CS 181 Artificial Intelligence (Fall 2021), Midterm Exam

Name (in Chinese): _____

ID#: _____

Instructions

- Time: 10:15–11:55am (100 minutes)
- This exam is closed-book, but you may bring one A4-size cheat sheet. Put all the study materials and electronic devices (with the exception of a calculator) 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 statements is/are true?

   A. A search algorithm is complete if it is guaranteed to find a solution if one exists.

   B. Breadth-First-Search (BFS) is an optimal tree search algorithm.

   C. 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 $O(b^s)$.

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

   E. None of the above.

2. In an A* algorithm, if h_1 and h_2 are admissible, which of the following is(are) also guaranteed to be admissible?

   A. $h_1 + h_2$

   B. $max(h_1, h_2)$

   C. $min(h_1, h_2)$

   D. $\alpha h_1 + (1 - \alpha)h_2, \alpha \in [0, 1]$

E. None of the above

3. Which of the following statements about adversarial search is/are correct?

   A. The effectiveness of alpha-beta pruning is highly dependent on the order in which the states are examined.

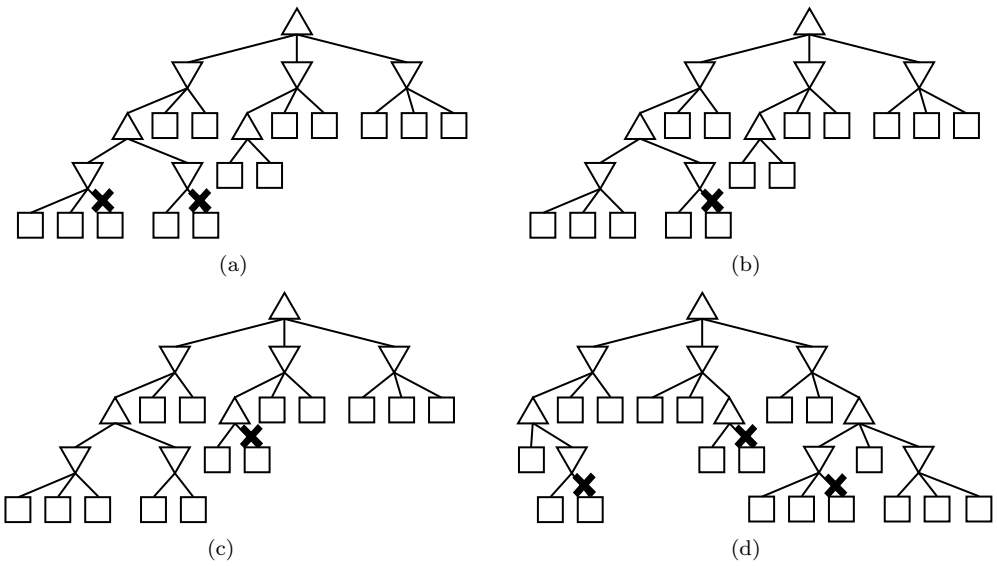
   B. The alpha-beta search algorithm computes the same optimal move as minimax, but achieves much greater efficiency by eliminating subtrees that are provably irrelevant.

   C. The time complexity of minimax algorithm is $O(b^m)$ and the space complexity is $O(bm)$ (assuming that when a state is expanded, all the child states are generated at once and kept in memory) where the maximum depth of the tree is m and there are b legal moves at each point.

   D. If your opponent is actually running a depth-2 minimax, using the result 80% of the time and moving randomly otherwise, then you should use expectimax as your main algorithm.

   E. None of the above.

4. Assume we run α-β pruning, expanding successors from left to right, on a game with trees as shown below.



   Which of the following statements is/are correct?

   A. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (a) will be achieved.

   B. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (b) will be achieved.

   C. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (c) will be achieved.

   D. There exists an assignment of utilities to the terminal nodes such that the pruning shown in Figure (d) will be achieved.

   E. None of the above

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. Compared with backward chaining, forward chaining may do more irrelevant work to the goal.

   C. $\neg A \vee \neg B \vee C \vee D$ is a Horn clause.

D. A is valid if and only if True $\models A$.

E. None of the above.

6. Which of the following statements of equivalence is/are correct?

   A. $A \vee \neg B \vee \neg C \equiv (\neg A \wedge B) \Rightarrow \neg C$.

   B. $(A \Rightarrow B) \vee (B \Rightarrow C) \equiv A \vee \neg A$.

   C. $(A \Rightarrow C) \wedge (B \Rightarrow C) \equiv (\neg A \wedge \neg B) \vee (\neg A \wedge C) \vee (\neg B \wedge C) \vee C$.

   D. $(A \wedge \neg B) \vee (B \wedge \neg A) \equiv (A \vee B) \wedge (\neg A \vee \neg B)$.

   E. None of the above.

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

   A. $\exists x$ $\mathrm{P}(x, y)$ is legal in FOL syntax.

   B. $\exists x$ $\mathrm{At}(x, \mathrm{Shanghai}) \Rightarrow \mathrm{Love}(x, \mathrm{ShanghaiTech})$ represents "Someone at Shanghai loves ShanghaiTech".

   C. $\exists x \forall y$ $\mathrm{IsFriend}(x, y) \equiv \forall y \exists x$ $\mathrm{IsFriend}(x, y)$.

   D. "No dog bites a child of its owner" can be translated as the following FOL expression: $\neg[\exists x \forall y$ $\mathrm{Dog}(x) \wedge \mathrm{Child}(y, \mathrm{Owner}(x)) \wedge \neg \mathrm{Bite}(x, y)]$.

   E. None of the above.

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

   A. If $A \perp\!\!\!\perp B$, then $P(A, B, C)$ can be written as $P(A, C)P(B|A, C)$.

   B. $P(B, C) = \sum_A P(B, C|A)$.

   C. When collecting a sample during likelihood weighting, evidence variables are not sampled.

   D. When collecting a sample during rejection sampling, variables can be sampled in any order.

   E. None of the above.

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

   A. Bayes Nets are always acyclic.

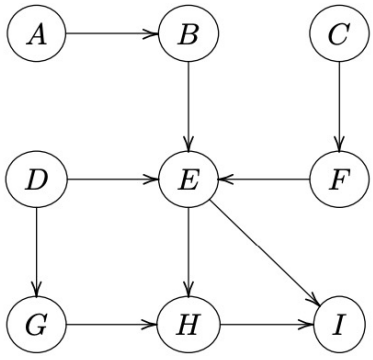
   B. If a Bayes Net has n variables, we need $O(n)$ space to store the joint distribution of all variables.

   C. If $A \perp\!\!\!\perp B$, then $(A \perp\!\!\!\perp B)|C$.

   D. If $(A \perp\!\!\!\perp B)|C$, then $A \perp\!\!\!\perp B$.

   E. None of the above

10. Consider the Bayes Net below, which of the following statements is/are correct?



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

   B. It is guaranteed that F is independent of I given E.

C. We can use Gibbs sampling to approximate $P(H|B)$.

D. The Markov blanket of H is B, D, E, F, G, I.

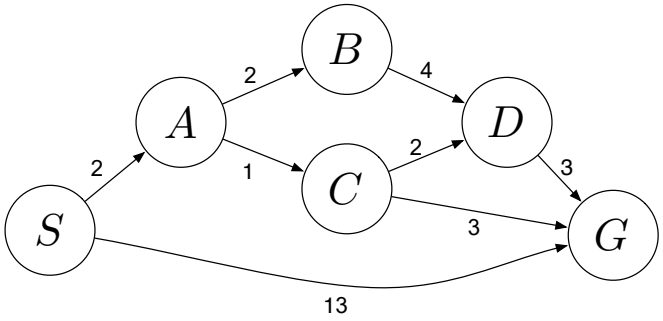
E. None of the above.

Solution:

1. AC
2. BCD
3. ABCD
4. BD
5. BD
6. ABCD
7. E
8. AC
9. A
10. C

2 Search & CSP (10 pt)

2.1 Search (5 pt)



Answer the following questions about the search problem shown above. S is the start state and G is the goal state. Break any ties alphabetically. For the questions that ask for a path, please give your answers in the form '$S - A - D - G$.'

(a) What path would breadth-first graph search return for this search problem? (0.5 pt)

Solution:
S-G

(b) What path would depth-first graph search return for this search problem? (0.5 pt)

Solution:
S-A-B-D-G

(c) What path would uniform cost graph search return for this search problem? (0.5 pt)
What is the sequence of the nodes that uniform cost search will expand? (0.5 pt)

Solution:
S-A-C-G

Solution:
S-A-C-B-D-G

(d) What path would A* graph search, using a consistent heuristic, return for this search problem? (1 pt)

Solution:
S-A-C-G

(e) Consider the heuristics for this problem shown in the table below. (2 pt)

State	h
S	5
A	4
B	6
C	2
D	3
G	0

(i) Is h admissible? **Yes** **No**
If your answer is "No", please give your reason.

(ii) Is h consistent? **Yes** **No**
If your answer is "No", please give your reason.

Solution:
(i) Yes

Solution:
(ii) No, because h(A)-h(C)=2 > 1 overestimate the cost from A to C

2.2 CSP (5 pt)

After years of competing, Xiyangyang decided to move to a new village with his friends, Meiyangyang, Feiyangyang, Lanyangyang, Nuanyangyang and Manyangyang. The move has forced Xiyangyang to change the housing assignments in the new village, which has 6 houses. He has decided to figure out the new assignments with a CSP in which the variables are Xiyangyang **(X)**, Meiyangyang **(M)**, Feiyangyang **(F)**, Lanyangyang **(L)**, Nuanyangyang **(N)**, and Manyangyang **(V)**, the values are which house they will stay in, from 1-6, and the constraints are:

i) No two of them can stay in the same house

ii) $\mathbf{X} > 3$

iii) **L** is less than **X**

iv) **M** is either 5 or 6

v) $\mathbf{X} > \mathbf{M}$

vi) **F** is even

vii) **N** is not 1 or 6

viii) $|\mathbf{N}\text{-}\mathbf{V}| = 1$

ix) $|\mathbf{X}\text{-}\mathbf{F}| = 2$

(a) **Unary constraints.** On the grid below cross out the values from each domain that are eliminated by enforcing **unary constraints**. (Grid on the right is a backup for you in case you messed up with your notations) (1 pt)

X	1	2	3	4	5	6
F	1	2	3	4	5	6
V	1	2	3	4	5	6
L	1	2	3	4	5	6
N	1	2	3	4	5	6
M	1	2	3	4	5	6

X	1	2	3	4	5	6
F	1	2	3	4	5	6
V	1	2	3	4	5	6
L	1	2	3	4	5	6
N	1	2	3	4	5	6
M	1	2	3	4	5	6

(b) **MRV.** According to the Minimum Remaining Value (MRV) heuristic, which variable should be assigned to first? Fill the circle of your choice. (1 pt)

◯ X ◯ F ◯ V ◯ L ◯ N ◯ M

(c) **Forward checking.** Regardless of your answer to the previous problem, assume that we choose to assign X first, and assign it the value 6. What are the resulting domains after enforcing unary constraints (from part a.) and running forward checking for this assignment? (Grid on the right is a backup for you in case you messed up with your notations) (1 pt)

X						6
F	1	2	3	4	5	6
V	1	2	3	4	5	6
L	1	2	3	4	5	6
N	1	2	3	4	5	6
M	1	2	3	4	5	6

X						6
F	1	2	3	4	5	6
V	1	2	3	4	5	6
L	1	2	3	4	5	6
N	1	2	3	4	5	6
M	1	2	3	4	5	6

(d) **Iterative Improving** 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:

X:6, F:4, V:3, L:2, N: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 reassignment, 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. (2 pt)

Variable	# violated
X	
F	
V	
L	
N	
M	

	1	2	3	4	5	6
X						
F						
V						
L						
N						
M						

Solution:

(i)

X ~~1 2 3~~ 4 5 6

F ~~1~~ 2 ~~3~~ 4 ~~5~~ 6

V 1 2 3 4 5 6

L 1 2 3 4 5 6

N ~~1~~ 2 3 4 5 ~~6~~

M ~~1 2 3 4~~ 5 6

(ii)

M

(iii)

X 6

F ~~1 2 3~~ 4 ~~5 6~~

V 1 2 3 4 5 ~~6~~

L 1 2 3 4 5 ~~6~~

N ~~1~~ 2 3 4 5 ~~6~~

M ~~1 2 3 4~~ 5 ~~6~~

(iv)

Variable	# violated
X	0
F	0
V	1
L	0
N	2
M	0

	1	2	3	4	5	6
X						
F						
V		x				
L						
N		x		x		
M						

3 Prune under the Chance Nodes! (10 pt)

Figure 1: Chance nodes often exists in card games. How to prune?

In class we have learned that we can do alpha-beta pruning to a minimax tree and reduce search time. However, we cannot apply the same approach to an expectimax tree due to incomplete information from the chance nodes. This question considers pruning in games with chance nodes.

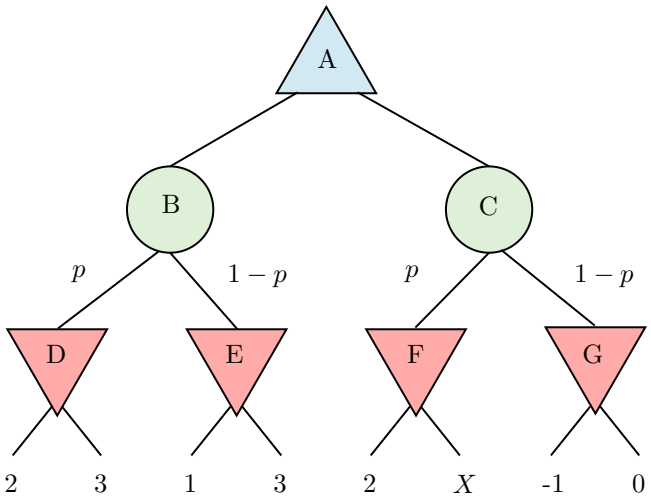


Figure 2: The complete game tree for a trivial game with chance nodes.

Figure 2 shows the complete game tree for a trivial game. The upper-triangle node is a max-node, the circle nodes are chance nodes (with probabilities assigned to the after-coming edges) and the lower-triangle nodes are min-nodes. Assume that the leaf nodes are to be evaluated in left-to-right order, and that before a leaf node is evaluated, we know nothing about its value – the range of possible values is $-\infty$ to ∞.

3.1 Node Values (3 pt)

Assume that $p = 0.5$ and $X = 0$. What are the values of all the internal nodes?

A	B	C	D	E	F	G

Solution:

A	B	C	D	E	F	G
1.5	1.5	-0.5	2	1	0	-1

3.2 Can we prune? (2 pt)

Still assume that $p = 0.5$ and $X = 0$. Given the values of the first seven leaves, do we need to evaluate the eighth leaf? Choose the correct answer by filling up the circle like ●. Ambiguous answer will receive no points.

- ○ Yes, we need to evaluate the eighth leaf.
- ○ No, we don't need to evaluate the eighth leaf.

Solution:
No, we don't.

3.3 Restricted Leaf Nodes (3 pt)

Assume that $p = 0.4$. Suppose the leaf node values are known to lie between -3 and 3 (include -3 and 3). After the first two leaves are evaluated, what is the value range of node B?

$$\boxed{} \leq B \leq \boxed{}$$

What is the maximal value of X that we can prune the last 2 leaves? (If you believe that the value does not exist, write down "N/A".)

$$\boxed{}$$

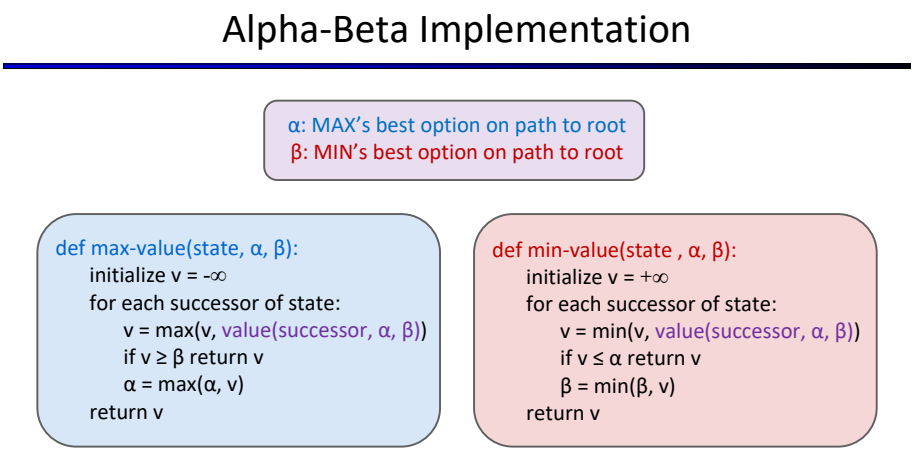
Solution:
$-1 \leq B \leq 2.6$
The maximal value of X is -1.

3.4 Alpha-Beta Pruning with Restricted Leaf Nodes (2 pt)

In problem 3.3, you have given the bound on node B. This shares the same idea as alpha-beta pruning. Recall that

- α = the value of the best (i.e., highest-value) choice we have found so far at any choice point along the path for MAX. Think: α = "at least."
- β = the value of the best (i.e., lowest-value) choice we have found so far at any choice point along the path for MIN. Think: β = "at most."

Now we can apply alpha-beta pruning on this game tree! We have already known how to deal with the MAX and MIN nodes in class:



Suppose the leaf node values are known to lie between $-k$ and k (include $-k$ and k, $k > 0$). This would lead to the initialization of α and β being $-k$ and k, respectively (instead of $-\infty$ and ∞ you've learned in class). Your job is to implement the algorithm for the chance nodes. A template has been written for you:

```
def exp-value(state, α, β):
    initialize v = 0, q = 0
    for each successor of state:
        p = probability(successor)
        q += p
        v += p * value(successor, (α−v−(1−q)k)/p, (β−v+(1−q)k)/p)
        if __[1]__ then return v + (1 − q)k
        if __[2]__ then return v − (1 − q)k
        α = max(α, __[3]__)
        β = min(β, __[4]__)
    return v
```

Fill in the blanks to complete the algorithm:

[1] _____

[2] _____

[3] _____

[4] _____

Solution:

1. $v + (1 - q) \cdot k \leq \alpha$
2. $v - (1 - q) \cdot k \geq \beta$
3. $v - (1 - q) \cdot k$
4. $v + (1 - q) \cdot k$

4 Logic (10 pt)

4.1 Propositional Logic (3pt)

4.1.1 CNF (1pt)

Given the knowledge base (KB):

$A \Leftrightarrow (B \wedge \neg C)$

$E \Rightarrow (A \vee \neg D)$

$\neg D \Rightarrow \neg C$

C

Convert KB to the conjunctive normal form.

Solution:
$(\neg A \vee B) \wedge (\neg A \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg D \vee \neg E) \wedge (\neg C \vee D) \wedge C$

4.1.2 Resolution Inference Rule (2pt)

Suppose we have $\alpha = \neg A \wedge \neg E$, prove KB$\models \alpha$ by resolution inference rule.
(Note: show the whole inference process and apply the resolution rule to two clauses each step.)

Solution:
First, construct KB$\wedge\neg\alpha$ as $(\neg A \vee B) \wedge (\neg A \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee \neg D \vee \neg E) \wedge (\neg C \vee D) \wedge C \wedge (A \vee E)$. Then apply resolution inference rule to get clauses with $\neg A$ and $\neg E$ respectively. Finally an empty clause is obtained.

4.2 First Order Logic (4pt)

In this question, we have function

$MapColor(x)$: the map color of x;

and predicates

$In(x, y)$: x is in y;

$Border(x, y)$: x borders y;

$Country(x)$: x is a country;

whose arguments are geographical regions, along with constant symbols for various regions (such as $Paris$). In each of the following we give an English sentence and a number of candidate logical expressions. For each of the logical expressions, state whether it

(1) correctly expresses the English sentence;

(2) is syntactically invalid and therefore meaningless;

(3) is syntactically valid but does not express the meaning of the English sentence.

For simplicity, we assume that $x \neq y$ in expressions with two variables, so you should not consider it as a missing component.

Note: Please indicate your answers with **clear correspondence** (e.g. i-(3), ii-(2), iii-(1)). For each question, you will get 2pt for all 3 correct answers, 1pt for only 2 correct answers, and 0pt for less than 2 correct answers.

a) No region in South America borders any region in Europe.

i. $\neg[\exists c, d\ In(c, SouthAmerica) \wedge In(d, Europe) \wedge Border(c, d)]$.

ii. $\forall c, d\ [In(c, SouthAmerica) \wedge In(d, Europe)] \Rightarrow \neg Border(c, d)$.

iii. $\neg\forall c\ In(c, SouthAmerica) \Rightarrow [\exists d\ In(d, Europe) \wedge \neg Border(c, d)]$.

b) No two adjacent countries have the same map color.

i. $\forall x, y\ \neg Country(x) \vee \neg Country(y) \vee \neg Border(x, y) \vee \neg(MapColor(x) = MapColor(y))$.

ii. $\forall x, y\ (Country(x) \wedge Country(y) \wedge Border(x, y) \Rightarrow \neg(MapColor(x) = MapColor(y))$.

iii. $\forall x, y\ (Country(x) \wedge Country(y) \wedge Border(x, y)) \Rightarrow MapColor(x \neq y)$.

Solution:
a): i-(1), ii-(1), iii-(3) b): i-(1), ii-(1), iii-(2).

4.3 Backward Chaining (3pt)

Suppose you are given the following axioms:

1. $p(7)$.
2. $p(13)$.
3. $\perp(3, 13)$.
4. $\neg|(13, 5)$.
5. $\forall x,\ \perp(x, 1)$.
6. $\forall x, y\ \perp(x, y) \Rightarrow \perp(y, x)$.
7. $\forall x, y\ p(x) \wedge p(y) \Rightarrow \perp(x, y)$.
8. $\forall x, y\ p(x) \wedge \neg|(x, y) \Rightarrow \perp(x, y)$.
9. $\forall x, y, z\ \perp(x, y) \wedge \perp(x, z) \Rightarrow \perp(x, *(y, z))$.

where $p(x)$ means x is a prime number; $\perp(x, y)$ means x and y are co-prime numbers, i.e. the greatest common factor of x and y is 1; $|(x, y)$ means $x|y$, i.e. x is a factor of y; $*(x, y)$ means $x * y$. For simplicity, we assume that numbers are distinct for different variables in an axiom.

Give a backward-chaining proof of the sentence $\perp(13, *(3, *(5, 7)))$ in the following table. Part of the proof is given in the table. Fill in the blanks to finish the proof.

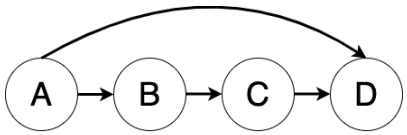
Steps	Axiom Number	Substitution	Goals
0	-	-	$\perp(13, *(3, *(5, 7)))$
1	9	$\{x/13, y/3, z/*(5, 7)\}$	$\perp(13, 3)$, $\perp(13, *(5, 7))$
2	6	$\{x/3, y/13\}$	$\perp(3, 13), \perp(13, *(5, 7))$
3	3	$\{\}$	
4	9		$\perp(13, 5), \perp(13, 7)$
5	8	$\{x/13, y/5\}$	
6	2		$\neg\vert(13, 5), \perp(13, 7)$
7	4	$\{\}$	
8	7		$p(13), p(7)$
9	2	$\{\}$	$p(7)$
10	1	$\{\}$	-

Solution:

Steps	Axiom Number	Substitution	Goals
0	-	-	$\perp(13, *(3, *(5, 7)))$
1	9	$\{x/13, y/3, z/*(5, 7)\}$	$\perp(13, 3)$, $\perp(13, *(5, 7))$
2	6	$\{x/3, y/13\}$	$\perp(3, 13), \perp(13, *(5, 7))$
3	3	$\{\}$	$\perp(13, *(5, 7))$
4	9	$\{x/13, y/5, z/7\}$	$\perp(13, 5), \perp(13, 7)$
5	8	$\{x/13, y/5\}$	$p(13), \neg\vert(13, 5), \perp(13, 7)$
6	2	$\{\}$	$\neg\vert(13, 5), \perp(13, 7)$
7	4	$\{\}$	$\perp(13, 7)$
8	7	$\{x/13, y/7\}$	$p(13), p(7)$
9	2	$\{\}$	$p(7)$
10	1	$\{\}$	-

5 Bayesian Network (10 pt)

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



$P(A)$	
$+a$	0.5
$-a$	0.5

$P(B\|A)$		
$+a$	$+b$	0.8
$+a$	$-b$	0.2
$-a$	$+b$	0.4
$-a$	$-b$	0.6

$P(C\|B)$		
$+b$	$+c$	0.1
$+b$	$-c$	0.9
$-b$	$+c$	0.7
$-b$	$-c$	0.3

$P(D\|A,C)$			
$+a$	$+c$	$+d$	0.6
$+a$	$+c$	$-d$	0.4
$+a$	$-c$	$+d$	0.1
$+a$	$-c$	$-d$	0.9
$-a$	$+c$	$+d$	0.2
$-a$	$+c$	$-d$	0.8
$-a$	$-c$	$+d$	0.5
$-a$	$-c$	$-d$	0.5

5.1 Markov blanket (1pt)

Write the Markov blanket of variable C.

Solution:
A, B, D

5.2 Markov network (1pt)

Convert the given Bayesian network to a Markov network and draw it down.

Solution:
Directed arrows to undirected edges. Link A and C additionally.

5.3 Exact inference (3pt)

Use variable elimination to calculate $p(+a|+d)$. Write down all steps to get full points.

Solution:

Expected Time (5min)

Eliminate B (1pt):

$p(+c|+a) = \sum_B p(B|+a)p(+c|B) = 0.8 * 0.1 + 0.2 * 0.7 = 0.22$
$P(-c|+a) = \sum_B p(B|+a)p(-c|B) = 0.8 * 0.9 + 0.2 * 0.3 = 0.78$
$P(+c|-a) = \sum_B p(B|-a)p(+c|B) = 0.4 * 0.1 + 0.6 * 0.7 = 0.46$
$P(-c|-a) = \sum_B p(B|-a)p(-c|B) = 0.4 * 0.9 + 0.4 * 0.3 = 0.54$

Eliminate C (1pt):

$p(+d|+a) = \sum_C p(C|+a)p(+d|+a, C) = 0.22 * 0.6 + 0.78 * 0.1 = 0.21$
$p(+d|-a) = \sum_C p(C|-a)p(+d|-a, C) = 0.46 * 0.2 + 0.54 * 0.5 = 0.362$

Normalize (1pt):

$p(+d) = \sum_A p(+d|A)p(A) = 0.5 * 0.21 + 0.5 * 0.362 = 0.286$
$P(+a|+d) = p(+d, +a)/p(+d) = 0.36713286713287$

5.4 Approximate Inference

5.4.1 Reject Sampling (1pt)

What is the expectation of a sample to be rejected if we want to approximate $P(+d|+a,+b)$?

Solution:
$P(+a,+b) = P(+a) * P(+b|+a) = 0.5 * 0.8 = 0.4$. So the expectation of a sample to be rejected is 1-0.4=0.6.

5.4.2 Likelihood Weighting (2pt)

If we want to use likelihood weighting sampling to approximate $P(+a|+b,+d)$, write down the weights of each sample and approximated $P(+a|+b,+d)$.

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

Solution:
$w_1 = p(+b|-a) * p(+d|-c,-a) = 0.4 * 0.5 = 0.2$
$w_2 = p(+b|+a) * p(+d|-c,+a) = 0.8 * 0.1 = 0.08$
$w_3 = p(+b|+a) * p(+d|-c,+a) = 0.8 * 0.1 = 0.08$
$w_4 = p(+b|-a) * p(+d|+c,-a) = 0.4 * 0.2 = 0.08$
$w_5 = p(+b|+a) * p(+d|+c,+a) = 0.8 * 0.6 = 0.48$
$P(+a|+b,+d) = \frac{0.08+0.08+0.48}{0.2+0.08+0.08+0.08+0.48} = 0.69565217391304$

5.4.3 Gibbs Sampling (2pt)

Consider doing Gibbs sampling for this example. Assume that we have initialized all variables to the values $-a, +b, +c, +d$. We then unassign the variable C, such that we have A $= -a$, B $= +b$, C $= ?$, D $= +d$. Calculate the probabilities for new values of C (i.e., $P(+c|-a,+b,+d), P(-c|-a,+b,+d)$) at this stage of the Gibbs sampling procedure.

Solution:
$w(+c) = p(+c|+b) * p(+d|-a,+c)$
$w(-c) = p(-c|+b) * p(+d|-a,-c)$
$P(+c) = \frac{w(+c)}{w(+c)+w(-c)}, P(-c) = \frac{w(-c)}{w(+c)+w(-c)}$
so, $P(+c) = \frac{0.1*0.2}{0.1*0.2+0.9*0.5} = 0.042553191489362$
$P(-c) = \frac{0.9*0.5}{0.1*0.2+0.9*0.5} = 0.95744680851064$