DATA130008 Spring 2017 | Introduction to Artificial Intelligence

Final Quiz

- You have approximately 1 hour and 20 minutes.
- The exam is closed book, closed notes except your one-page paper in A4 size.
- Answers need to be provided in English.
- 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.

First name	
Last name	
SID	

For staff use only:

Q1.	Short Questions	/15
Q2.	Variants of Trees	/20
Q3.	CSPs	/20
Q4.	Finding Waldo	/15
Q5.	Instantiated Elimination	/15
Q6.	Argg! Sampling for the Legendary Treasure	/15
	Total	/100

Q1. [15 pts] Short Questions

Circle your answer.

(a) [5 pts] **Search**

(i) [*true* or *<u>false</u>*] Uniform-cost search will never expand more nodes than A*-search.

(ii) [*true* or *<u>false</u>*] Depth-first search will always expand more nodes than breadth-first search.

(iii) [*<u>true</u>* or *false*] The heuristic $h(n) = 0$ is admissible for every search problem.

(iv) [*true* or *<u>false</u>*] The heuristic $h(n) = 1$ is admissible for every search problem.

(v) [*<u>true</u>* or *false*] The heuristic $h(n) = c(n)$, where $c(n)$ is the true cheapest cost to get from the node n to a goal state, is admissible for every search problem.

(b) [2 pts] **CSPs**

(i) [*<u>true</u>* or *false*] The most-constrained variable heuristic provides a way to select the next variable to assign in a backtracking search for solving a CSP.

(ii) [*true* or *<u>false</u>*] By using the most-constrained variable heuristic and the least-constraining value heuristic we can solve every CSP in time linear in the number of variables.

(c) [3 pts] **Games**

(i) [*true* or *<u>false</u>*] When using alpha-beta pruning, it is possible to get an incorrect value at the root node by choosing a bad ordering when expanding children.

(ii) [*true* or *<u>false</u>*] When using alpha-beta pruning, the computational savings are independent of the order in which children are expanded.

(iii) [*true* or *<u>false</u>*] When using expectimax to compute a policy, re-scaling the values of all the leaf nodes by multiplying them all with 10 can result in a different policy being optimal.

(d) [3 pts] **MDPs** For this question, assume that the MDP has a finite number of states.

(i) [*true* or *<u>false</u>*] For an MDP (S, A, T, γ, R) if we only change the reward function R the optimal policy is guaranteed to remain the same.

(ii) [*<u>true</u>* or *false*] Value iteration is guaranteed to converge if the discount factor (γ) satisfies $0 < \gamma < 1$.

(iii) [*true* or *<u>false</u>*] Policies found by value iteration are superior to policies found by policy iteration.

(e) [2 pts] **Reinforcement Learning**

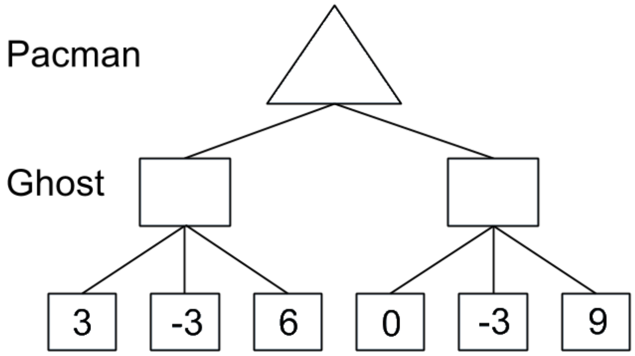
(i) [*<u>true</u>* or *false*] Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.

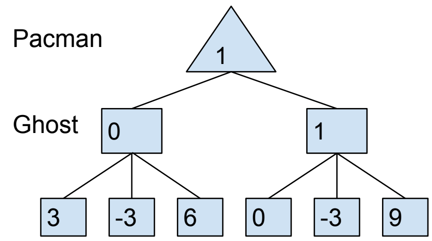
(ii) [*true* or *<u>false</u>*] If an MDP has a transition model T that assigns non-zero probability for all triples $T(s, a, s')$ then Q-learning will fail.

Q2. [20 pts] Variants of Trees

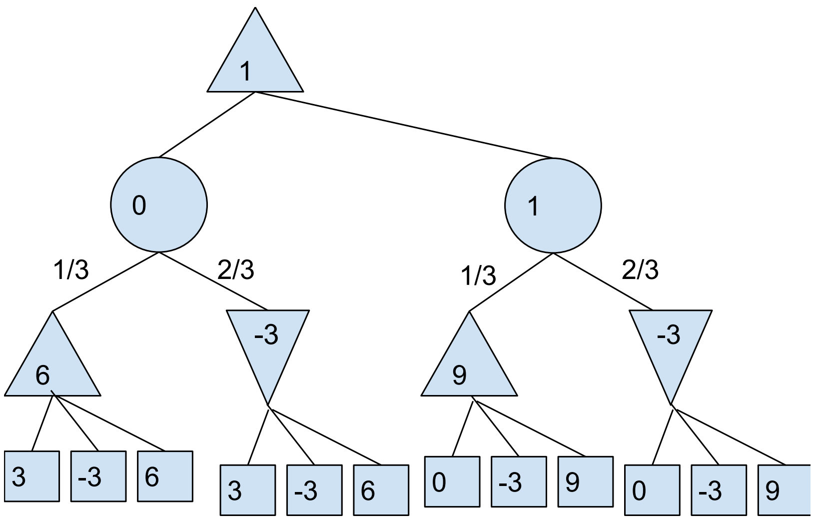
(a) Pacman is going to play against a careless ghost, which makes a move that is optimal for Pacman $\frac{1}{3}$ of the time, and makes a move that that minimizes Pacman's utility the other $\frac{2}{3}$ of the time.

(i) [3 pts] Fill in the correct utility values in the game tree below where Pacman is the maximizer:

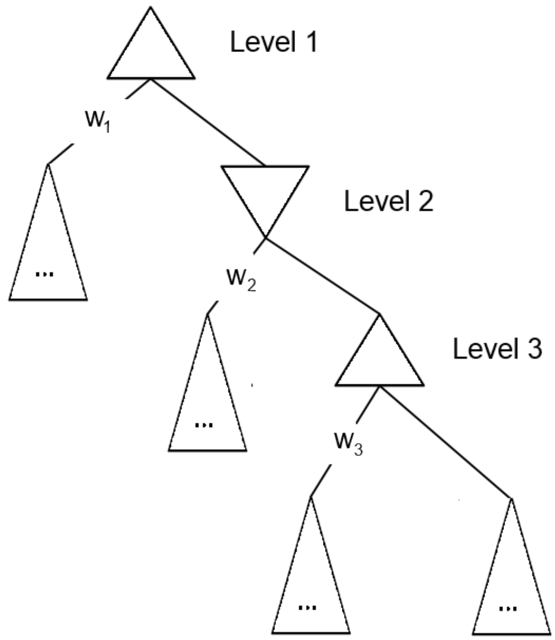




(ii) [3 pts] Draw a complete game tree for the game above that contains only max nodes, min nodes, and chance nodes.



(b) Consider a modification of alpha-beta pruning where, rather than keeping track of a single value for α and β, you instead keep a list containing the best value, w_i, for the minimizer/maximizer (depending on the level) at each level up to and including the current level. Assume that the root node is always a max node. For example, consider the following game tree in which the first 3 levels are shown. When the considering the right child of the node at level 3, you have access to w_1, w_2, and w_3.



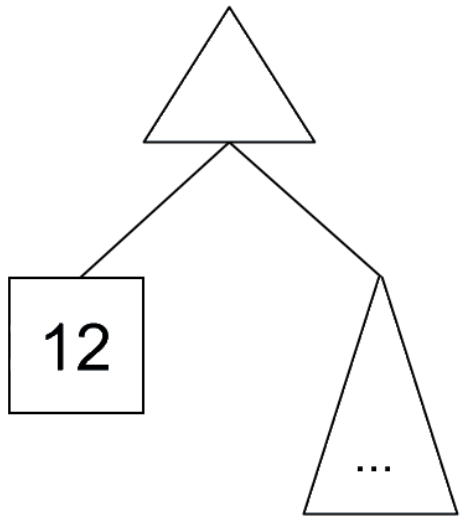
(i) [2 pts] Under this new scenario, what is the pruning condition for a max node at the n^{th} level of the tree (in terms of v and $w_1...w_n$)? $v > min_i(w_i)$, where i is even;

(ii) [2 pts] What is the pruning condition for a min node at the n^{th} level of the tree?
$v < max_i(w_i)$, where i is odd;

(iii) [3 pts] What is the relationship between α, β and the list of $w_1...w_n$ at a max node at the n^{th} level of the tree?

- ○ $\sum_i w_i = \alpha + \beta$
- ○ $\max_i w_i = \alpha, \quad \min_i w_i = \beta$
- ○ $\min_i w_i = \alpha, \quad \max_i w_i = \beta$
- ○ $w_n = \alpha, \quad w_{n-1} = \beta$
- ○ $w_{n-1} = \alpha, \quad w_n = \beta$
- ○ None of the above. The relationship is $\beta = min(w_2, w_4, w_6...), \quad \alpha = max(w_1, w_3, w_5...)$

(c) Pacman is in a dilemma. He is trying to maximize his overall utility in a game, which is modeled as the following game tree.



The left subtree contains a utility of 12. The right subtree contains an unknown utility value. An oracle has told you that the value of the right subtree is one of -3, -9, or 21. You know that each value is equally likely, but without exploring the subtree you do not know which one it is.

Now Pacman has 3 options:

1. Choose left;
2. Choose right;
3. Pay a cost of c = 1 to explore the right subtree, determine the exact utility it contains, and then make a decision.

(i) [3 pts] What is the expected utility for option 3? $12 * \frac{2}{3} + 21 * \frac{1}{3} - 1 = 14$

(ii) [4 pts] For what values of c (for example, $c > 5$ or $-2 < c < 2$) should Pacman choose option 3? If option 3 is never optimal regardless of the value for c, write None.
According to the previous part, we have $15 - c > 12$, then we will choose option 3.
Therefore, $c < 3$

Q3. [20 pts] CSPs

(a) [7 pts] **Pacman's new house**

After years of struggling through mazes, Pacman has finally made peace with the ghosts, Blinky, Pinky, Inky, and Clyde, and invited them to live with him and Ms. Pacman. The move has forced Pacman to change the rooming assignments in his house, which has 6 rooms. He has decided to figure out the new assignments with a CSP in which the variables are Pacman **(P)**, Ms. Pacman **(M)**, Blinky **(B)**, Pinky **(K)**, Inky **(I)**, and Clyde **(C)**, the values are which room they will stay in, from 1-6, and the constraints are:

i) No two agents can stay in the same room
ii) **P** > 3
iii) **K** is less than **P**
iv) **M** is either 5 or 6
v) **P** $>$ **M**
vi) **B** is even
vii) **I** is not 1 or 6
viii) $|\mathbf{I}\text{-}\mathbf{C}| = 1$
ix) $|\mathbf{P}\text{-}\mathbf{B}| = 2$

(i) [1 pt] **Unary constraints** On the grid below cross out the values from each domain that are eliminated by enforcing unary constraints.

P	~~1~~	~~2~~	~~3~~	4	5	6
B	~~1~~	2	~~3~~	4	~~5~~	6
C	1	2	3	4	5	6
K	1	2	3	4	5	6
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	5	6

The unary constraints are ii, iv, vi, and vii. ii crosses out 1,2, and 3 for P. iv crosses out 1,2,3,4 for M. vi crosses out 1,3, and 5 for B. vii crosses out 1 and 6 for I. K and C have no unary constraints, so their domains remain the same.

(ii) [1 pt] **MRV** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned to first?

○ P ○ B ○ C ○ K ○ I ● M

M has the fewest value remaining in its domain (2), so it should be selected first for assignment.

(iii) [2 pts] **Forward Checking** For the purposes of decoupling this problem from your solution to the previous problem, assume we choose to assign P first, and assign it the value 6. What are the resulting domains after enforcing unary constraints (from part i) and running forward checking for this assignment?

P						6
B	~~1~~	~~2~~	~~3~~	4	~~5~~	~~6~~
C	1	2	3	4	5	~~6~~
K	1	2	3	4	5	~~6~~
I	~~1~~	2	3	4	5	~~6~~
M	~~1~~	~~2~~	~~3~~	~~4~~	5	~~6~~

In addition to enforcing the unary constraints from part i, the domains are further constrained by all constraints involving P. This includes constraints i, iii, v, and ix. i removes 6 from the domains of all variables. iii removes 6 from the domain of K (already removed by constraint i). v removes 6 from the domain of M (also already removed by i). ix removes 2 and 6 from the domain of B.

(iv) [3 pts] **Iterative Improvement** Instead of running backtracking search, you decide to start over and run iterative improvement with the min-conflicts heuristic for value selection. Starting with the following assignment:

P:6, B:4, C:3, K:2, I:1, M:5

First, for each variable write down how many constraints it violates in the table below.
Then, in the table on the right, for all variables that could be selected for assignment, put an x in any box

that corresponds to a possible value that could be assigned to that variable according to min-conflicts. When marking next values a variable could take on, only mark values different from the current one.

Variable	# violated
P	0
B	0
C	1
K	0
I	2
M	0

	1	2	3	4	5	6
P						
B						
C		x				
K						
I		x		x		
M						

Both I and C violate constraint viii, because |I-C|=2. I also violates constraint vii. No other variables violate any constraints. According to iterative improvement, any conflicted variable could be selected for assignment, in this case I and C. According to min-conflicts, the values that those variables can take on are the values that minimize the number of constraints violated by the variable. Assigning 2 or 4 to I causes it to violate constraint i, because other variables already have the values 2 and 4. Assigning 2 to C also only causes C to violate 1 constraint.

(b) [5 pts] **Variable ordering**

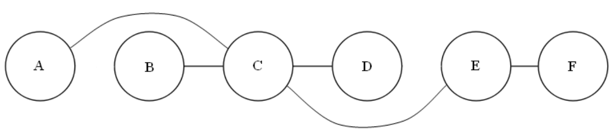
We say that a variable X is backtracked if, after a value has been assigned to X, the recursion returns at X without a solution, and a different value must be assigned to X.

For this problem, consider the following three algorithms:

1. Run backtracking search with no filtering
2. Initially enforce arc consistency, then run backtracking search with no filtering
3. Initially enforce arc consistency, then run backtracking search while enforcing arc consistency after each assignment

(i) [5 pts]

For each algorithm, circle all orderings of variable assignments that guarantee that no backtracking will be necessary when finding a solution to the CSP represented by the following constraint graph.



Algorithm 1	Algorithm 2	Algorithm 3
A-B-C-D-E-F	A-B-C-D-E-F	$\boxed{\text{A-B-C-D-E-F}}$
F-E-D-C-B-A	F-E-D-C-B-A	$\boxed{\text{F-E-D-C-B-A}}$
C-A-B-D-E-F	$\boxed{\text{C-A-B-D-E-F}}$	$\boxed{\text{C-A-B-D-E-F}}$
B-D-A-F-E-C	B-D-A-F-E-C	$\boxed{\text{B-D-A-F-E-C}}$
D-E-F-C-B-A	D-E-F-C-B-A	$\boxed{\text{D-E-F-C-B-A}}$
B-C-D-A-E-F	$\boxed{\text{B-C-D-A-E-F}}$	$\boxed{\text{B-C-D-A-E-F}}$

Algorithm 1:
No filtering means that there are no guarantees that an assignment to one variable has consistent assignments in any other variable, so backtracking may be necessary.
Algorithm 2:
This algorithm is very similar to the tree-structured CSP algorithm presented in class, in which arcs are enforced from one right to left, and then variables are assigned from left to right. The arcs enforced in that algorithm are a subset of all arcs enforced when enforcing arc consistency. Thus, any linear ordering of variables in which each variable is assigned before all of its children in the tree will guarantee no backtracking.
Algorithm 3:
Any first assignment can be the root of a tree, which, from class, we know is consistent and will not require backtracking. This assignment can then be viewed as conditioning the graph on that variable, and after re-running arc consistency, it can be removed from the graph. This results in either one or two tree-structured graphs that are also arc consistent, and the process can be repeated.

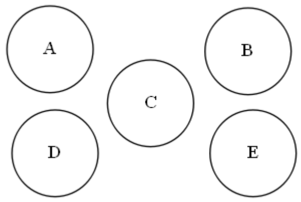
(c) [8 pts] **All Satisfying Assignments** Now consider a modified CSP in which we wish to find every possible satisfying assignment, rather than just one such assignment as in normal CSPs. In order to solve this new problem, consider a new algorithm which is the same as the normal backtracking search algorithm, except that when it sees a solution, instead of returning it, the solution gets added to a list, and the algorithm backtracks. Once there are no variables remaining to backtrack on, the algorithm returns the list of solutions it has found.

For each graph below, select whether or not using the MRV and/or LCV heuristics could affect the number of nodes expanded in the search tree in this new situation.

The remaining parts all have a similar reasoning. Since every value has to be checked regardless of the outcome of previous assignments, the order in which the values are checked does not matter, so LCV has no effect.
In the general case, in which there are constraints between variables, the size of each domain can vary based on the order in which variables are assigned, so MRV can still have an effect on the number of nodes expanded

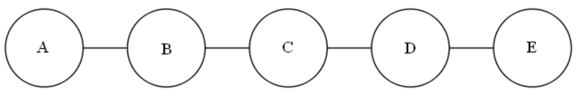
for the new ”find all solutions” task.
The one time that MRV is guaranteed to not have any effect is when the constraint graph is completely disconnected, as is the case for part i. In this case, the domains of each variable do not depend on any other variable's assignment. Thus, the ordering of variables does not matter, and MRV cannot have any effect on the number of nodes expanded.

(i) [2 pts]



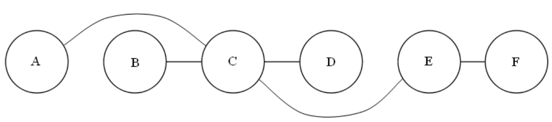
- ● Neither MRV nor LCV can have an effect.
- ○ Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

(ii) [2 pts]



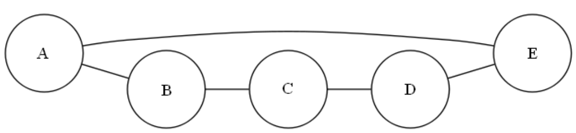
- ○ Neither MRV nor LCV can have an effect.
- ● Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

(iii) [2 pts]



- ○ Neither MRV nor LCV can have an effect.
- ● Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

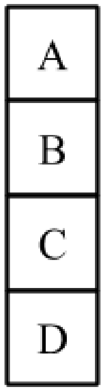
(iv) [2 pts]



- ○ Neither MRV nor LCV can have an effect.
- ● Only MRV can have an effect.
- ○ Only LCV can have an effect .
- ○ Both MRV and LCV can have an effect.

Q4. [15 pts] Finding Waldo

You are part of the Search Team to find Waldo. Waldo randomly moves around floors A, B, C, and D. Waldo's location at time t is X_t. At the end of each timestep, Waldo stays on the same floor with probability 0.5, goes upstairs with probability 0.3, and goes downstairs with probability 0.2. If Waldo is on floor A, he goes down with probability 0.2 and stays put with probability 0.8. If Waldo is on floor D, he goes upstairs with probability 0.3 and stays put with probability 0.7.



X_0	$P(X_0)$
A	0.1
B	0.2
C	0.3
D	0.4

(a) [2 pts] Fill in the table below with the distribution of Waldo's location at time $t = 1$.

X_t	$P(X_1)$
A	$0.1 * 0.8 + 0.2 * 0.3 = 0.14$
B	$0.2 * 0.5 + 0.1 * 0.2 + 0.3 * 0.3 = 0.21$
C	$0.3 * 0.5 + 0.4 * 0.3 + 0.2 * 0.2 = 0.31$
D	$0.4 * 0.7 + 0.3 * 0.2 = 0.34$

(b) [4 pts] $F_T(X)$ is the fraction of timesteps Waldo spends at position X from $t = 0$ to $t = T$. The system of equations to solve for $F_\infty(A)$, $F_\infty(B)$, $F_\infty(C)$, and $F_\infty(D)$ is below. Fill in the blanks.

$$\underline{0.8}\ F_\infty(A) + \underline{0.3}\ F_\infty(B) + \underline{0}\ F_\infty(C) + \underline{0}\ F_\infty(D) = F_\infty(A)$$

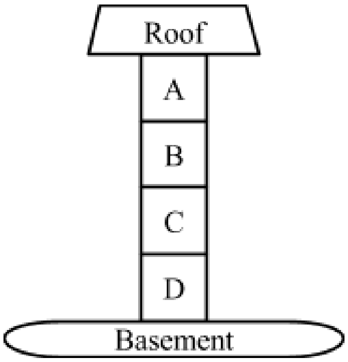
$$\underline{0.2}\ F_\infty(A) + \underline{0.5}\ F_\infty(B) + \underline{0.3}\ F_\infty(C) + \underline{0}\ F_\infty(D) = F_\infty(B)$$

$$\underline{0}\ F_\infty(A) + \underline{0.2}\ F_\infty(B) + \underline{0.5}\ F_\infty(C) + \underline{0.3}\ F_\infty(D) = F_\infty(C)$$

$$\underline{0}\ F_\infty(A) + \underline{0}\ F_\infty(B) + \underline{0.2}\ F_\infty(C) + \underline{0.7}\ F_\infty(D) = F_\infty(D)$$

$$\underline{1}\ F_\infty(A) + \underline{1}\ F_\infty(B) + \underline{1}\ F_\infty(C) + \underline{1}\ F_\infty(D) = 1$$

To aid the search a sensor S_r is installed on the roof and a sensor S_b is installed in the basement. Both sensors detect either sound ($+s$) or no sound ($-s$). The distribution of sensor measurements is determined by d, the number of floors between Waldo and the sensor. For example, if Waldo is on floor B, then $d_b = 2$ because there are two floors (C and D) between floor B and the basement and $d_r = 1$ because there is one floor (A) between floor B and the roof. The prior of the both sensors' outputs are identical and listed below. **Waldo will not go onto the roof or into the basement.**



X_0	$P(X_0)$
A	0.1
B	0.2
C	0.3
D	0.4

S_r	$P(S_r \mid d_r)$
$+s$	$0.3 * d_r$
$-s$	$1 - 0.3 * d_r$

S_b	$P(S_b \mid d_b)$
$+s$	$1 - 0.3 * d_b$
$-s$	$0.3 * d_b$

S	$P(S)$
$+s$	0.5
$-s$	0.5

(c) You decide to track Waldo by particle filtering with 3 particles.

(i) [2 pts] At time $t = 2$, the particles are at positions $X_1 = A$, $X_2 = B$ and $X_3 = C$. Without incorporating any sensory information, what is the probability that the particles will be resampled as $X_1 = B$, $X_2 = B$, and $X_3 = C$, after time elapse?

Answer: $P(X_3 = B \mid X_2 = A)P(X_3 = B \mid X_2 = B)P(X_3 = C \mid X_2 = C)$
$= (0.2)(0.5)(0.5) = 0.05$

(ii) [2 pts] Assume the particles after time elapsing are $X_1 = B$, $X_2 = C$, $X_3 = D$, and the sensors observe $S_r = +s$ and $S_b = -s$.

What are the particle weights given these observations?

Particle	Weight
$X_1 = B$	$P(S_r = +s \mid d_r = 1)P(S_b = -s \mid d_b = 2) = 0.18$
$X_2 = C$	$P(S_r = +s \mid d_r = 2)P(S_b = -s \mid d_b = 1) = 0.18$
$X_3 = D$	$P(S_r = +s \mid d_r = 3)P(S_b = -s \mid d_b = 0) = 0$

(iii) [2 pts] Assume the particles after time elapsing are $X_1 = B$, $X_2 = C$, $X_3 = D$, and the particle weights in the following table. What is the probability the particles will be resampled as $X_1 = B$, $X_2 = B$, and $X_3 = D$?

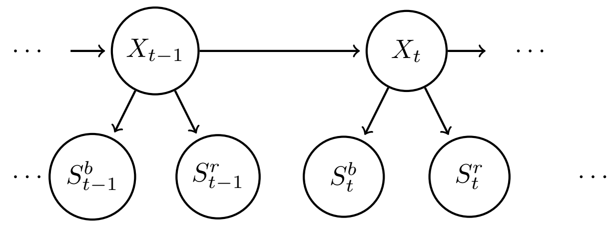
Particle	Weight
$X = B$	0.1
$X = C$	0.6
$X = D$	0.3

$0.1 * 0.1 * 0.3 = 0.003$

(d) [3 pts] **Note: the r and b subscripts from before will be written here as superscripts.**

Part of the expression for the forward algorithm update for Hidden Markov Models is given below. $s^r_{0:t}$ are all the measurements from the roof sensor $s^r_0, s^r_1, s^r_2, \ldots, s^r_t$. $s^b_{0:t}$ are all the measurements from the roof sensor $s^b_0, s^b_1, s^b_2, \ldots, s^b_t$.

Which of the following are correct completions of line (4)? Circle all that apply.



$$
\begin{aligned}
P(x_t|s^r_{0:t}, s^b_{0:t}) &\propto P(x_t, s^r_{0:t}, s^b_{0:t}) && (1)\\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, s^r_{0:t}, s^b_{0:t}) && (2)\\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, s^r_{0:t-1}, s^r_t, s^b_{0:t-1}, s^b_t) && (3)\\
&= \sum_{x_{t-1}} \underline{\hspace{6cm}} P(x_t|x_{t-1}) P(x_{t-1}, s^r_{0:t-1}, s^b_{0:t-1}) && (4)
\end{aligned}
$$

- ● $P(s^r_t, s^b_t|x_{t-1}, x_t, s^r_{0:t-1}, s^b_{0:t-1})$
- ● $P(s^r_t|x_t)P(s^b_t|x_t)$
- ○ $P(s^r_t|x_{t-1})P(s^b_t|x_{t-1})$
- ○ $P(s^r_t|s^r_{t-1})P(s^b_t|s^b_{t-1})$
- ● $P(s^r_t, s^b_t|x_t)$
- ● $P(s^r_t, s^b_t|x_t, x_{t-1})$
- ○ None of the above.

There are two equally-correct interpretations of this question: (1) completing the mathematical expression for the probability and (2) completing the algorithmic update for the probability.

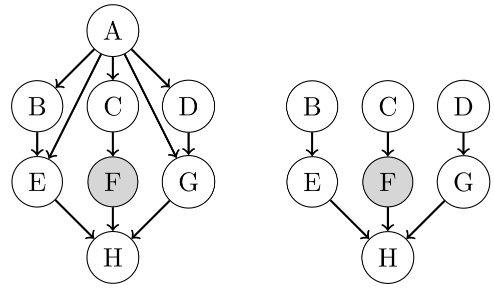
Selecting the answers above is correct for interpretation (1): in the Hidden Markov Model, these four probabilities are identical.

Selecting answer 2 alone is correct for interpretation (2): in the Hidden Markov Model, the forward algorithm has the conditional probabilities of the observations given the present state. While the other three choices above are mathematically equivalent, they are not available to the algorithm during execution.

Both correct interpretations earned full credit.

Q5. [15 pts] Instantiated Elimination

(a) Difficulty of Elimination. Consider answering $P(H \mid +f)$ by variable elimination in the Bayes' nets N and N'.
Elimination order is alphabetical.
All variables are binary $+/-$.
Factor size is the number of unobserved variables in a factor made during elimination.



N **N′**

(i) [2 pts] What is the size of the largest factor made during variable elimination for N?

5

$F(B, C, D, E, G)$ after eliminating A. 2^5, the number of factor entries was also accepted.

(ii) [2 pts] What is the size of the largest factor made during variable elimination for N'?

2

$F(G, H, +f)$ after eliminating E.

Variable elimination in N can take a lot of work! If only A were observed...

(b) Instantiation Sets. To simplify variable elimination in N, let's pick an *instantiation set* to pretend to observe, and then do variable elimination with these additional instantiations.

Consider the original query $P(H \mid +f)$, but let A be the instantiation set so $A = a$ is observed. Now the query is H with observations $F = +f, A = a$.

(i) [2 pts] What is the size of the largest factor made during variable elimination with the $A = a$ instantiation? 2

Observing $A = a$ renders the children conditionally independent and prevents its elimination. The large factor $F(B, C, D, E, G)$ is never made and elimination proceeds similarly to elimination in N' except for the presence of a in the factors.

(ii) [2 pts] Given a Bayes' net over n binary variables with k variables chosen for the instantiation set, how many instantiations of the set are there?

2^k

For a selected instantiation set of k binary variables, there are 2^k total instantiations of these variables. The alternative interpretation of how many size k instantiation sets exist from n variables, $\binom{n}{k}$ was also accepted.

(c) Inference by Instantiation. Let's answer $P(H| + f)$ by variable elimination with the instantiations of A.

(i) [2 pts] What quantity does variable elimination for $P(H| + f)$ with the $A = +a$ instantiation compute *without normalization*? That is, which choices are equal to the entries of the last factor made by elimination?

○ $P(H \mid +f)$	● $P(H, +a, +f)$	○ $P(H, +f \mid +a)$
○ $P(H \mid +a)$	○ $P(H, +a \mid +f)$	○ $P(H \mid +a, +f)$

At the end of variable elimination, the last factor is equal to the equivalent entries of the joint distribution with the eliminated variables summed out and the selected values of the evidence variables. Normalization gives the conditional $p(+h \mid +a, +f) = \frac{f(+h,+a,+f)}{f(+h,+a,+f)+f(-h,+a,+f)}$, but here the factor is kept unnormalized.

(ii) [2 pts] Let $I_+(H) = F(H, +a, +f)$ and $I_-(H) = F(H, -a, +f)$ be the last factors made by variable elimination with instantiations $A = +a$ and $A = -a$. Which choices are equal to $p(+h \mid +f)$?

○ $I_+(+h) \cdot p(+a) \cdot I_-(+h) \cdot p(-a)$

○ $I_+(+h) \cdot p(+a) + I_-(+h) \cdot p(-a)$

○ $I_+(+h) + I_-(+h)$

○ $\frac{I_+(+h) \cdot p(+a) \cdot I_-(+h) \cdot p(-a)}{\sum_h I_+(h) \cdot p(+a) \cdot I_-(h) \cdot p(-a)}$

○ $\frac{I_+(+h) \cdot p(+a) + I_-(+h) \cdot p(-a)}{\sum_h I_+(h) \cdot p(+a) + I_-(h) \cdot p(-a)}$

● $\frac{I_+(+h) + I_-(+h)}{\sum_h I_+(h) + I_-(h)}$

The last factors are the entries from the corresponding joint $I_+(+h) = p(+h, +a, +f)$ and $I_-(+h) = p(+h, -a, +f)$ and so on.

By the law of total probability $p(+h, +f) = p(+h, +a, +f) + p(+h, -a, +f)$, so the joint of the original query and evidence can be computed from the instantiated elimination factors. For the conditional $p(+h, +f)$, normalize by the sum over the query $\sum_h p(h, +f) = \sum_h \sum_a p(h, a, +f) = f(+h, +a, +f) + f(+h, -a, +f) + f(-h, +a, +f) + f(-h, -a, +f)$ where the joint over the query and evidence is again computed from the law of total probability over A.

Working with the joint in this way is why the last factor of instantiated elimination was left unnormalized. (Those who answered the conditional $P(H \mid +a, +f)$ in the previous part were awarded credit for multiplying the marginal $p(+a), p(-a)$ back in as this is the correct chain rule were $P(H \mid +a, +f)$ correct.)

(d) [3 pts] **Complexity of Instantiation.** What is the time complexity of instantiated elimination? Let n = number of variables, k = instantiation set size, f = size of the largest factor made by elimination without instantiation, and i = size of the largest factor made by elimination with instantiation. Mark the tightest bound. Variable elimination without instantiation is $O(n \exp(f))$.

○ $O(n \exp(k))$

○ $O(n \exp(i))$

● $O(n \exp(i + k))$

○ $O(n \exp(f))$

○ $O(n \exp(f - k))$

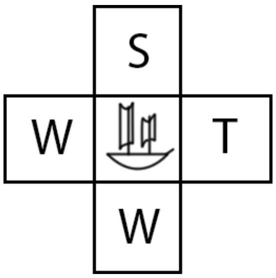
○ $O(n \exp(i/f))$

To carry out instantiated elimination, we have to do variable elimination $\exp(k)$ times for all the settings of the instantiation set. Each of these eliminations takes time bounded by $n \exp(i)$ as the largest factor is the most expensive to eliminate and there are n variables to eliminate.

If the instantiation set is not too large and the size of the factors made by instantiation elimination are small enough this method can be exponentially faster than regular elimination. The catch is how to select the instantiation set.

Q6. [15 pts] Argg! Sampling for the Legendary Treasure

Little did you know that Michael and John are actually infamous pirates. One day, they go treasure hunting in the Ocean of Bayes, where rumor says a great treasure lies in wait for explorers who dare navigate in the rough waters. After navigating about the ocean, they are within grasp of the treasure. Their current configuration is represented by the boat in the figure below. They can only make one move, and must choose from the actions: (North, South, East, West). Stopping is not allowed. They will land in either a whirlpool (W), an island with a small treasure (S), or an island with the legendary treasure (T). The utilities of the three types of locations are shown below:

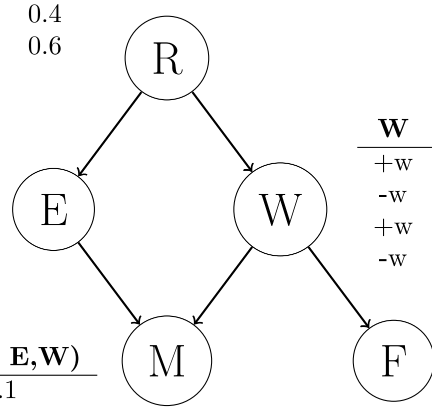


State	**U(State)**
T (Legendary Treasure)	100
S (Small Treasure)	25
W (Whirlpool)	-50

The success of their action depends on the random variable **Movement (M)**, which takes on one of two values: (+m, -m). The Movement random variable has many relationships with other variables: Presence of Enemy Pirates (E), Rain (R), Strong Waves (W), and Presence of Fishermen (F). The Bayes' net graph that represents these relationships is shown below:

R	**P(R)**
+r	0.4
-r	0.6

E	**R**	**P(E | R)**
+e	+r	0.3
-e	+r	0.7
+e	-r	0.6
-e	-r	0.4



W	**R**	**P(W | R)**
+w	+r	0.9
-w	+r	0.1
+w	-r	0.2
-w	-r	0.8

M	**E**	**W**	**P(M | E,W)**
+m	+e	+w	0.1
-m	+e	+w	0.9
+m	+e	-w	0.45
-m	+e	-w	0.55
+m	-e	+w	0.35
-m	-e	+w	0.65
+m	-e	-w	0.9
-m	-e	-w	0.1

F	**W**	**P(F | W)**
+f	+w	0.15
-f	+w	0.85
+f	-w	0.75
-f	-w	0.25

In the following questions we will follow a two-step process:
– (1) Michael and John observed the random variables $R = -r$ and $F = +f$. We then determine the distribution for $P(M| -r, +f)$ via sampling.

– (2) Based on the estimate for $P(M| -r, +f)$, after committing to an action, landing in the intended location of an action successfully occurs with probability $P(M = +m| -r, +f)$. The other three possible landing positions occur with probability $\frac{P(M=-m|-r,+f)}{3}$ each. Use this transition distribution to calculate the optimal action(s) to take and the expected utility of those actions.

(a) **(i)** [1 pt] **Rejection Sampling**: You want to estimate $P(M = +m | -r, +f)$ by rejection sampling. Below is a list of samples that were generated using prior sampling. Cross out those that would be rejected by rejection sampling.

$\cancel{+r \quad +e \quad +w \quad -m \quad -f}$	$-r \quad -e \quad +w \quad -m \quad +f$
$\cancel{-r \quad -e \quad +w \quad -m \quad -f}$	$\cancel{+r \quad -e \quad +w \quad +m \quad -f}$
$-r \quad +e \quad -w \quad -m \quad +f$	$-r \quad -e \quad -w \quad +m \quad +f$
$\cancel{+r \quad -e \quad -w \quad +m \quad -f}$	$\cancel{+r \quad -e \quad -w \quad +m \quad +f}$
$-r \quad -e \quad -w \quad -m \quad +f$	$-r \quad +e \quad +w \quad -m \quad +f$
$-r \quad +e \quad -w \quad -m \quad +f$	$-r \quad +e \quad -w \quad -m \quad +f$

All samples without the conditioning $-r, +f$ are rejected.

(ii) [1 pt] What is the approximation for $P(M = +m | -r, +f)$ using the remaining samples?
$\frac{1}{7}$, the fraction of accepted samples with $+m$ instantiated.

(iii) [1 pt] What are the optimal action(s) for Michael and John based on this estimate of $P(M = +m|-r, +f)$?
South, West. As $p(+m| - r, +f) = \frac{1}{7}$, $p(-m| - r, +f) = \frac{6}{7}$. Michael and John will succeed in the selected action $\frac{1}{7}$ of the time, or take one of the other 3 actions with equal probability of $\frac{2}{7}$. In this case, $p(+m| - r, +f)$ is so low that deciding to head in the direction of the whirlpool actually decreases the chances of landing in it.

(iv) [1 pt] What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m|-r, +f)$?
$\frac{1}{7} * (-50) + \frac{2}{7} * (-50) + \frac{2}{7} * (25) + \frac{2}{7} * (100) = \frac{100}{7}$, the weighted sum of all four outcomes.

(b) **(i)** [2 pts] **Likelihood Weighting:** Suppose instead that you perform likelihood weighting on the following samples to get the estimate for $P(M = +m| - r, +f)$. You receive 4 samples consistent with the evidence.

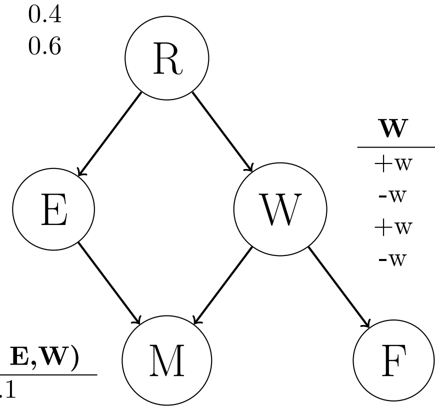
Sample	**Weight**
$-r \quad -e \quad +w \quad +m \quad +f$	$P(-r)P(+f\| + w) = 0.6 * 0.15 = 0.09$
$-r \quad -e \quad -w \quad +m \quad +f$	$P(-r)P(+f\| - w) = 0.6 * 0.75 = 0.45$
$-r \quad -e \quad +w \quad -m \quad +f$	$P(-r)P(+f\| + w) = 0.6 * 0.15 = 0.09$
$-r \quad +e \quad -w \quad -m \quad +f$	$P(-r)P(+f\| - w) = 0.6 * 0.75 = 0.45$

(ii) [1 pt] What is the approximation for $P(M = +m| - r, +f)$ using the samples above?
$\frac{0.09+0.45}{0.09+0.45+0.09+0.45} = \frac{1}{2}$

(iii) [1 pt] What are the optimal action(s) for Michael and John based on this estimate of $P(M = +m|-r, +f)$?
East

(iv) [1 pt] What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m|-r, +f)$?
$\frac{1}{6} * (-50) + \frac{1}{6} * (-50) + \frac{1}{6} * (25) + \frac{1}{2} * (100) = \frac{75}{2}$

Here is a copy of the Bayes' Net, repeated for your convenience.



R	P(R)
+r	0.4
-r	0.6

E	R	P(E | R)
+e	+r	0.3
-e	+r	0.7
+e	-r	0.6
-e	-r	0.4

W	R	P(W | R)
+w	+r	0.9
-w	+r	0.1
+w	-r	0.2
-w	-r	0.8

M	E	W	P(M | E,W)
+m	+e	+w	0.1
-m	+e	+w	0.9
+m	+e	-w	0.45
-m	+e	-w	0.55
+m	-e	+w	0.35
-m	-e	+w	0.65
+m	-e	-w	0.9
-m	-e	-w	0.1

F	W	P(F | W)
+f	+w	0.15
-f	+w	0.85
+f	-w	0.75
-f	-w	0.25

(c) **(i)** [3 pts] **Gibbs Sampling**. Now, we tackle the same problem, this time using Gibbs sampling. We start out with initializing our evidence: $R = -r$, $F = +f$. Furthermore, we start with this random sample:

$$-r \;\; +e \;\; -w \;\; +m \;\; +f.$$

We select variable E to resample. Calculate the numerical value for:
$P(E = +e|R = -r, W = -w, M = +m, F = +f)$.

$P(E = +e|R = -r, W = -w, M = +m, F = +f) = \frac{P(+e|-r)P(+m|+e,-w)}{P(+e|-r)P(+m|+e,-w)+P(-e|-r)P(+m|-e,-w)}$

$= \frac{0.6*0.45}{0.6*0.45+0.4*0.9} = \frac{3}{7}$

We resample for a long time until we end up with the sample:

$$-r \;\; -e \;\; +w \;\; +m \;\; +f.$$

Michael and John are happy for fixing this one sample, but they do not have enough time left to compute another sample before making a move. They will let this one sample approximate the distribution: $P(M = +m| - r, +f)$.

(ii) [1 pt] What is the approximation for $P(M = +m| - r, +f)$, using this one sample?
1

(iii) [1 pt] What are the optimal action(s) for Michael and John based on this estimate of $P(M = +m|-r, +f)$?
East

(iv) [1 pt] What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m|-r, +f)$?
100