CSEP 573 - Artificial Intelligence
Midterm 2
March 7, 2019

Name: Jianming Xiao.

p1	p2	total
18 +9	3	42
p3 2 ~~8~~ 1	p4 9	

This test is closed book; no calculators or Internet is allowed. (You may bring one 8.5 x 11" piece of paper with anything written on it, if you like).

If any question is ambiguous, feel free to make an assumption in order to answer it, but A) **state your assumption clearly** as part of the answer, and B) your grade will reflect the quality of the assumption.

1) True / False: Circle the correct answer. (2 points each) **We'll give you one point free if you leave the answer blank** (zero points for the wrong answer), so guess with care; it can hurt your score. Some questions are tricky; read carefully!

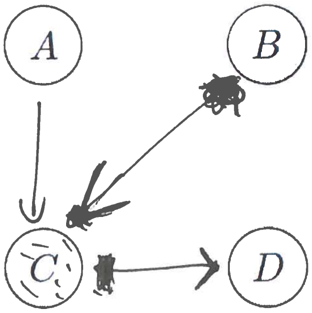
4	1	T	[✓ F]	A pattern database helps an agent avoid wasting time in cycles by storing previously-expanded states.
	b	T	[✓ F]	If two heuristics, h1 and h2, are admissible then their sum is admissible.
4	c	[✓ T]	F	Breadth-first search is complete (is guaranteed to find a solution) as long as the state space has finite branching factor.
	d	[✓ T]	F	Higher values for the discount factor, γ, will, in general, cause value iteration to converge more slowly.
2	e	T	[✓ F]	A lower value for the discount factor, γ, will cause the agent to focus on temporally distant (long term) rewards.
–	f	[✓ T]	F	The purpose of labeling in the LRTDP algorithm is to recognize when a trial returns to a previously visited state.
+	g	[✓ T]	F	If a reinforcement agent's learning rate, α, is increased its policy will more heavily weight recent experience compared to older experience.
	h	[✓ T]	F	Particle filters are often a preferred method for solving HMMs with large or continuous state spaces.
1	i	T	F	Cumulative regret is a better choice than simple regret for balancing the exploration / exploitation tradeoff when training in a simulator.
1	j	T	F	UCB is an optimal (within constant factors) strategy for balancing the exploration / exploitation tradeoff when maximizing simple regret.
2	k	[✓ T]	F	A heuristic is `admissible' if, when applied to a partial solution, it never overestimates the cost of a full solution that is reachable from the partial solution. (Equivalently, never underestimates the utility of such a solution).

0	l	T	[✓ F]	Consider a POMDP = {S, A, T, R,ɣ, Z, O} and let B = <{s_1, … s_n}, {p_1, …, p_n}> be a belief state, where s_i has probability p_i in the distribution. Let H(B) = Max_i V(s_i), where V(s_i) is the value of state s_i in the corresponding MDP = {S, A, T, R, ɣ}. **Claim**: H is admissible.
2	m	T	[✓ F]	If events A and B are independent (A ⊥ B), then they are also conditionally independent on any event C (A ⊥ B | C)
1	n	T	F	If A ⊥ B | C and B ⊥ C | A, then C ⊥ A | B
1	o	T	F	Adding edges to a Bayes net allows it to encode a larger space of possibilities
0	p	[✓ T]	F	When initialized with an admissible heuristic, approximate Q-learning with feature vectors will always converge to the optimal policy
1	q	T	F	Stochastic shortest path problems (SSPs) are strictly more expressive than infinite-horizon, discounted reward MDPs.
2	r	T	[✓ F]	The transition probabilities in an HMM may change over time.
2	s	[✓ T]	F	Policy iteration will converge to the optimal policy regardless of how it is initialized.

~~A⊥B⊥C.~~

2) Bayes Nets. (4 points). Draw the edges in a Bayes net over the four random variables shown to the right, that make the following independence assumptions: A ⊥ D | C and B ⊥ D | C (where we are using ⊥ to denote independence). The network must not make *any* additional independence assumptions. If no such network is possible, write "impossible." If no arrows are needed, write "none needed."

A⊥B -1



3) HMMs. (6 points) Consider the HMM specified with the state space S = {U, D, C} and the observations O = {H, L}. Let P(s_0 = C) = 1.0. Compute the probabilities P(s | o) given the transition and emission probabilities below.

Need to normalize -2

x=	U	D	C
P(s_1 = x | o_1 = H)	0.24	0.06	?? 0.32 0.2
P(s_2 = x | o_1 = H)	0.1304	0.024	0.0893.

->	U	D	C
U	0.8	0	0.2
D	0	0.4	0.6
C	0.3	0.3	0.4

	H	L
U	0.8	0.2
D	0.2	0.8
C	0.5	0.5

0.048
0.0036
0.128

0.1796
0.0893

B(s_1 = U) = 0.3.
B(s_1 = U | o_1 = H) = 0.24

+1 Marginalization idea

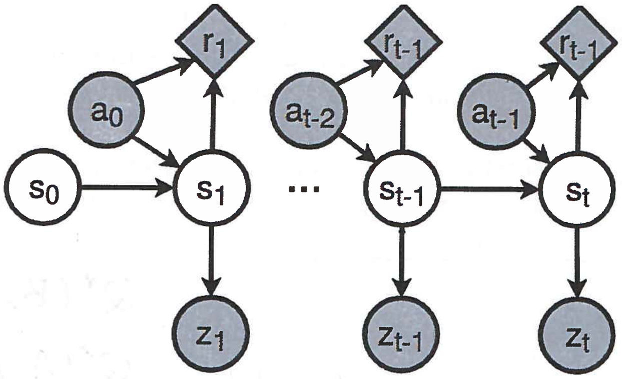
$0.24 \times 0.8 \times 0.8 = 0.192 \times 0.8 = 0.1536$
$0.32 \times 0.3 \times 0.8 = 0.096 \times 0.8 = 0.0768$

?

close here but ~~[illegible]~~ why x0.8?

0.024 x 0.[illegible] 0.0.44
~~0.0192~~
0.096 x 0.2 = 0.0192
0.240

3) POMDPS. Dynamic Bayes nets are often used for a compact description of Markovian models (HMM, MDP, POMDP, ...). For example, consider the following DBN model of a new variant of POMDP. Here, the transition function is T(s, a, s'), T: |S| * |A| * |S| -> [0,1] where s is the *start state* and s' is the *end state* of action a, and is specified as the conditional probability table for each of the s_i nodes in the network (for i>0).



a) (2 points) Which of the following best describes the DBN's model of the Boolean observation function?

w) O(s, z), O:|S| * |Z| -> [0,1] where s is the *start state* and z is the observation

x) O(s, z), O:|S| * |Z| -> [0,1] where s is the *end state* and z is the observation

y) O(s, z, a), O:|S| * |Z| * |A| -> [0,1] where s is the *start state*, z is the observation, and a is the action

z) O(s, z, a), O:|S| * |Z| * |A| -> [0,1] where s is the *end state*, z is the observation, and a is the action

Choose one	
	w
	x
✓	y
✓	z

0

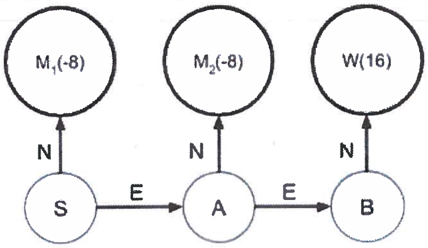
b) (2 points) Assuming the agent is at step *t*, its belief state b_t represent the posterior probability of each state, i.e. $b_t(s)=P(s|a_0, z_1, a_1,...a_{t-1}, z_t)$. Describe the expected instant reward of the agent if it performs action a_t using general POMDP model parameters such as b_t, a_t, z_t, O, T, R, and γ.

$$R(s, a, \overset{s'}{z\cancel{x}}) = \sum_t b_t(s) \cdot T(s, a_t, s') O(\cancel{s'}, a_t, z_t)$$

$R(a_t, s')$

1

4) MDPs and Reinforcement Learning. Your robot uses the MDP shown to the right in order to represent navigation on Mars. Nasty Martians live in terminal states M1 and M2 and useful water can be found in terminal state W; their rewards are shown inside parentheses. Other states (S, A, and B) provide no reward. The robot may only travel N or E (and from state B it may only go N). only actions it can take is to either go East (E) or go North (N) (it can only go North from state B). SInce you are an awesome robotic engineer, there is no noise when transitioning to different states and actions always result in the intended motion.



a) (4 points) What are the optimal q-values for the state S if discount rate, ɣ=0.5. Please put your answers in the boxes, and show your work below.

Q*(S,N) =	-8
Q*(S, E) =	4

4

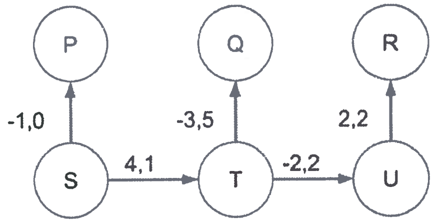
$Q^*(B,N) = 16$

$Q^*(A,N) = -8$

$Q^*(A,E) = 8.$

$Q^*(S,E) = 4$

Alas, the spaceship engineers weren't so good and your robot is accidentally landed on Titan instead, and damaged in the process, so the effectors are no longer calibrated; hence, the transition and reward functions are unknown. Luckily, you implemented approximate Q-learning as a backup in case your robot found itself in an unfamiliar environment. Specifically, you chose to approximate the Q-function as a linear combination of two features, f_1 and f_2, whose values are shown on the edges in the diagram to the right. For example, $f_1(S, N) = -1$, $f_2(S, N)= 0$, and $f_1(T, N) = -3$.



b) (8 points) Use approximate Q-learning to compute the corresponding weights, w_1 and w_2, for the following (partial) trial. Use a learning rate of α=0.5 and a discount factor of ɣ= 1.0. Assume all weights are initially zero.

i) Start State: S, Action: E, End State: T, Reward: 8

$Q = w_1 f_1 + w_2 f_2,$

$Q(S,E) = 0.$

$Q(S,E) \leftarrow (1-0.5)\,Q(S,E) + 0.5(8 + 1\cdot 0). = 4.$

$w_1 \leftarrow 0 + 0.5[(8 + 1\cdot 0) - 4]\cdot 4 = 8.$

$w_2 \leftarrow 0 + 0.5[(8+1\cdot 0) - 4]\cdot 1 = 2.$

$\therefore \begin{cases} w_1 = 8 \\ w_2 = 2. \end{cases}$

3

ii) Start State: T, Action: E, End State: U, Reward: 12

$Q(U,N) = 8\cdot 2 + 2\cdot 2 = 20.$

$Q(T,E) = 8\cdot -2 + 2\cdot 2 = -12.$

$Q(T,E) \leftarrow (1-0.5)\,Q(T,E) + 0.5(12 + 1\cdot 20) = -6 + 16 = 10.$

$w_1 \leftarrow 8 + 0.5[(12 + 1\cdot 20) - 10]\cdot -2 = 8 + 0.5\cdot 22\cdot -2 = -14$

$w_2 \leftarrow 2 + 0.5[(12 + 1\cdot 20) - 10]\cdot 2 = 2 + 0.5\cdot 22\cdot 2 = 24.$

2

$\therefore \begin{cases} w_1 = -14 \\ w_2 = 24 \end{cases}$