$ using add-k smoothing with $k = 1$ (also smooth $P(W)$). Fill in their values in the tables below:

W	$\mathbf{P}(W)$
w	
b	

S_1	W	$\mathbf{P}(S_1|W)$
g	w	
r	w	
g	b	
r	b	

S_2	W	$\mathbf{P}(S_2|W)$
g	w	
r	w	
g	b	
r	b	

What if instead of naive Bayes we use the following graphical model and fill in probability tables with estimates that assign highest probability to the data we observed:



(d) (2 pt) What's the posterior probability $P(W = b|S_1 = r, S_2 = r)$? (Hint: you should not have to do a lot of work.)

(e) (1 pt) What is it about the problem that makes the second graphical model more apt?

(f) (2 pt) Let's see what perceptron does with the data we observed. Use only the two features f_{S_1} and f_{S_2} where $f_{S_1} = +1$ if $S_1 = g$ and $f_{S_1} = -1$ if $S_1 = r$, and similarly for f_{S_2}. Treat W as the label.

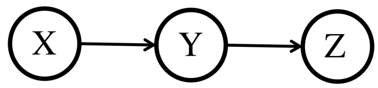
Initialize all weights to 0 and perform one pass of perceptron training on the data, doing updates in the order that the data points were observed. Break ties by choosing $W = w$. What are the final weights?

Day	S_1	S_2	W
1	g	r	w
2	g	r	w
3	g	r	w
4	r	g	w
5	r	g	w
6	r	g	w
7	r	r	b

(g) (2 pt) Will perceptron converge if you run it long enough? Justify your answer.

10. (7 points) Sampling

Assume the following Bayes net, and the corresponding distributions over the variables in the Bayes net:



X	$\mathbf{P}(X)$
$+x$	$2/5$
$-x$	$3/5$

Y	X	$\mathbf{P}(Y\|X)$
$+y$	$+x$	$2/3$
$-y$	$+x$	$1/3$
$+y$	$-x$	$3/4$
$-y$	$-x$	$1/4$

Z	Y	$\mathbf{P}(Z\|Y)$
$+z$	$+y$	$1/3$
$-z$	$+y$	$2/3$
$+z$	$-y$	$1/5$
$-z$	$-y$	$4/5$

(a) (1 pt) Your task is now to estimate $\mathbf{P}(+y|+x,+z)$ using rejection sampling. Below are some samples that have been produced by prior sampling (that is, the rejection stage in rejection sampling hasn't happened yet). Cross out whichever of the following samples that would be rejected by rejection sampling:

$$\begin{array}{ccc} +x, & +y, & +z \\ -x, & +y, & +z \\ -x, & -y, & +z \\ +x, & -y, & -z \\ +x, & -y, & +z \end{array}$$

(b) (2 pt) Using rejection sampling, give an estimate of $\mathbf{P}(+y|+x,+z)$ from these samples, or state why it cannot be computed.

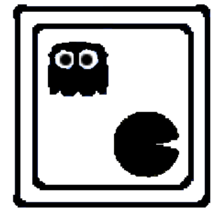
(c) (2 pt) Using the following samples (which were generated using likelihood weighting), estimate $\mathbf{P}(+y|+x,+z)$ using likelihood weighting, or state why it cannot be computed.

$$\begin{array}{ccc} +x, & +y, & +z \\ +x, & -y, & +z \\ +x, & +y, & +z \end{array}$$

(d) (2 pt) Which query is better suited for likelihood weighting, $\mathbf{P}(Z|X)$ or $\mathbf{P}(X|Z)$? Justify your answer.

11. (7 points) Pursuit Evasion

Pacman is trapped in the following 2 by 2 maze with a hungry ghost (the horror)! When it is his turn to move, Pacman must move one step horizontally or vertically to a neighboring square. When it is the ghost's turn, he must also move one step horizontally or vertically. The ghost and Pacman alternate moves. After every move (by either the ghost or Pacman) if Pacman and the ghost occupy the same square, Pacman is eaten and receives utility -100. Otherwise, he receives a utility of 1. The ghost attempts to minimize the utility that Pacman receives. **Assume the ghost makes the first move**.



For example, with a discount factor of $\gamma = 1.0$, if the ghost moves down, then Pacman moves left, Pacman earns a reward of 1 after the ghost's move and -100 after his move for a total utility of -99.

Note that this game is not guaranteed to terminate.

(a) **(1 pt)** Assume a discount factor $\gamma = 0.5$, where the discount factor is applied once every time either Pacman or the ghost moves. What is the minimax value of the truncated game after 2 ghost moves and 2 Pacman moves? (Hint: you should not need to build the minimax tree)

(b) **(1 pt)** Assume a discount factor $\gamma = 0.5$. What is the minimax value of the complete (infinite) game? (Hint: you should not need to build the minimax tree)

(c) **(2 pt)** Why is value iteration superior to minimax for solving this game?

(d) **(3 pt)** This game is similar to an MDP because rewards are earned at every timestep. However, it is also an adversarial game involving decisions by two agents.

Let s be the state (e.g. the position of Pacman and the ghost), and let $A_P(s)$ be the space of actions available to Pacman in state s (and similarly let $A_G(s)$ be the space of actions available to the ghost). Let $N(s, a) = s'$ denote the successor function (given a starting state s, this function returns the state s' which results after taking action a). Finally, let $R(s)$ denote the utility received after moving to state s.

Write down an expression for $P^*(s)$, the value of the game to Pacman as a function of the current state s (analogous to the Bellman equations). Use a discount factor of $\gamma = 1.0$. Hint: your answer should include $P^*(s)$ on the right hand side.

$P^*(s) =$