CS 181 Artificial Intelligence (Fall 2020), Midterm Exam

Name (in Chinese): ______________________

ID#: ______________________

Instructions

- Time: 8:15–9:55am (100 minutes)
- This exam is closed-book, but you may bring one A4-size cheat sheet. Put all the study materials and electronic devices into your bag and put your bag in the front, back, or sides of the classroom.
- You can write your answers in either English or Chinese.
- Two blank pieces of paper are attached, which you can use as scratch paper. Raise your hand if you need more paper.

1 Multiple choice (10 pt)

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 0.5 point if you select a non-empty proper subset of the correct answers. Fill your answers in the table below.

1	2	3	4	5

6	7	8	9	10

1. Which of the following tree search algorithm can **always** find the optimal solution?

   A. Depth-First Search (DFS).
   B. Breadth-First Search (BFS).
   C. Uniform Cost Search (UCS) with nonnegative edge.
   D. Greedy Search.
   E. A* Search.

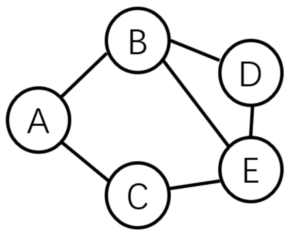
2. Consider the search problem in a finite state space. Which of the following statement(s) is/are correct?

   A. Suppose the depth of the shallowest solution is s and the branching factor is b. BFS would take time complexity of $O(b^s)$ and space complexity of $O(bs)$.
   B. Iterative deepening is not complete under certain circumstances.
   C. For UCS, both time complexity and space complexity are exponential in its effective depth.

D. A^* is optimal if the heuristic is admissible in tree search.

E. None of the above

3. On the map coloring CSP represented by the following binary constraint graph, we run backtracking search starting from variable A. The domain of each variable is $D = \{\text{R, G, B}\}$. Suppose we do not enforce arc consistency. Given the variable order "A B C D E" during backtracking search, which of the following statements is/are true.



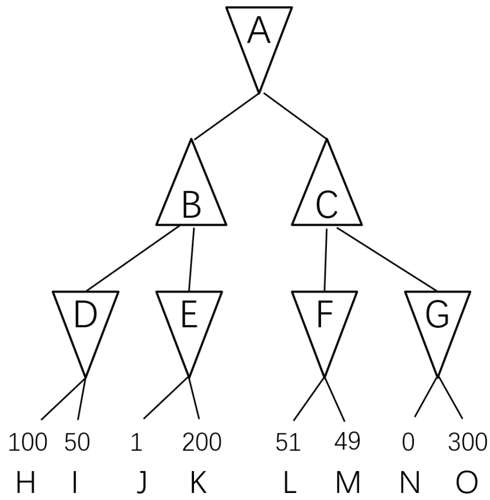
A. The given variable order satisfies the minimum remaining values ordering.

B. The value assignment R G G B R satisfies the least constraining value ordering.

C. The value assignment R G B B R satisfies the least constraining value ordering.

D. The value assignment R B B G R satisfies the least constraining value ordering.

4. Run min-max search with alpha-beta pruning in the following game tree. Choose all the nodes that will not be visited because of pruning.



A. G

B. K

C. N

D. O

5. Which of the following statements about propositional logic is/are correct?

A. If a sentence S is satisfiable, then $\neg S$ is unsatisfiable.

B. If a sentence S is unsatisfiable, then $\neg S$ is satisfiable but not valid.

C. A, B, C are clauses. Suppose applying resolution to A, B produces C. So we have $A \wedge B \equiv C$.

D. A, B, C are clauses. Suppose applying resolution to A, B produces C. So we have $A \wedge B \implies C$.

6. Which of the following statements about first-order logic is/are correct?

A. $\exists\ x, \text{Likes}(x, y)$ satisfies FOL syntax, where x, y are variables.

B. $\forall x, \text{Is}(x, \text{cat}) \wedge \text{Cute}(x)$ means "every cat is cute".

C. $\exists x, \text{Likes}(x, \text{iphone}) \equiv \neg\forall x \neg\text{Likes}(x, \text{iphone})$.

D. None of the above.

7. Which one of the following statements about first order logic is correct?

   A. For every pair of expressions, there is one and only one single most general unifier (or MGU) that is unique up to renaming and substitution of variables.

   B. Forward chaining is sound since every inference is essentially an application of Modus Ponens, but not complete since not every entailed atomic sentence will be derived.

   C. Backward chaining can avoid infinite loops by checking current goal against every goal on stack, but cannot avoid repeated subgoals since the previous results are gone.

   D. Let the domain be people in the world. $Loves(x, y)$ means "x loves y" (y may not love x). Then $\exists y \forall x\ Loves(x, y)$ entails $\forall x \exists y\ Loves(x, y)$.

8. Given factors $P(C|A), P(D|A, B, C), P(B|A, C)$, what is the resulting factor after joining over C and summing over C?

   A. $P(D|A)$

   B. $P(C, D|A)$

   C. $P(B, C, D|A)$

   D. $P(C, B|A, D) * P(A|D)$

   E. None of the above.

9. Which of the following statement(s) is/are correct?

   A. Bayes Net can be a cyclic graph.

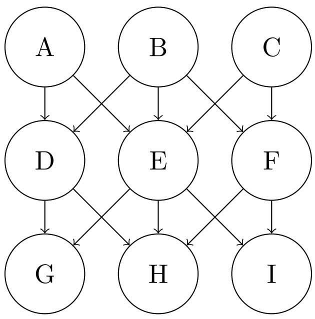
   B. In Bayes Net, each node is conditionally independent of all its ancestor nodes in the graph, given all of its parents.

   C. $P(X, Y, Z) = P(Z|X, Y)P(Y|X)P(X)$

   D. $X \perp\!\!\!\perp Y|Z$ can imply $P(X, Y) = P(X)P(Y)$.

   E. None of the above

10. Consider the Bayes Net below, which of the following statement(s) is/are correct?



   A. It is guaranteed that G is independent of H given D.

   B. It is guaranteed that A is independent of C.

   C. It is guaranteed that D is independent of C given F.

   D. It is guaranteed that E is independent of F given B and C.

   E. None of the above.

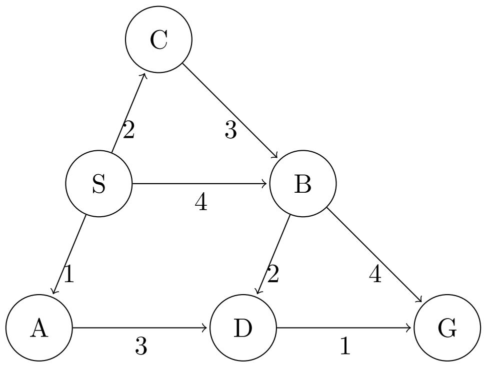
Solution:

1. C
2. CD
3. BD
4. BD
5. D
6. C
7. D
8. E
9. BC
10. BD

2 Search & CSP (10pt)

2.1 Search (5pt)

Consider performing graph search with the following directed graph, where S is the start state and G is the goal state. In all cases, resolve ties based on the alphabetical order (for example, if B and C have the same priority, then B is expanded first). Remember that in graph search, a state is expanded only once.



2.1.1 Determine the Strategy (2pt)

For each of the following paths, which search strategy might return the path? Place a checkmark (✓) in the corresponding cell.
Note: for BFS and DFS, ignore the cost.

Path	Breadth-first Search	Depth-first Search	Uniform cost search	None of the them
S-B-D-G				
S-B-G				
S-A-D-G				

Solution:

Path	Breadth-first Search	Depth-first Search	Uniform cost search	None of the them
S-B-D-G				✓
S-B-G	✓			
S-A-D-G		✓	✓	

2.1.2 Heuristic function property (3pt)

Recall that the A* algorithm will return an optimal result if and only if the heuristic function is *admissible* and *consistent*. For each heuristic function, circle whether it is admissible and whether it is consistent with respect to the search problem given above.

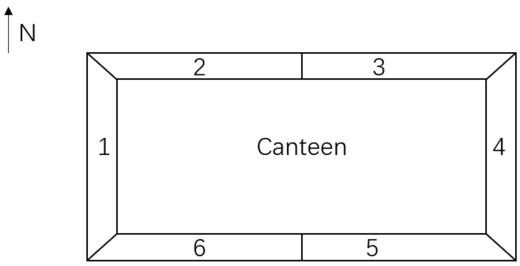
Heuristic H	$H(S)$	$H(A)$	$H(B)$	$H(C)$	$H(D)$	$H(G)$	Admissible	Consistent
α	5	4	4.5	5	1	0	Yes No	Yes No
β	5	3	2	5	1	0	Yes No	Yes No
γ	5	4	2	5	1	0	Yes No	Yes No

Solution:

Heuristic H	$H(S)$	$H(A)$	$H(B)$	$H(C)$	$H(D)$	$H(G)$	Admissible	Consistent
α	5	4	4.5	5	1	0	No	No
β	5	3	2	5	1	0	Yes	No
γ	5	4	2	5	1	0	Yes	Yes

2.2 Domain Filtering in CSP (5pt)

The university decorated a new canteen and would recruit six food sellers. The following is the map of the new canteen.



The new canteen has six positions (labelled from 1 to 6 in the map) for the following six sellers:

- Dumplings (L)
- Hotpot (H)
- Noddles (N)
- Fried rice (F)
- Barbeque (B)
- Pancake (P)

Positions can be *next to* one another, if they share a wall (for instance, positions 6-1). Positions can also be *opposite from* one another (specifically, positions 1-4, 2-6, 3-5). Due to resource limitation, the air exhaust fan cannot be installed on the south side of the canteen (i.e. positions 5 and 6) and water supply is not available on the north side of the canteen(i.e., positions 2 and 3).

The constraints are as follows:

i Dumplings and noddles must have water supply.

ii Dumplings and noddles must not be next to each other.

iii Barbeque and hotpot must have air exhaust fan, because they produce heavy smoke.

iv Fried rice must not be opposite from Barbeque.

v Hotpot must be next to noddles.

vi Dumplings must be next to pancake.

vii No two sellers may occupy the same position.

a) (2pt) The table below shows the variable domains after unary constraints have been enforced and position 1 has been assigned to variable D. Cross out all the values that are eliminated by running Forward Checking after the assignment of variable D.

D	1					
N	1			4	5	6
B	1	2	3	4		
F	1	2	3	4	5	6
H	1	2	3	4		
P	1	2	3	4	5	6

b) (2pt) The table below shows the variable domains after unary constraints have been enforced, position 1 has been assigned to variable D, and then position 5 has been assigned to variable N. Cross out all the values that are eliminated if arc consistency is enforced after the assignments.

D	1					
N					5	
B	1	2	3	4		
F	1	2	3	4	5	6
H	1	2	3	4		
P	1	2	3	4	5	6

c) (1pt) Does this CSP has a solution? If yes, write out one of them.

Solution:

a) The table below shows the variable domains after unary constraints have been enforced and the value 1 has been assigned to the variable D.
Cross out all values that are eliminated by running Forward Checking after this assignment.

D	1					
N	~~1~~			4	5	~~6~~
B	~~1~~	2	3	4		
F	~~1~~	2	3	4	5	6
H	~~1~~	2	3	4		
P	~~1~~	2	~~3~~	~~4~~	~~5~~	6

b) The table below shows the variable domains after unary constraints have been enforced, the value 1 has been assigned to the variable D, and now the value 5 has been assigned to variable N.
Cross out all values that are eliminated if arc consistency is enforced after this assignment. (Note that enforcing arc consistency will subsume all previous pruning.)

D	1					
N					5	
B	~~1~~	2	3	~~4~~		
F	~~1~~	2	3	~~4~~	~~5~~	6
H	~~1~~	~~2~~	~~3~~	4		
P	~~1~~	2	~~3~~	~~4~~	~~5~~	6

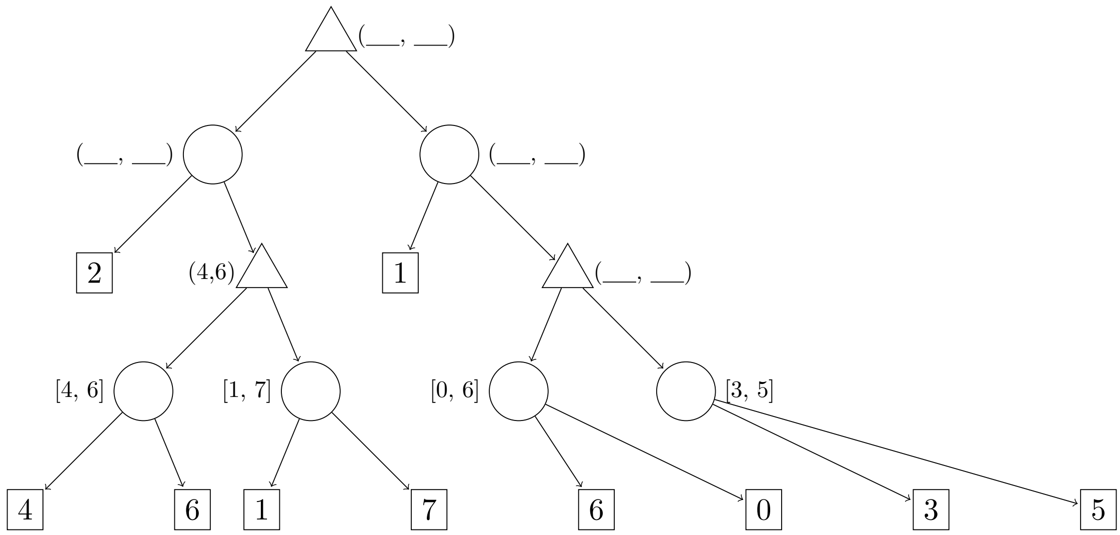
c) Yes.

$$(D, N, B, F, H, P) = (1, 5, 2, 3, 4, 6) or (1, 5, 3, 2, 4, 6) or (1, 5, 3, 6, 4, 2)$$

3 (Minimax) Pacman and Casper (10 pt)

Consider a two-player game between Pacman and a ghost (Casper) in which both agents alternate moves. As usual, Pacman tries to maximize his own utility. Unlike the usual minimizer ghost, Casper is friendly and **helps Pacman** by maximizing his utility. **Pacman is unaware** of this and acts as if Casper is playing against him.

For the game tree for this game, the value pair at each node is (u, v), where u is the value of the subtree as determined by Pacman, and v is the value of the subtree as determined by Casper. For example, in the subtree below with values (4, 6), Pacman believes Casper would choose the left action (with value = 4), but Casper actually chooses the right action (with value = 6), since that is better for Pacman.



3.1 Minimax (4 pt)

Fill in the remaining (u, v) values in the tree for this game above, where Pacman is the root. Casper nodes are circles, and Pacman nodes are triangles. (Hint: Pacman would make decision according to its own belief.)

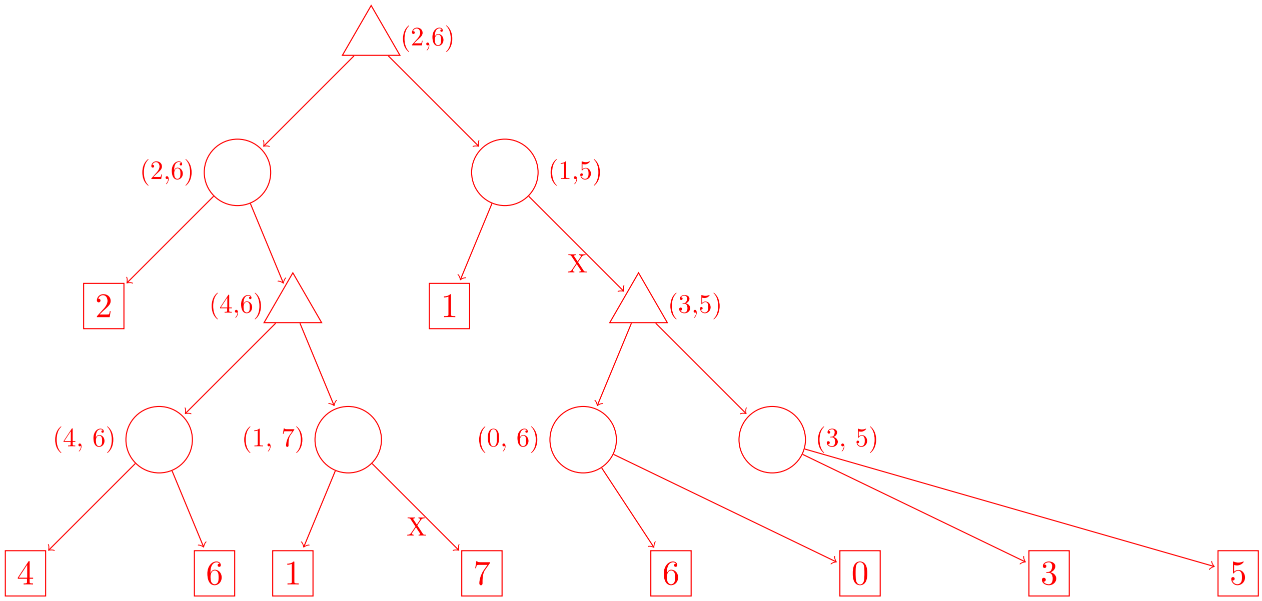
3.2 Pruning (4 pt)

In the game tree above, put an 'X' on the branches that can be pruned and do not need to be explored when **Casper** computes the value of the tree. Assume that children are visited left-to-right.

Solution:

The u value of Pacman's nodes is the maximum of the u values of the immediate children nodes since Pacman believes that the values of the nodes are given by u. The v value of Pacman's nodes is the v value from the child node that attains the maximum u value since, during Pacman's turn, he determines the action that is taken.

The u value of the ghost nodes is the minimum of the u values of the immediate children nodes since Pacman believes Casper would choose the action that minimizes his utility. The v value of the Casper nodes is the maximum of the v values of the immediate children nodes since, during his turn, she chooses the action that maximizes Pacman's utility.



3.3 Game result(2 pt)

What would the value of the entire game tree be if Pacman knew that Casper is friendly?

Solution:

7. It would be the maximum over the entire tree.

4 Logic (10 pt)

For multiple choices problems in this section, please choose the correct answer by filling up the circle like ●. For other problems, fill in the box or table.

4.1 CNF (2pt)

Let $\uparrow$ be a binary operation defined by the following truth table:

P	Q	$P \uparrow Q$
T	T	F
T	F	T
F	T	T
F	F	T

Given the knowledge base (KB):

$$A \Leftrightarrow ((B \wedge C) \uparrow D)$$

Convert KB to the conjunctive normal form.

Solution:

From the truth table, we can find that

$$P \uparrow Q \equiv \neg P \vee \neg Q$$

Thus,

$$\begin{aligned}
KB &\equiv A \Leftrightarrow ((B \wedge C) \uparrow D) \\
&\equiv A \Leftrightarrow (\neg (B \wedge C) \vee \neg D) \\
&\equiv A \Leftrightarrow (\neg B \vee \neg C \vee \neg D) \\
&\equiv (A \Rightarrow (\neg B \vee \neg C \vee \neg D)) \wedge ((\neg B \vee \neg C \vee \neg D) \Rightarrow A) \\
&\equiv (\neg A \vee \neg B \vee \neg C \vee \neg D) \wedge (\neg (\neg B \vee \neg C \vee \neg D) \vee A) \\
&\equiv (\neg A \vee \neg B \vee \neg C \vee \neg D) \wedge ((B \wedge C \wedge D) \vee A) \\
&\equiv (\neg A \vee \neg B \vee \neg C \vee \neg D) \wedge (A \vee B) \wedge (A \vee C) \wedge (A \vee D)
\end{aligned}$$

So the answer is

$$(\neg A \vee \neg B \vee \neg C \vee \neg D) \wedge (A \vee B) \wedge (A \vee C) \wedge (A \vee D)$$

4.2 First Order Logic (4pt)

Assume that the domain is people in the world. Consider the following sentence:

$$\neg \forall y \forall x (\neq (x, y) \Rightarrow Loves(x, y) \vee Hates(x, y))$$

where $Loves(x, y)$ means "x loves y" (y may not love x), $Hates(x, y)$ means "x hates y" (y may not hate x), $\neq (x, y)$ means "$x \neq y$".

1. Move negation inwards and try to rewrite the statements so that negations appear only with predicates (that is, so that no negation is outside a quantifier or an expression involving logical connectives).

Solution:

$$\exists y \exists x (\neq (x,y) \land \neg Loves(x,y) \land \neg Hates(x,y))$$

The order of x, y does not matter.

2. Which of the following sentences correctly describes the meaning of this sentence?

- ○ Not every people in the world love or hate all other people.
- ○ Not every people in the world both love and hate all other people.
- ○ There exists a person who is both loved and hated by some other person.
- ○ There exists a person who is neither loved nor hated by some other person.

Solution:
There exists a person who is neither loved nor hated by some other person.

4.3 Forward Chaining, Backward Chaining (4pt)

Suppose you are given the following axioms:

1. $|(1,5)$
2. $|(4,8)$
3. $\forall x \ |(x,x)$
4. $\forall x \ |(x, *(x,1))$
5. $\forall x \ |(*(x,1), x)$
6. $\forall x,y \ |(*(x,y), *(y,x))$
7. $\forall w,x,y,z \ |(w,y) \land |(x,z) \Rightarrow |(*(w,x), *(y,z))$
8. $\forall x,y,z \ |(x,y) \land |(y,z) \Rightarrow |(x,z)$

where $|(x,y)$ means $x|y$, i.e. x is a factor of y; $*(x,y)$ means $x * y$.

Notice that you can only use the axioms above for this problem, not anything else you may know about arithmetic.

1. Consider the forward chaining algorithm. For query $|(8,1)$, the algorithm will

- ○ Give an answer (i.e. Prove $|(8,1)$ successfully.)
- ○ Terminate with no answer
- ○ Never terminate

Solution:
Never terminate. From the axioms we cannot get $|(8,1)$. Since forward chaining is semi-decideable, it cannot say no to every non-entailed sentence.

2. Consider backward chaining algorithm. Suppose we want to prove $|(4, *(5,8))$ through backward chaining. The first few steps are given to you:

Steps	Axiom Number	Substitution	Goals
0			$\mid(4, *(5,8))$
1	8	$\{x/4, y/8, z/*(5,8)\}$	$\mid(4,8)$, $\mid(8, *(5,8))$
2	2	$\{\}$	$\mid(8, *(5,8))$
3	8	$\{x/8, y/*(8,1), z/*(5,8)\}$	$\mid(8, *(8,1))$, $\mid(*(8,1), *(5,8))$
4	4	$\{x/8\}$	$\mid(*(8,1), *(5,8))$

Now we need to prove $\mid(*(8,1), *(5,8))$. Is it possible for us to continue and prove $\mid(*(8,1), *(5,8))$ successfully? We say it can be proved if and only if all the goals can be reached finally. If it is possible, please write down a possible move of the next step (must be valid towards the end, i.e. goals are reachable and not repeating existed goals). If not, leave the table blank.

- ○ We can prove $\mid(*(8,1), *(5,8))$ successfully.
- ○ We cannot prove $\mid(*(8,1), *(5,8))$ successfully.

Steps	Axiom Number	Substitution	Goals
5			

Solution:

We can continue and prove $\mid(*(8,1), *(5,8))$ successfully.

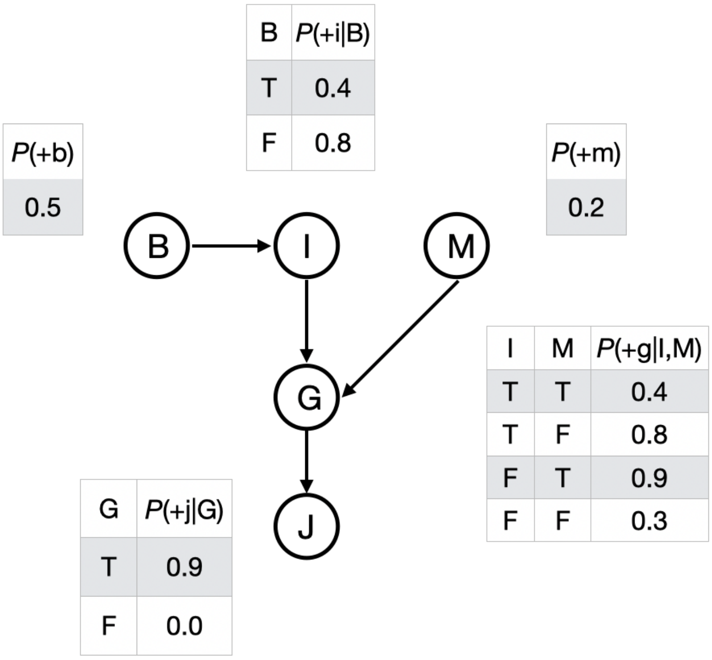
Steps	Axiom Number	Substitution	Goals
5	8	$\{x/*(8,1), y/*(1,8), z/*(5,8)\}$	$\mid(*(8,1), *(1,8))$, $\mid(*(1,8), *(5,8))$

Or

Steps	Axiom Number	Substitution	Goals
5	8	$\{x/*(8,1), y/*(8,5), z/*(5,8)\}$	$\mid(*(8,1), *(8,5))$, $\mid(*(8,5), *(5,8))$

5 Bayesian Network (10 pt)

Consider the following Bayesian network. (Note: $+x$ represents $X =$ true.)



5.1 Markov blanket (2pt)

Write the Markov blanket of the variable I.

Solution:
B, G, M

5.2 Exact Inference (5pt)

Using variable elimination to calculate $P(+m|+g)$. Write down the variable eliminating steps to get full points.

Solution:
Eliminate B
Factors: $f(B), f(+g|I, M), f(M), f(I|B)$

$f(+i) = \sum_b P(+i|b) * P(b) = P(+i|+b) * P(+b) + P(+i|-b) * P(-b) = 0.4*0.5+0.8*0.5 = 0.6$
$f(-i) = \sum_b P(-i|b) * P(b) = P(-i|+b) * P(+b) + P(-i|-b) * P(-b) = 0.6*0.5+0.2*0.5 = 0.4$

Eliminate I
Factors: $f(+g|I, M), f(M), f(I)$
$f(+g|+m) = \sum_i P(+g|i, +m) * P(i) = P(+g|+i, +m) * P(+i) + P(+g|-i, +m) * P(-i) = 0.4 * 0.6 + 0.9 * 0.4 = 0.6$
$f(+g|-m) = \sum_i P(+g|i, -m) * P(i) = P(+g|+i, -m) * P(+i) + P(+g|-i, -m) * P(-i) = 0.8 * 0.6 + 0.3 * 0.4 = 0.6$

Eliminate J
...

Join the remaining factors
Factors: $f(+g|M), f(M)$
$f(+g, +m) = f(+g|+m) * P(+m) = 0.6 * 0.2 = 0.12$

$f(+g,-m) = f(+g|-m) * P(-m) = 0.6 * 0.8 = 0.48$

Normalization
$P(+m|+g) = P(+g|+m) * P(+m)/P(+g) = 0.2$

5.3 Approximate Inference (3pt)

Suppose that we want to calculate $P(J|+i,+m)$ and the topological sort of variables is B, I, M, G, J.

5.3.1 Likelihood Weighting

Write down the likelihood weights of the following samples if it can be generated using likelihood weighting; otherwise, write False.

- $[true, true, true, true, true]$
- $[false, true, false, false, false]$

Solution:
$w = 1 * P(+i|+b) * P(+m) = 1 * 0.4 * 0.2 = 0.08$
False

5.3.2 Gibbs Sampling

Suppose that the initial state is $[false, true, true, false, false]$, we sample I now and get a random number $r = 0.43$. We determine the value (True/False) of random variable I based on the value of the random number r in the following way. If $r \leq P(+i|\ldots)$, then I is True; otherwise I is False. Write the new state after sampling I.

Solution:
$f(+i|-b,+m,-g) = P(+i|-b) * P(-g|+i,+m) = 0.48$
$f(-i|-b,+m,-g) = 0.02$
So $P(+i|\ldots) = 0.96$
$r \leq P(+i|-b,+m,-g)$, I is True, new state is $[false, true, true, false, false]$