

- 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 false] Uniform-cost search will never expand more nodes than A*-search.
- (ii) [true or false] Depth-first search will always expand more nodes than breadth-first search.
- (iii) [true or false] The heuristic $h(n) = 0$ is admissible for every search problem.
- (iv) [true or false] The heuristic $h(n) = 1$ is admissible for every search problem.
- (v) [true 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) [true 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 false] 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 false] 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 false] When using alpha-beta pruning, the computational savings are independent of the order in which children are expanded.
- (iii) [true or false] 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.

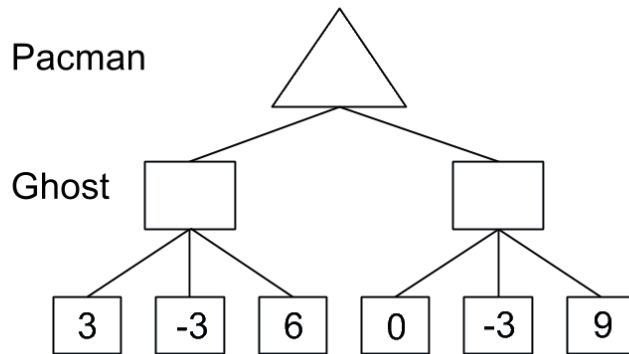
- (i) [true or false] For an MDP (S, A, T, γ, R) if we only change the reward function R the optimal policy is guaranteed to remain the same.
- (ii) [true or false] Value iteration is guaranteed to converge if the discount factor (γ) satisfies $0 < \gamma < 1$.
- (iii) [true or false] Policies found by value iteration are superior to policies found by policy iteration.

(e) [2 pts] Reinforcement Learning

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

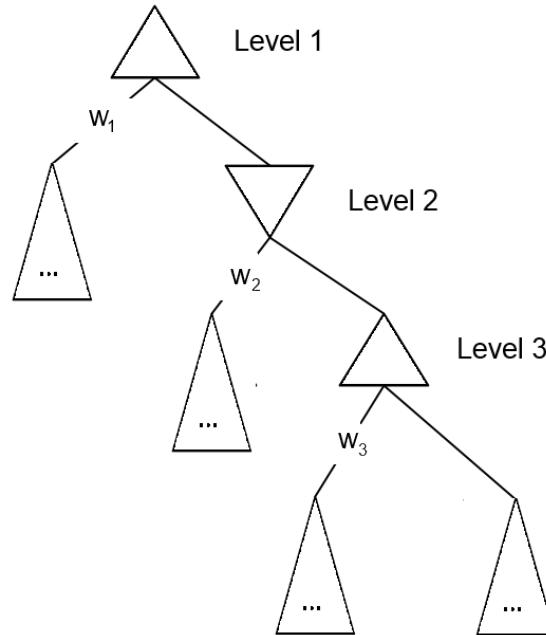
Q2. [20 pts] Variants of Trees

- (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 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 considering the right child of the node at level 3, you have access to w_1 , w_2 , and w_3 .



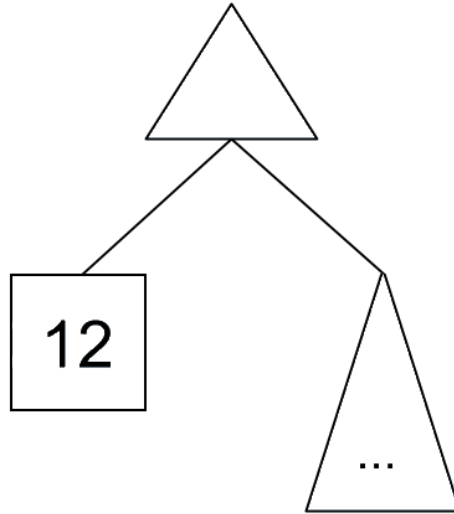
- (i) [2 pts] Under this new scenario, what is the pruning condition for a max node at the n^{th} level of the tree (in terms of v and $w_1 \dots w_n$)?

- (ii) [2 pts] What is the pruning condition for a min node at the n^{th} level of the tree?

- (iii) [3 pts] What is the relationship between α , β and the list of $w_1 \dots w_n$ 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 _____

- (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?

(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.

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) $|I - C| = 1$
- ix) $|P - B| = 2$

- (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

- (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

- (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

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

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

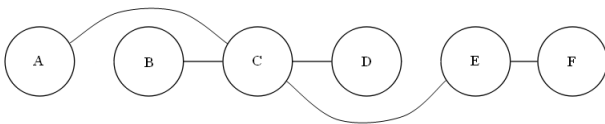
We say that a variable X is backtracked if, after a value has been assigned to X , the recursion returns at X without a solution, and a different value must be assigned to X .

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.

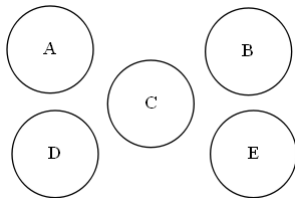


Algorithm 1	Algorithm 2	Algorithm 3
A-B-C-D-E-F	A-B-C-D-E-F	A-B-C-D-E-F
F-E-D-C-B-A	F-E-D-C-B-A	F-E-D-C-B-A
C-A-B-D-E-F	C-A-B-D-E-F	C-A-B-D-E-F
B-D-A-F-E-C	B-D-A-F-E-C	B-D-A-F-E-C
D-E-F-C-B-A	D-E-F-C-B-A	D-E-F-C-B-A
B-C-D-A-E-F	B-C-D-A-E-F	B-C-D-A-E-F

- (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.

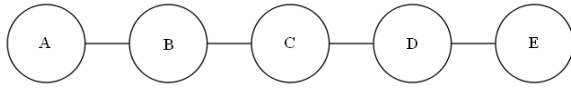
For each graph below, select whether or not using the MRV and/or LCV heuristics could affect the number of nodes expanded in the search tree in this new situation.

(i) [2 pts]



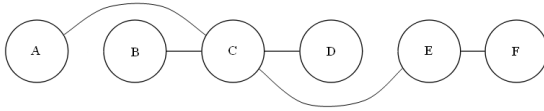
- ☐ Neither MRV nor LCV can have an effect.
- ☐ Only MRV can have an effect.
- ☐ Only LCV can have an effect .
- ☐ Both MRV and LCV can have an effect.

(ii) [2 pts]



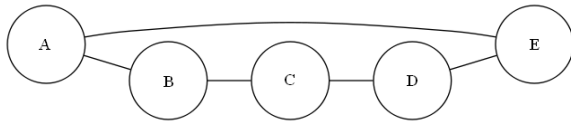
- ☐ Neither MRV nor LCV can have an effect.
- ☐ Only MRV can have an effect.
- ☐ Only LCV can have an effect .
- ☐ Both MRV and LCV can have an effect.

(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.

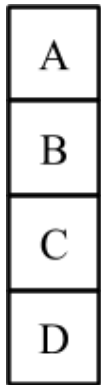
(iv) [2 pts]



- ☐ 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	
B	
C	
D	

- (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.

$$\text{___ } F_\infty(A) + \text{___ } F_\infty(B) + \text{___ } F_\infty(C) + \text{___ } F_\infty(D) = F_\infty(A)$$

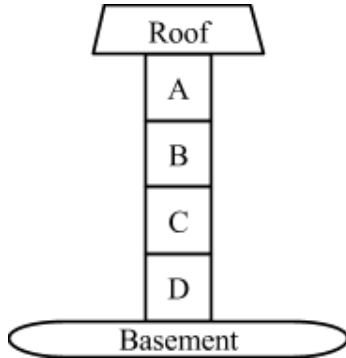
$$\text{___ } F_\infty(A) + \text{___ } F_\infty(B) + \text{___ } F_\infty(C) + \text{___ } F_\infty(D) = F_\infty(B)$$

$$\text{___ } F_\infty(A) + \text{___ } F_\infty(B) + \text{___ } F_\infty(C) + \text{___ } F_\infty(D) = F_\infty(C)$$

$$\text{___ } F_\infty(A) + \text{___ } F_\infty(B) + \text{___ } F_\infty(C) + \text{___ } F_\infty(D) = F_\infty(D)$$

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

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

S_b	$P(S_b d_b)$
+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?

(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
$X_1 = B$	
$X_2 = C$	
$X_3 = D$	

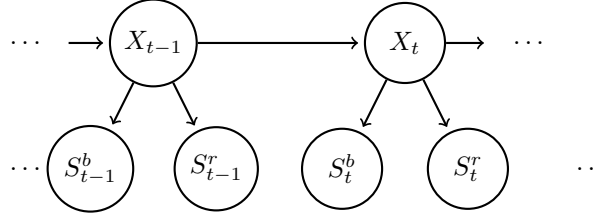
(iii) [2 pts] Assume the particles after time elapsing are $X_1 = B$, $X_2 = C$, $X_3 = D$, and the particle weights in the following table. What is the probability the particles will be resampled as $X_1 = B$, $X_2 = B$, and $X_3 = D$?

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

(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. $s_{0:t}^r$ are all the measurements from the roof sensor $s_0^r, s_1^r, s_2^r, \dots, s_t^r$. $s_{0:t}^b$ are all the measurements from the roof sensor $s_0^b, s_1^b, s_2^b, \dots, s_t^b$.

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



$$P(x_t | s_{0:t}^r, s_{0:t}^b) \propto P(x_t, s_{0:t}^r, s_{0:t}^b) \quad (1)$$

$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, s_{0:t}^r, s_{0:t}^b) \quad (2)$$

$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, s_{0:t-1}^r, s_t^r, s_{0:t-1}^b, s_t^b) \quad (3)$$

$$= \sum_{x_{t-1}} \text{_____} P(x_t | x_{t-1}) P(x_{t-1}, s_{0:t-1}^r, s_{0:t-1}^b) \quad (4)$$

- ☐ $P(s_t^r, s_t^b | x_{t-1}, x_t, s_{0:t-1}^r, s_{0:t-1}^b)$
- ☐ $P(s_t^r | x_t) P(s_t^b | x_t)$
- ☐ $P(s_t^r | x_{t-1}) P(s_t^b | x_{t-1})$
- ☐ $P(s_t^r | s_{t-1}^r) P(s_t^b | s_{t-1}^b)$
- ☐ $P(s_t^r, s_t^b | x_t)$
- ☐ $P(s_t^r, s_t^b | x_t, x_{t-1})$
- ☐ None of the above.

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' .

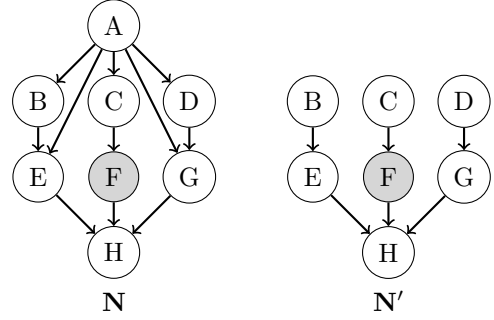
Elimination order is alphabetical.

All variables are binary $+/-$.

Factor size is the number of unobserved variables in a factor made during elimination.

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

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



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?

- (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?

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

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

- | | | |
|--------------------------------------|--|--|
| <input type="radio"/> $P(H \mid +f)$ | <input type="radio"/> $P(H, +a, +f)$ | <input type="radio"/> $P(H, +f \mid +a)$ |
| <input type="radio"/> $P(H \mid +a)$ | <input type="radio"/> $P(H, +a \mid +f)$ | <input type="radio"/> $P(H \mid +a, +f)$ |

- (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 $p(+h \mid +f)$?

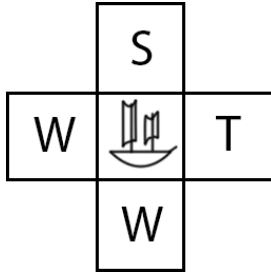
- | | |
|---|--|
| <input type="radio"/> $I_+(+h) \cdot p(+a) \cdot I_-(+h) \cdot p(-a)$ | <input type="radio"/> $\frac{I_+(+h) \cdot p(+a) \cdot I_-(+h) \cdot p(-a)}{\sum_h I_+(h) \cdot p(+a) \cdot I_-(h) \cdot p(-a)}$ |
| <input type="radio"/> $I_+(+h) \cdot p(+a) + I_-(+h) \cdot p(-a)$ | <input type="radio"/> $\frac{I_+(+h) \cdot p(+a) + I_-(+h) \cdot p(-a)}{\sum_h I_+(h) \cdot p(+a) + I_-(h) \cdot p(-a)}$ |
| <input type="radio"/> $I_+(+h) + I_-(+h)$ | <input type="radio"/> $\frac{I_+(+h) + I_-(+h)}{\sum_h I_+(h) + I_-(h)}$ |

- (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 $O(n \exp(f))$.

- | | | |
|--------------------------------------|--|--|
| <input type="radio"/> $O(n \exp(k))$ | <input type="radio"/> $O(n \exp(i))$ | <input type="radio"/> $O(n \exp(i + k))$ |
| <input type="radio"/> $O(n \exp(f))$ | <input type="radio"/> $O(n \exp(f - k))$ | <input type="radio"/> $O(n \exp(i/f))$ |

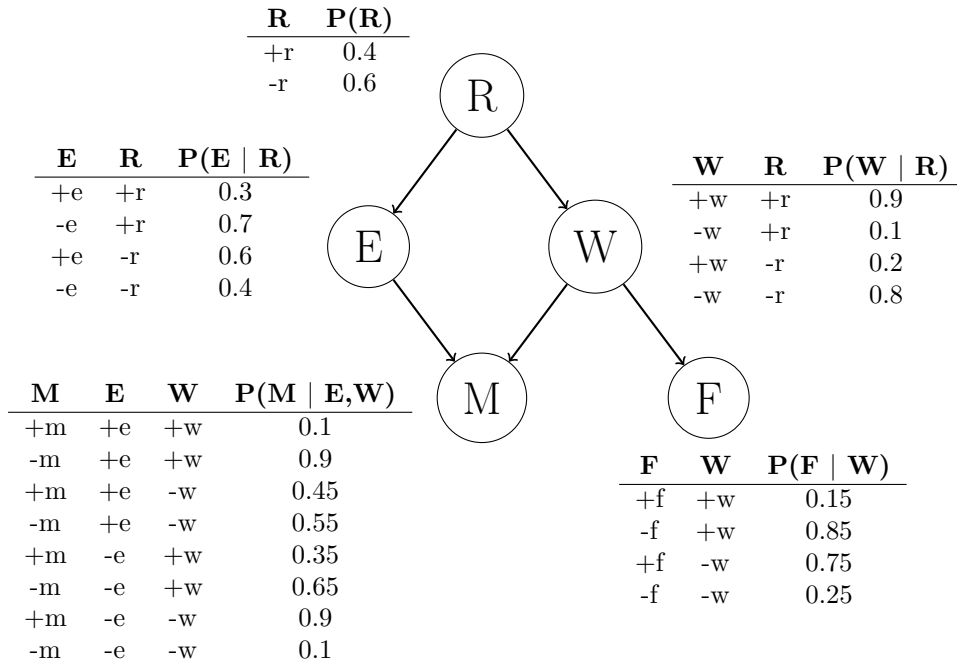
Q6. [15 pts] Argg! Sampling for the Legendary Treasure

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:



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.

$+r$	$+e$	$+w$	$-m$	$-f$	$-r$	$-e$	$+w$	$-m$	$+f$
$-r$	$-e$	$+w$	$-m$	$-f$	$+r$	$-e$	$+w$	$+m$	$-f$
$-r$	$+e$	$-w$	$-m$	$+f$	$-r$	$-e$	$-w$	$+m$	$+f$
$+r$	$-e$	$-w$	$+m$	$-f$	$+r$	$-e$	$-w$	$+m$	$+f$
$-r$	$-e$	$-w$	$-m$	$+f$	$-r$	$+e$	$+w$	$-m$	$+f$
$-r$	$+e$	$-w$	$-m$	$+f$	$-r$	$+e$	$-w$	$-m$	$+f$

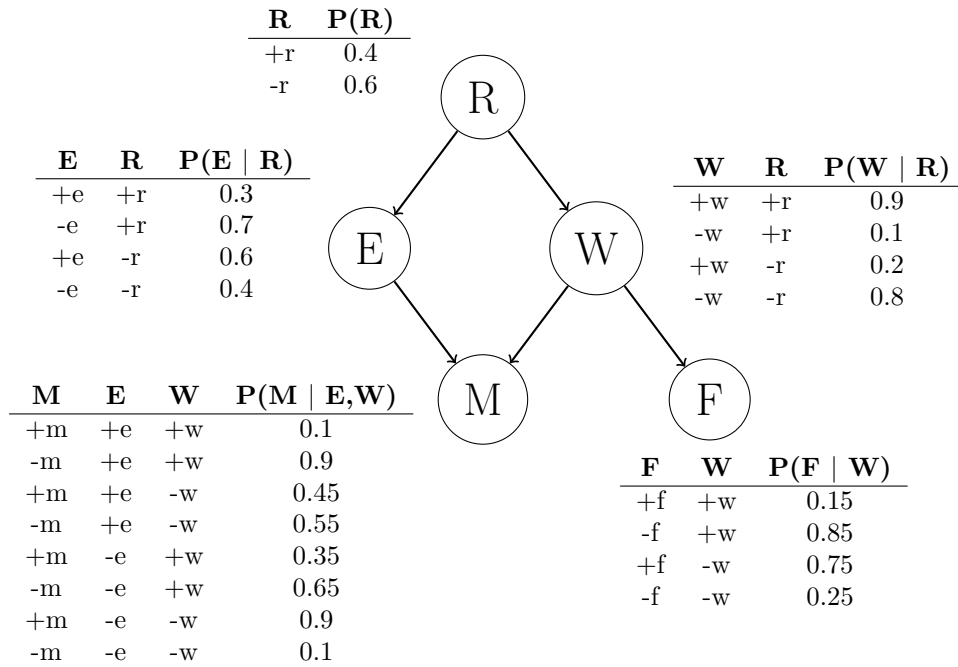
- (ii) [1 pt] What is the approximation for $P(M = +m | -r, +f)$ using the remaining samples?
- (iii) [1 pt] What are the optimal action(s) for Michael and John based on this estimate of $P(M = +m | -r, +f)$?
- (iv) [1 pt] What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m | -r, +f)$?

- (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
$-r$ $-e$ $+w$ $+m$ $+f$	
$-r$ $-e$ $-w$ $+m$ $+f$	
$-r$ $-e$ $+w$ $-m$ $+f$	
$-r$ $+e$ $-w$ $-m$ $+f$	

- (ii) [1 pt] What is the approximation for $P(M = +m | -r, +f)$ using the samples above?
- (iii) [1 pt] What are the optimal action(s) for Michael and John based on this estimate of $P(M = +m | -r, +f)$?
- (iv) [1 pt] What is the expected utility for the optimal action(s) based on this estimate of $P(M = +m | -r, +f)$?

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



- (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:
 $P(E = +e | R = -r, W = -w, M = +m, F = +f)$.

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