FIT5047 Intelligent Systems
Solutions to Practice Questions for Final Exam

1. [**Agents**] Consider the vacuum agent presented in class, but assume that the room has 8 squares and a square can get dirty after it has been cleaned.

   (a) Use PEAS to specify the task environment.
   SOLUTION:
   Performance measure: cleanliness of the room, number of moves.
   Environment: room, furniture, people.
   Actuators: vaccum suction.
   Sensors: vision, touch.

   (b) Specify the attributes of the environment.
   SOLUTION:
   Known (the conditions of the environment are known)
   Stochastic (squares can become dirty at any time)
   Observable (everything can be seen)
   Multiple agents (there are people who may dirty the room)
   Sequential (next action depends on previous action)
   Static (the environment may change, but not while the robot is deciding)
   Continuous (the wheels of the robot may go in any direction, and stop at any point).

2. [**Agents**] Consider shopping for used AI books on the Internet.

   (a) Use PEAS to specify the task environment
   SOLUTION:
   Performance measure: time spent, money spent.
   Environment: network, electronic device.
   Actuators: Hands.
   Sensors: Eyes, Ears (if there are voices from the web page).

   (b) Specify the attributes of the environment.
   SOLUTION
   Partially observable (could not know all used AI books in the Internet)
   Deterministic (you click, the device responds)
   Sequential (next actions, e.g., open a webpage depends on previous action, e.g. click the link)
   Dynamic (the status of the book might change while the agent is thinking)
   Discrete (the actions have a finite number of distinct states: click the book, view the book, buy the book etc.)
   Single agent.

3. **[Backtrack]** Specify a state representation, operators and goal test to solve the N-queen problem with N=4:
Four queens have to be placed on a 4X4 board where no row will have more than one queen, no column will have more than one queen, and no diagonal will have more than one queen.
Illustrate the workings of the backtracking algorithm (Backtrack1) to solve this problem.

SOLUTION

Note: The solution is trying to align with what you've learned in the lecture, you can of course have a better representation and better operators to find a solution faster.

State representation:
An empty list of tiles [0000] to represent no queen is placing in the board. Each position in the list represents the row number (between 1-4) of the queen that is in column i.

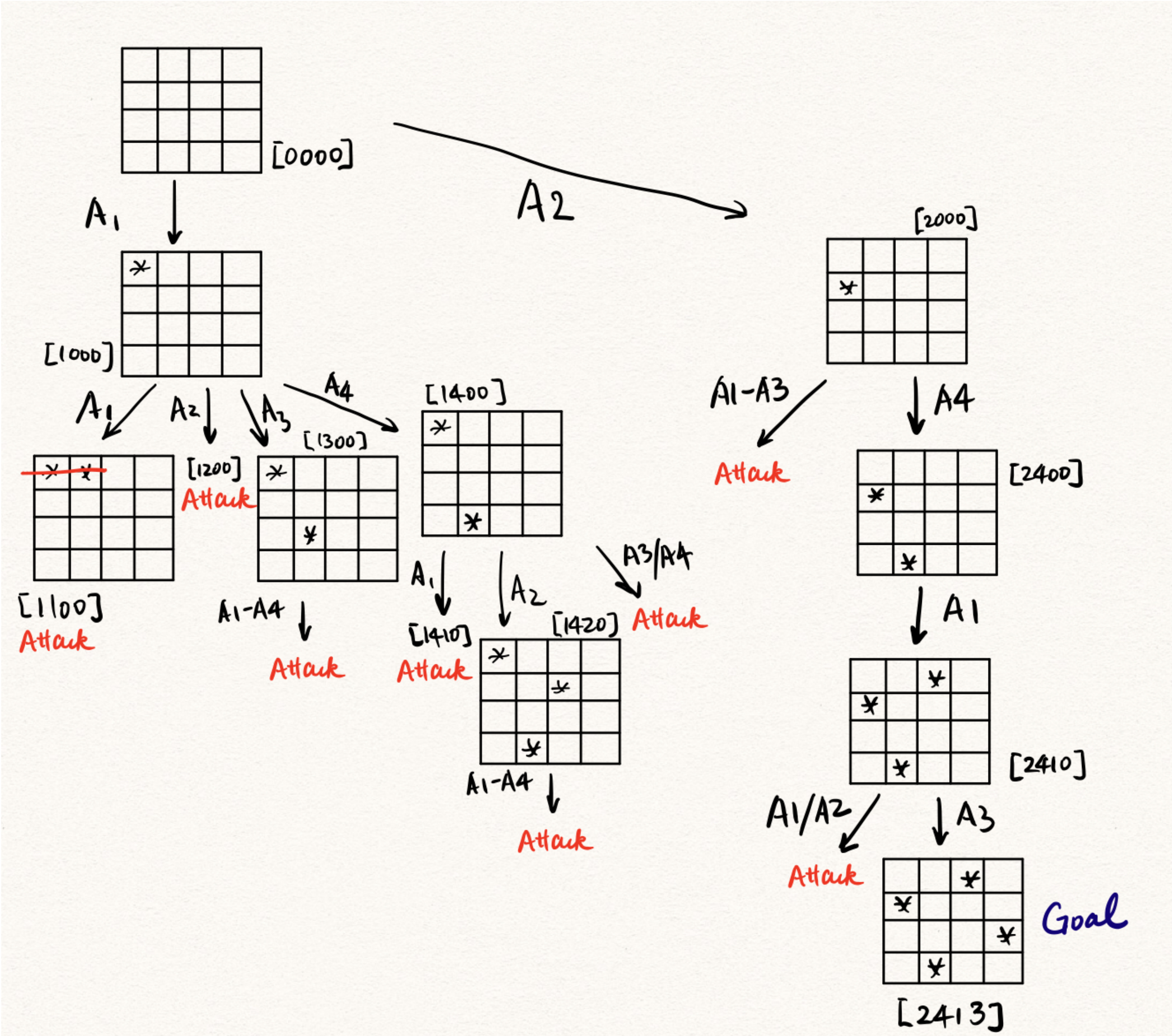
For example, a representation board = [1234] means a board with queens ($\star$) as below:

$\star$			
	$\star$		
		$\star$	
			$\star$

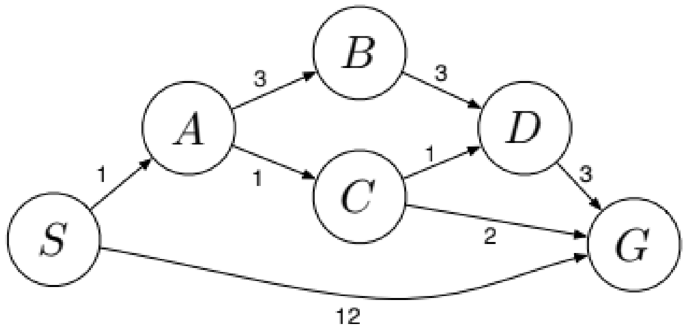
Operators:
Assign the tile value for the next unassigned queen, i.e., A_i means "assign the queen to row i, and to the first empty column". So A_i can be applied at most 4 times.

Goal test:
Check rows: All the numbers in the list are different.
Check columns: You don't need to do column check for this representation as no queen will be in the same column.
Check diagonals: Any pair of queen should have different row and column difference. If a queen is in row $r1$, column $c1$ and another queen in $r2$ and $c2$, then $|r1-r2| \neq |c1-c2|$ should hold for any pair of queens. For instance, if the board is [3124], the second queen ($r1 = 1$, $c1 = 2$) and the third queen ($r2 = 2$, $c2 = 3$) are attacking each other diagonally because $|1 - 2| = |2 - 3|$.

Backtrack illustration:


[0000]
A1
A2
[1000]
[2000]
A1
A2
A3
A4
[1100]
Attack
[1200]
Attack
[1300]
A1-A4
Attack
[1400]
A1
A2
A3/A4
Attack
[1410]
Attack
[1420]
A1-A4
Attack
A1-A3
Attack
A4
[2400]
A1
[2410]
A1/A2
Attack
A3
Goal
[2413]

4. **[GraphSearch]**



Answer the following questions about the search problem shown above. Break any ties alphabetically. S is the start and G is the goal.

(a) Breadth-first graph search
List the nodes according to their order of expansion. For each expansion, list OPEN (**with the nodes in the correct order**) and CLOSED.
SOLUTION:
S OPEN={A, G}, CLOSED={S}
G is generated, so no more expansions.

What path would BFS return for this search problem?
SOLUTION: S-G

(b) Uniform cost graph search
List the nodes according to their order of expansion. For each expansion, list OPEN (**with the nodes in the correct order**) and CLOSED.
SOLUTION:
S OPEN={A(1), G(12)}, CLOSED={S(0)}
A OPEN={C(2), B(4), G(12)}, CLOSED={S(0), A(1)}
C OPEN={D(3), G(min{12, **4**}), B(4) }, CLOSED={S(0), A(1), C(2)} **NOTE:** You can use strategies such as deeper path first when you are breaking ties.
D OPEN={G(min{12, **4**, 6}), B(4)}, CLOSED={S(0), A(1), C(2), D(3)}
G OPEN={B(4)}, CLOSED={S(0), A(1), C(2), D(3), G(4)}

What path would uniform-cost return for this search problem?
SOLUTION: S-A-C-G

(c) Depth-first graph search

List the nodes according to their order of expansion. For each expansion, list OPEN (**with the nodes in the correct order**) and CLOSED.

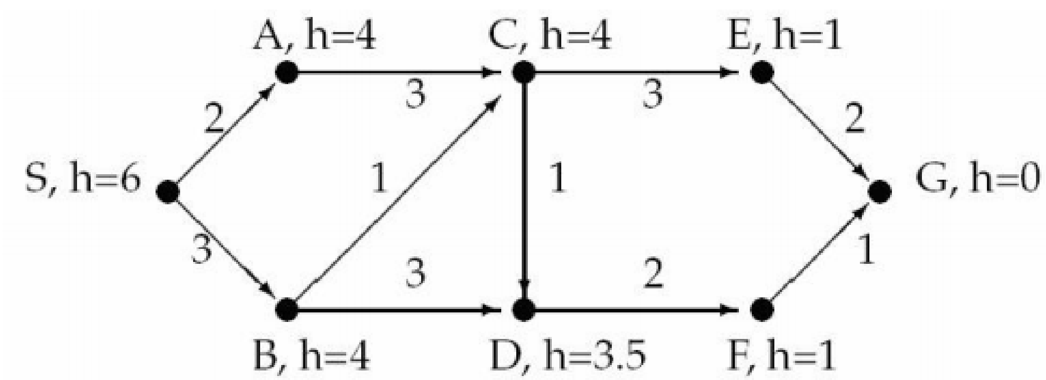
SOLUTION:

S OPEN={A(1), G(12)}, CLOSED={S(0)}
A OPEN={B(4), C(2), G(12)}, CLOSED={S(0), A(1)}
B OPEN={D(7), C(2), G(12)}, CLOSED={S(0), A(1), B(4)}
D OPEN={G(10), C(2)}, CLOSED={S(0), A(1), B(4), D(7)}
G OPEN={C(2)}, CLOSED={S(0), A(1), B(4), D(7), G(10)}

What path would DFS return for this search problem?

SOLUTION: S-A-B-D-G

5. **[GraphSearch]** Suppose we want to use algorithm A* on the graph below to find the shortest path from node S to node G. Each node is labeled by a capital letter and the value of a heuristic function. Each edge is labeled by the cost to traverse that edge.



For this problem:

- Apply algorithm A* to this graph, filling in the table below. Indicate the f, g, and h values of each node in OPEN and CLOSED as shown in the first two rows of the table.
  Assume that if you find a path to a node already in the queue that you update its cost (using the lower f value) instead of adding another copy of that node to the queue.

  SOLUTION:

iteration	node expanded	OPEN	CLOSED
0		S=0+6=6	
1	S	A=2+4=6; B=3+4=7	S
2	A	B=7, C=9	S, A
3	B	C=8, D=9.5	S, A, B
4	C	E=8, D=8.5	S, A, B, C
5	E	D=8.5, G=9	S, A, B, C, E
6	D	F=8, G=9	S, A, B, C, E, D
7	F	G=8	S, A, B, C, E, D, F
8	G		S, A, B, C, E, D, F, G

- Show the path found by algorithm A* on the graph above.
  SOLUTION: S-B-C-D-F-G.

6. [**Algorithm A***] The evaluation function *f(n)* = *d(n)* + *W(n)*, where *d(n)* is the cost of arriving at node n and *W(n)* is the sum of Manhattan Distance for each misplaced tile, is used in conjunction with an *A** algorithm to search from the start node:

2	8	3
1	6	4
7		5

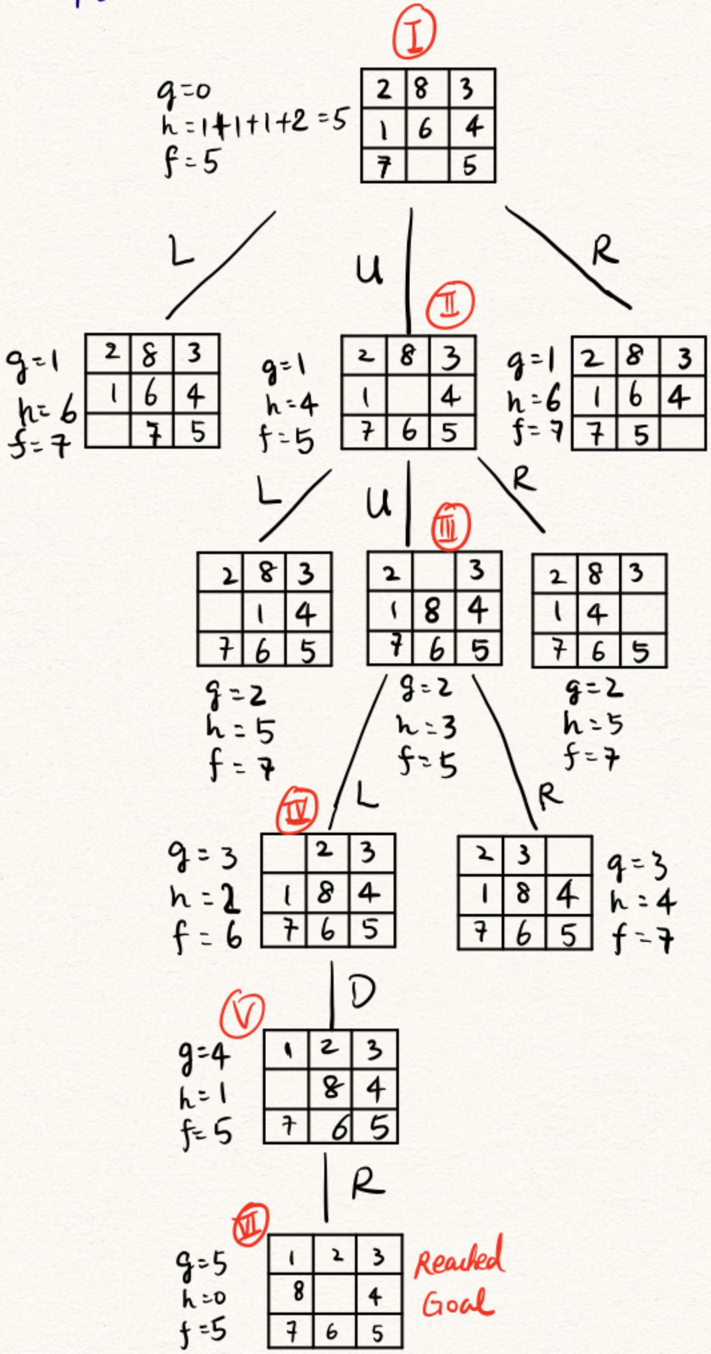
To the goal node:

1	2	3
8		4
7	6	5

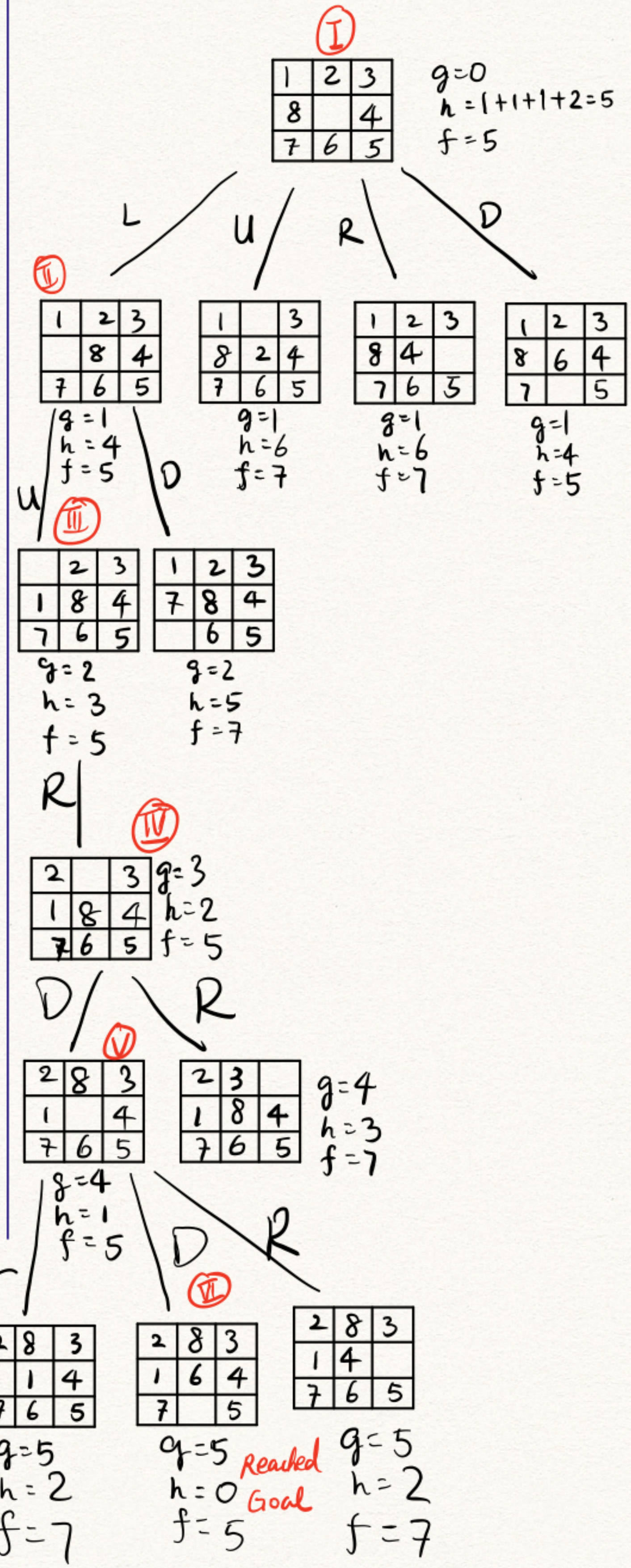
Use this evaluation function to search forward (from start node to goal node) and backward (from the goal node to the start node). Where would the backward search meet the forward search?

SOLUTION:
From the search paths, we can see that Step III in Forward search met the same state of Step IV in backward search.

Forward Search



Backward Search

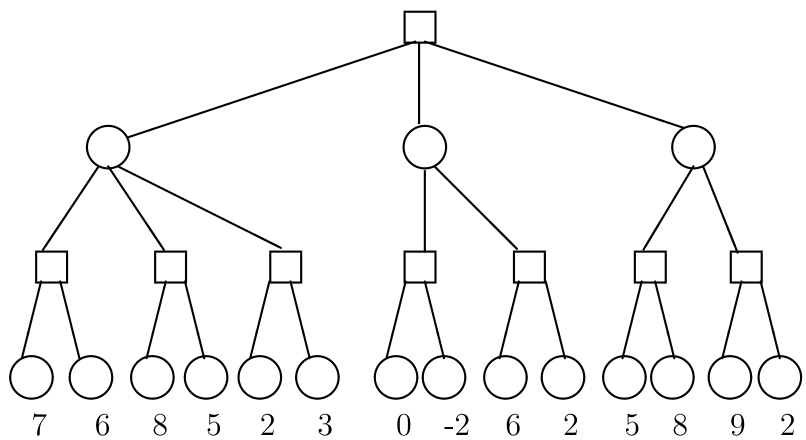


7. **[Search]** Which of the following are true and which are false? Explain your answers.

   (a) Depth-first search always expands at least as many nodes as A* search with an admissible heuristic.
   SOLUTION: False: a lucky DFS might expand exactly d nodes to reach the goal. A* dominates any graph-search algorithm that is guaranteed to find an optimal solution.

   (b) h(n) = 0 is an admissible heuristic for the 8-puzzle.
   SOLUTION: True: h(n) = 0 is always an admissible heuristic, since costs are nonnegative.

8. **[Game playing]** Consider the following game tree, where MAX plays in the square positions and MIN plays in the circles.

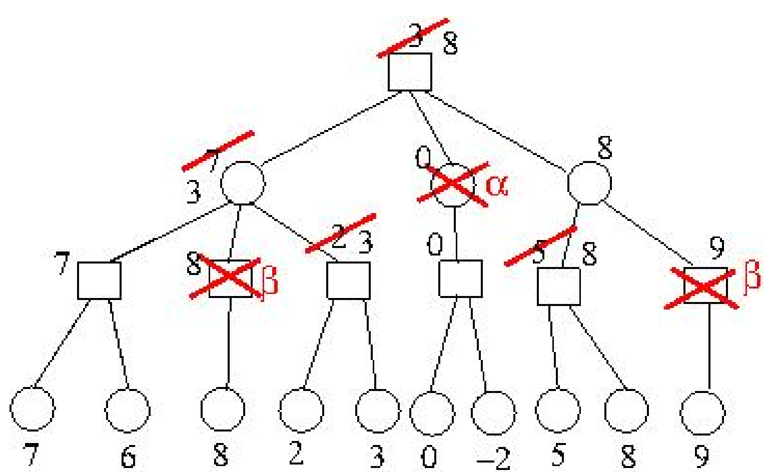


   (a) Conduct an α-β search of this game starting at the leftmost node to determine which move should MAX make. Draw the resulting game tree so that only the visited nodes appear in your diagram, i.e., **without** the nodes that are cut off. Indicate clearly the backed up value of each node, the updates performed on the backed up values, the α cut-offs and the β cut-offs you have performed.

   (b) What is the best move for MAX and what is its backed up value?

SOLUTION:

   (a) α-β search.



(b) Best move is to the right.
Backed-up value is 8.

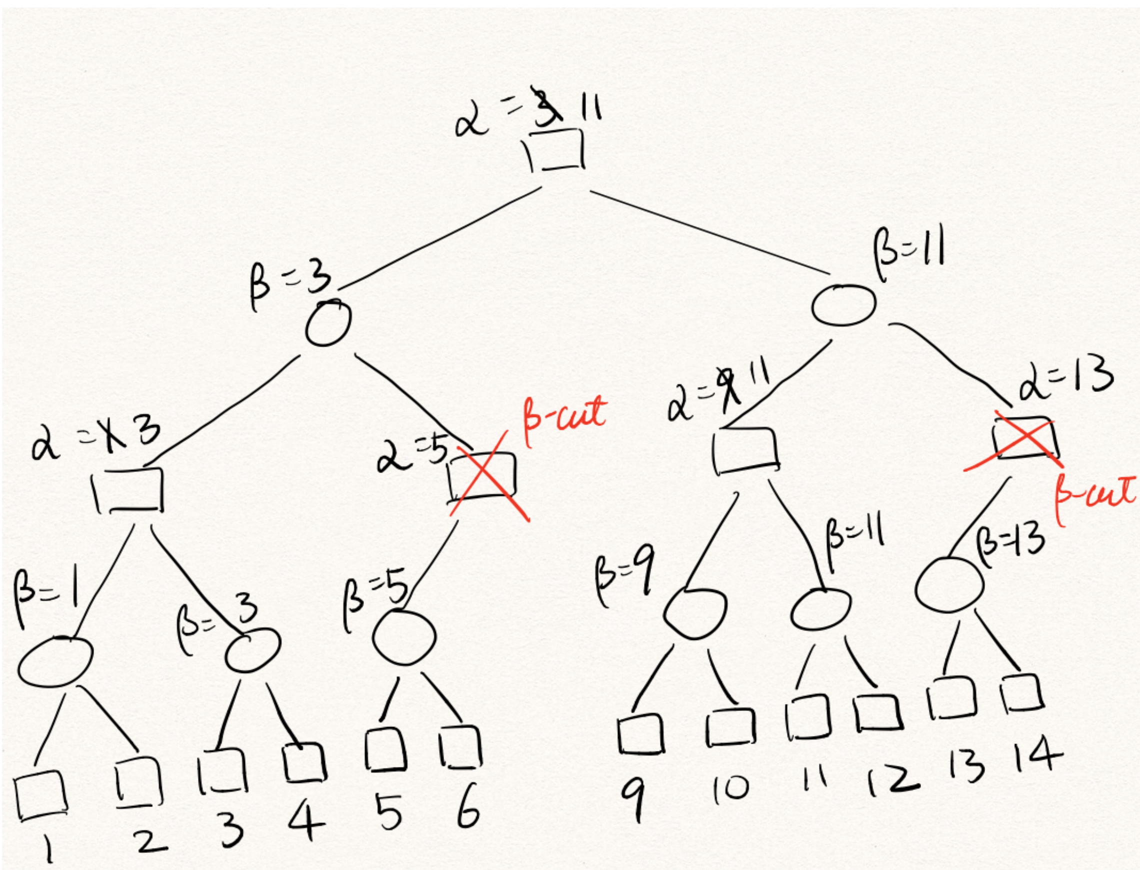
9. **[Game playing]** Consider a game tree with branching factor 2 and depth 5. Assume that it is MAX's turn to play, and that the evaluation function at the leaf nodes is in increasing order from left to right, such that its value for the leftmost node is 1, and for the rightmost node is 16 (the leaf nodes are MAX nodes).

   Conduct an α-β search of this game tree, starting from leftmost-node-first. In your α-β tree, clearly indicate the propagation of the α and β values, the performed cutoffs and the inspected leaf nodes.

   Upon completion of the search, state the final backed-up value of the root node and the recommended move (Left or Right). Also state the number of α and β cutoffs performed, and the number of leaf nodes generated.

   SOLUTION:

   - The game tree is as follows.



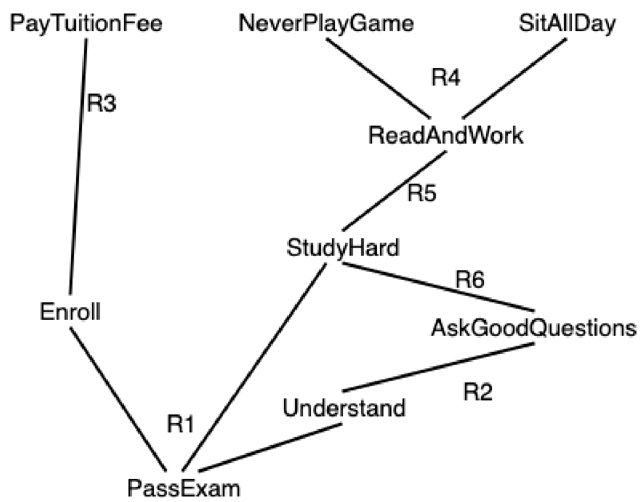
   - The final backed - up value is 11 and the recommended move is **to the right**. There were no α cut-offs, 2 β cut-offs, and 12 leaf nodes were generated.

10. [**Forward/backward chaining**] Apply forward and backward reasoning to the following Horn clauses to prove PassExam.

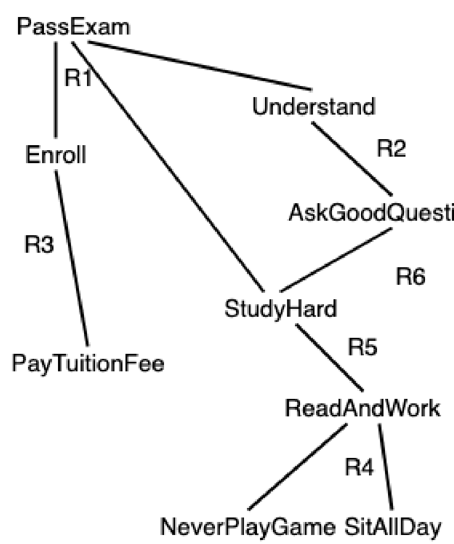
R1: $Enroll \wedge StudyHard \wedge Understand \Rightarrow PassExam$
R2: $AskGoodQuestions \Rightarrow Understand$
R3: $PayTuitionFee \Rightarrow Enroll$
R4: $NeverPlayGame \wedge SitAllDay \Rightarrow ReadAndWork$
R5: $ReadAndWork \Rightarrow StudyHard$
R6: $StudyHard \Rightarrow AskGoodQuestions$
$PayTuitionFee$
$NeverPlayGame$
$SitAllDay$

SOLUTION:

Forward chaining:



Backward chaining:



11. **[First order logic]** Assuming predicates $PARENT(p, q)$ and $FEMALE(p)$ and constants $Joan$ and $Kevin$, with the obvious meanings, express each of the following sentences in first-order logic. (You may use the abbreviation $\exists^1$ to mean "there exists exactly one.")

    (a) Joan has a daughter (possibly more than one, and possibly sons as well).
    SOLUTION: $\exists x\ PARENT(Joan, x) \wedge FEMALE(x)$.

    (b) Joan has exactly one daughter (but may have sons as well).
    SOLUTION: $\exists^1\ PARENT(Joan, x) \wedge FEMALE(x)$.

    (c) Joan has exactly one child, a daughter.
    SOLUTION: $\exists x\ PARENT(Joan, x) \wedge FEMALE(x) \wedge [\forall y\ PARENT(Joan, y) \Rightarrow y = x]$.

    (d) Joan and Kevin have exactly one child together.
    SOLUTION: $\exists^1 c\ PARENT(Joan, c) \wedge PARENT(Kevin(c)$.

    (e) Joan has at least one child with Kevin, and no children with anyone else.
    SOLUTION: $\exists c\ PARENT(Joan, c) \wedge PARENT(Kevin, c) \wedge \forall d \forall p\ [PARENT(Joan, d) \wedge PARENT(p, d)] \Rightarrow [p = Joan \vee p = Kevin]$

12. **[First order logic]** Consider a vocabulary with the following symbols:
    $OCCUPATION(p, o)$: Predicate. Person p has occupation o.
    $CUSTOMER(p1, p2)$: Predicate. Person $p1$ is a customer of person $p2$.
    $BOSS(p1, p2)$: Predicate. Person $p1$ is a boss of person $p2$.
    $Doctor, Surgeon, Lawyer, Actor$: Constants denoting occupation.
    $Emily, Joe$: Constants denoting people.

    Use these symbols to write the following assertions in First order logic:

    (a) Emily is either a surgeon or a lawyer.
    SOLUTION: $OCCUPATION(Emily, Surgeon) \vee OCCUPATION(Emily, Lawyer)$.

    (b) Joe is an actor, but he also holds another job.
    SOLUTION: $OCCUPATION(Joe, Actor) \wedge \exists p\ (p \neq Actor) \wedge OCCUPATION(Joe, p)$.

    (c) All surgeons are doctors.
    SOLUTION: $\forall p\ OCCUPATION(p, Surgeon) \Rightarrow OCCUPATION(p, Doctor)$.

    (d) Joe does not have a lawyer (i.e., is not a customer of any lawyer).
    SOLUTION: $\neg \exists p\ CUSTOMER(Joe, p) \wedge OCCUPATION(p, Lawyer)$.

    (e) Emily has a boss who is a lawyer.
    SOLUTION: $\exists p\ BOSS(p, Emily) \wedge OCCUPATION(p, Lawyer)$.

    (f) There exists a lawyer all of whose customers are doctors.
    SOLUTION: $\exists p\ OCCUPATION(p, Lawyer) \wedge \forall q\ CUSTOMER(q, p) \Rightarrow OCCUPATION(q, Doctor)$.

    (g) Every surgeon has a lawyer.
    SOLUTION: $\forall p\ OCCUPATION(p, Surgeon) \Rightarrow \exists q\ OCCUPATION(q, Lawyer) \wedge CUSTOMER(p, q)$.

13. **[First-order logic]** Indicate which of the following Predicate Calculus sentences are correct representations of the corresponding English sentences, and explain why or why not.

		SOLUTION
1	[$\forall x$ PERSON(x)] $\Rightarrow$ [$\exists y$ MOTHER(y,x)] Everybody has a mother	No, the scope of the x quantifier is incorrect.
2	OLD(DOG(Fido)) Fido is an old dog	No, we can't have a predicate of a predicate.
3	DOG(Fido) $\wedge$ OLD(Fido) Fido is an old dog	YES
4	$\forall x$ [MATHEMATICAL-THEORY(x) $\Rightarrow$ x] All mathematical theories are true	No, you can't imply a variable.
5	$\exists s$ [ARISTOTLE-SAID(s) $\wedge$ $\neg$TRUE(s)] Aristotle told a lie	YES
6	$\neg\exists v\ \exists b$ [BUTCHER(b) $\wedge$ VEGETARIAN(v)] There are no vegetarian butchers	No, the variables are different.
7	$\neg\exists b\ \exists d$ BUTCHER(b) $\wedge$ DOG(d) $\wedge$ OWNS(b,d) No butcher owns a dog	YES

14. **[First-order logic]** The expression $LAST(x,y)$ means that y is the last element of the list x. We have the following axioms:

    (a) $LAST(\{u,NIL\},u)$, where u is an element.

    (b) $LAST(y,z) \Rightarrow LAST(\{x,y\},z)$, where x and z are elements, and y is a list.

    Use the answer extraction method to find v, the last element of the list (2,1): $LAST(\{2,1,NIL\},v)$

    SOLUTION:
    The negated goal is: $\neg LAST(\{2,1,NIL\},v)$
    The disjunction of the goal and the negated goal is:
    P3. $\neg LAST(\{2,1,NIL\},v) \vee LAST(\{2,1,NIL\},v)$

    Converting wffs to clauses:

    P1. $LAST(\{u,NIL\},u)$

    P2. $\neg LAST(y,z) \vee LAST(\{x,y\},z)$

    Applying resolution:

    P3 and P2:
    P3. $\underline{\neg LAST(\{2,1,NIL\},v)} \vee LAST(\{2,1,NIL\},v)$ P2. $\neg LAST(y,z) \vee \underline{LAST(\{x,y\},z)}$
    mgu: $\{x|2,y|\{1,NIL\},z|v\}$
    resolvent: P4. $\neg LAST(\{1,NIL\},v) \vee LAST(\{2,1,NIL\},v)$

    P4 and P1:
    P1. $\underline{LAST(\{u,NIL\},u)}$ P4. $\underline{\neg LAST(\{1,NIL\},v)} \vee LAST(\{2,1,NIL\},v)$
    mgu: $\{u|1,v|1\}$
    resolvent: nil $\vee$ $LAST(\{2,1,NIL\},1)$

    Thus, the last element of this list $\{2,1,NIL\}$ is 1.

15. **[Unification]** For each of the following pairs of literals indicate whether or not they unify. If they do unify, show a most general unifier. If they do not unify, explain why not.

    (a) STRONGER(x, LexLuthor), STRONGER(LoisLane,y)
    SOLUTION: mgu = {x|LoisLane,y|LexLuthor}

    (b) HOLDING(x, Girlfriend(x)), HOLDING(Superman,y)
    SOLUTION: mgu = {x|Superman,y|Girlfriend(Superman)}

    (c) FLY(Girlfriend(x)), ¬FLY(LoisLane)
    SOLUTION: No, you can't unify a predicate with its negation, and you can't unify a constant with a function.

    (d) ¬LIKES(x,Enemy(x)), ¬LIKES(Enemy(y), y)
    SOLUTION: No, you can substitute x|Enemy(y), which yields
    ¬LIKES(Enemy(y),Enemy(Enemy(y))).
    However, you can't substitute a function that contains y for y itself.

16. **[Unification]** For each pair of atomic sentences, give the most general unifier if it exists. If they do not unify, explain why not.

    (a) $P(A, B, B), P(x, y, z)$.
    SOLUTION: $x/A, y/B, z/B$

    (b) $Q(y, G(A, B)), Q(G(x, x), y)$.
    SOLUTION: No unifier. These two sentences can firstly use $y|G(x, x)$, then they become $Q(G(x, x), G(A, B))$ and $Q(G(x, x), G(x, x))$. They cannot unify because x cannot bind to both A and B.

    (c) $OLDER(Father(y), y), OLDER(FATHER(x), John)$.
    SOLUTION: $y/John, x/John$.

    (d) $KNOWS(Father(y), y), KNOWS(x, x)$.
    SOLUTION: No unifier, because we first unify $x|Father(y)$, yielding
    $KNOWS(Father(y), y), KNOWS(Father(y), Father(y))$. Now, the second y cannot unify with $Father(y)$.

17. Consider the following statements:

```
1. All hounds howl at night.
2. Anyone who has any cats will not have any mice.
3. Light sleepers do not have anything which howls at night.
4. John has either a cat or a hound.
5. If John is a light sleeper, then John does not have any mice.
```

    (a) Using only the predicates $HOUND$, $HOWL$, $HAVE$, CAT, $MOUSE$ and $LIGHTSLEEPER$, represent these statements as well formed formulas (wffs).
    SOLUTION:
    1. $\forall x\ (HOUND(x) \Rightarrow HOWL(x))$
    2. $\forall p \forall y\ (HAVE(p, y) \wedge CAT(y) \Rightarrow \neg \exists z\ HAVE(p, z) \wedge MOUSE(z))$
    3. $\forall p\ (LIGHTSLEEPER(p) \Rightarrow \neg \exists x\ (HAVE(p, x) \wedge HOWL(x)))$
    4. $\exists w\ (HAVE(John, w) \wedge (CAT(w) \vee HOUND(w)))$
    5. $LIGHTSLEEPER(John) \Rightarrow \neg \exists z\ (HAVE(John, z) \wedge MOUSE(z))$

(b) Convert Statements 1-4 to clauses.

SOLUTION:

1. $\neg HOUND(x1) \vee HOWL(x1)$
2. $\neg HAVE(p2, y2) \vee \neg CAT(y2) \vee \neg HAVE(p2, z2) \vee \neg MOUSE(z2)$
3. $\neg LIGHTSLEEPER(p3) \vee \neg HAVE(p3, x3) \vee \neg HOWL(x3)$
4. $HAVE(John, A) \wedge (CAT(A) \vee HOUND(A))$

(c) Use resolution to prove Statement 5.

SOLUTION:

Negate Statement 5 and convert to clause:

$\neg(LIGHTSLEEPER(John) \Rightarrow \neg\exists z\ (HAVE(John, z) \wedge MOUSE(z)))$

$\neg(\neg LIGHTSLEEPER(John) \vee \neg\exists z\ (HAVE(John, z) \wedge MOUSE(z)))$

$LIGHTSLEEPER(John) \wedge \exists z\ (HAVE(John, z) \wedge MOUSE(z))$

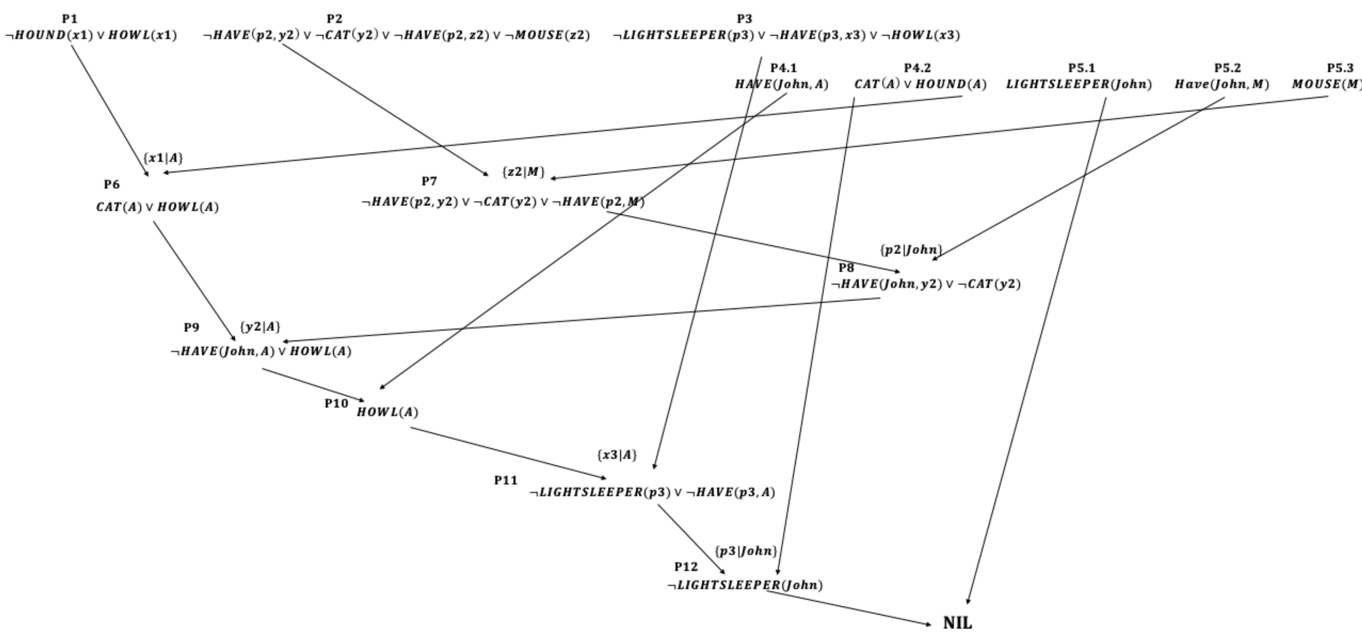
Skolemize: $LIGHTSLEEPER(John) \wedge HAVE(John, M) \wedge MOUSE(M)$

Clauses:

5.1 $LIGHTSLEEPER(John)$

5.2 $HAVE(John, M)$

5.3 $MOUSE(M)$



18. **[Resolution]** From "Horses are animals", it follows that "The head of a horse is the head of an animal." Demonstrate that this inference is valid by carrying out the following steps:

(a) Translate the premise and the conclusion into a well formed formula (wff) in First order logic. Use three predicates: $HEADOF(h, x)$ (meaning "h is the head of x"), $HORSE(x)$, and $ANIMAL(x)$.

SOLUTION:

Premise: $\forall x\ HORSE(x) \Rightarrow ANIMAL(x)$

Conclusion: $\forall x \forall h\ HORSE(x) \wedge HEADOF(h, x) \Rightarrow \exists y\ ANIMAL(y) \wedge HEADOF(h, y)$

(b) Negate the conclusion, and convert the premise and the negated conclusion into CNF.

SOLUTION:

Premise: $\neg HORSE(x) \vee ANIMAL(x)$

Negated Conclusion:

$\neg(\forall x \forall h\ HORSE(x) \wedge HEADOF(h,x) \Rightarrow \exists y\ ANIMAL(y) \wedge HEADOF(h,y))$

$\neg(\forall x \forall h \neg(HORSE(x) \wedge HEADOF(h,x)) \vee \exists y\ ANIMAL(y) \wedge HEADOF(h,y))$

$(\exists x \exists h \neg\neg(HORSE(x) \wedge HEADOF(h,x)) \wedge \neg\exists y\ ANIMAL(y) \wedge HEADOF(h,y))$

$\exists x \exists h (HORSE(x) \wedge HEADOF(h,x)) \wedge \forall y \neg(ANIMAL(y) \wedge HEADOF(h,y))$

$\exists x \exists h (HORSE(x) \wedge HEADOF(h,x)) \wedge \forall y (\neg ANIMAL(y) \vee \neg HEADOF(h,y))$

Remove universal quantifier and Skolemization:

$(HORSE(G) \wedge HEADOF(H,G)) \wedge (\neg ANIMAL(y) \vee \neg HEADOF(H,y))$

Clauses:

C1. $HORSE(G)$

C2. $HEADOF(H,G)$

C3. $\neg ANIMAL(y) \vee \neg HEADOF(H,y)$

(c) Use resolution to show that the conclusion follows from the premise.

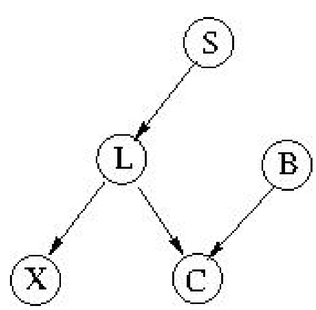
SOLUTION:

Resolve C3 and C2 (substitution $y|G$): R1. $\neg ANIMAL(G)$

Resolve R1 with Premise (substitution $x|G$): R2. $\neg HORSE(G)$

Resolve R2 with C1: nil (contradiction)

19. **[Bayesian Reasoning]** Consider the following Bayesian network for another version of the medical diagnosis example, where *B=Bronchitis, S=Smoker, C=Cough, X=Positive X-ray* and *L=Lung cancer*, and all nodes are Boolean. Suppose that the prior for a patient being a smoker is 0.25, and the prior for the patient having bronchitis is 0.05. List the pairs of nodes that are conditionally independent in the following situations, and explain why.



(a) There is no evidence for any of the nodes.
SOLUTION: L and B (C is common effect), S and B (C is common effect between L and B), X and B (C is common effect between L and B).

(b) The lung cancer node (L) is set to true (and there is no other evidence).
SOLUTION: S and X (chain), S and C (chain), S and B (chain to C), X and C (common cause), X and B (common cause with C).

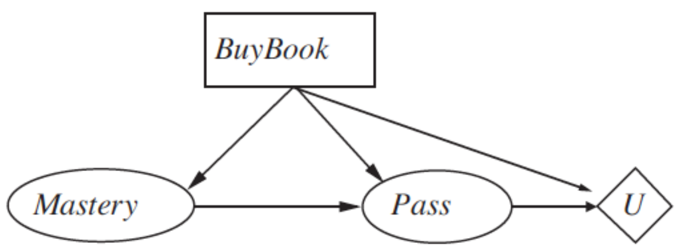
(c) The smoker node (S) is set to true (and there is no other evidence).
SOLUTION: L and B (C is common effect), X and B (C is common effect between L and B). Note that S and B are not mentioned because S is an evidence node, and it is no longer a variable.

(d) The cough node (C) is set to true (and there is no other evidence).
SOLUTION: There are no conditional independencies in this situation.

20. [**Bayesian Network**] Consider a student who has the choice to buy or not buy a textbook for a course. We'll model this as a decision problem with one Boolean decision Node, B, indicating whether the student chooses to buy the book, and two Boolean chance nodes, M, indicating whether the student has mastered the material in the book, and P, indicating whether the student passes the course. Of course, there is also an *additive* utility node, U. A certain student, Sam, has a utility function: 0 for not buying the book and -\$100 for buying it; and \$2000 for passing the course and 0 for not passing. In this case, the additive utility for passing and buying the book is $2000 - 100$.

Sam's conditional probability estimates are as follows:
$Pr(p|b, m) = 0.9$ $\quad Pr(m|b) = 0.9$
$Pr(p|b, \neg m) = 0.5$ $\quad Pr(m|\neg b) = 0.7$
$Pr(p|\neg b, m) = 0.8$
$Pr(p|\neg b, \neg m) = 0.3$
You might think that P would be independent of B given M, but this course has an open-book final – so having the book helps.

(a) Draw the decision network for this problem.
SOLUTION:



(b) Compute the expected utility of buying the book and of not buying it.
SOLUTION: For each of $B = b$ and $B = \neg b$, we compute $Pr(p|B)$ and thus $Pr(\neg p|B)$ by marginalizing out M, then use this to compute the expected utility.

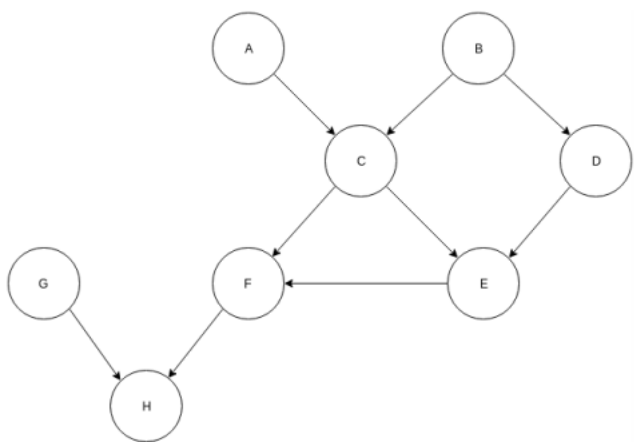
$$
\begin{aligned}
Pr(p|b) &= \sum_m Pr(p|b,m) \times Pr(m|b) \\
&= 0.9 \times 0.9 + 0.5 \times 0.1 \\
&= 0.86 \\
Pr(p|\neg b) &= \sum_m Pr(p|\neg b,m) \times Pr(m|\neg b) \\
&= 0.8 \times 0.7 + 0.3 \times 0.3 \\
&= 0.65
\end{aligned}
$$

The expected utilities are thus:

$$
\begin{aligned}
EU[b] &= \sum_p Pr(p|b) \times U(p,b) \\
&= 0.86 \times (2000 - 100) + 0.14 \times (-100) \\
&= 1620 \\
EU[\neg b] &= \sum_p Pr(p|\neg b) \times U(p,\neg b) \\
&= 0.65 \times 2000 + 0.35 \times 0 \\
&= 1300
\end{aligned}
$$

(c) What should Sam do?
SOLUTION: Buy the book, Sam.

21. **[D-separation]** According to this Figure, determine whether the following claims to be True or False and justify your answer.



(a) $B \perp G|A$
SOLUTION: True, they are independent. Without knowing H (common effect), the information of B cannot propagate to G. It does not matter whether A is known.

(b) $C \perp D|F$
SOLUTION: False, they are not independent. Knowing F allows the information in C to propagate through E (common effect), then propagate to D (common effect).

(c) $C \perp D|A$
SOLUTION: False, they are not independent. Without knowing B (common cause), the information in C can propagate to D. Note that if there was evidence for E (common effect), information could propagate from C to D through E.

However, in this situation we have no evidence for E, hence this path is blocked. Also note that it does not matter whether A is known.

(d) $H \perp B | C, F$
SOLUTION: True, they are independent. Knowing C and F (chain rule) blocks the information between H and B.

22. **[Supervised Machine Learning]** Consider the following dataset consisting of 3 binary features and a binary class variable.

Feature 1	Feature 2	Feature 3	Class
1	0	0	0
1	0	1	1
1	1	1	1
1	1	1	1
0	1	1	0
0	0	1	0

(a) Based on the dataset above, estimate the prior probability of Class 1.
SOLUTION: $\Pr(\text{Class} = 1) = 3/6 = 0.5$

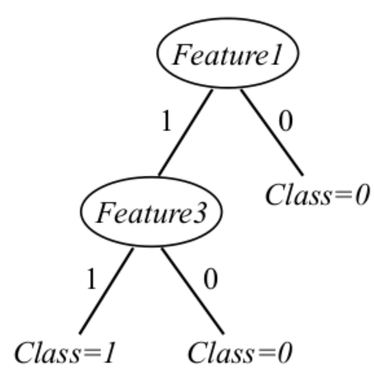
(b) What is the initial entropy of the class labels over all the data? Show the formula before plugging-in numbers.
SOLUTION:
$H(\text{Class}) = -\Pr(\text{Class=1}) \log_2 \Pr(\text{Class=1}) - \Pr(\text{Class=0}) \log_2 \Pr(\text{Class=0})$
$H(\text{Class}) = -0.5 \log(0.5) - 0.5 \log(0.5) = 1$

(c) If we were to split the data into 2 groups based on the value of Feature 1, what would be the entropy for each group?
SOLUTION:
$H(\text{Class}|\text{Feature1=1}) = -(3/4) \log(3/4) - (1/4) \log(1/4) = .81$
$H(\text{Class}|\text{Feature1=0}) = -(1) \log(1) - (0) \log(0) = 0$

(d) What is the Information Gain of Feature 1?
SOLUTION:
$IG(\text{Feature1})$

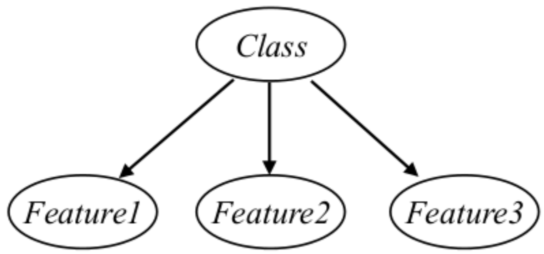
$$\begin{aligned} &= H(\text{Class}) - (\Pr(F1 = 1)H(\text{Class}|F1 = 1) + \Pr(F1 = 0)H(\text{Class}|F1 = 0)) \\ &= 1 - ((4/6) \times 0.81 + (2/6) \times 0) = (1 - 0.54) = 0.46 \end{aligned}$$

(e) You are told the Information Gain for Feature 2 and Feature 3 **after splitting on Feature1** is 0.08 and 0.19 respectively. Draw a decision tree for this problem.
SOLUTION:



(f) We will now build a Naïve Bayes model for predicting the class from these data. Draw the Bayesian Network that corresponds to the Naïve Bayes model.

SOLUTION:



(g) From the data table, write down the conditional probability estimates required for the Naïve Bayes model.

SOLUTION:

Pr(Feature1 =1|Class=1) = 3/3 = 1
Pr(Feature1 =1|Class=0) = 1/3

Pr(Feature2 =1|Class=1) = 2/3
Pr(Feature2 =1|Class=0) = 1/3

Pr(Feature3 =1|Class=1) = 3/3 = 1
Pr(Feature3 =1|Class=0) = 2/3

(h) What class does the Naïve Bayes model predict for a new data item with values (1, 1, 1) for Features 1, 2 and 3 respectively? What is the probability of each class? How does the Naïve Bayes classification compare to the decision tree classification?

SOLUTION:

$$\begin{aligned}
&\Pr(C=1|F1=1, F2=1, F3=1)\\
&\quad = \alpha \Pr(F1=1|C=1)\Pr(F2=1|C=1)\Pr(F3=1|C=1)\Pr(C=1)\\
&\quad = \alpha \times 1 \times 2/3 \times 1 \times 1/2 = 1/3 \times \alpha
\end{aligned}$$

$$\begin{aligned}
&\Pr(C=0|F1=1, F2=1, F3=1)\\
&\quad = \alpha \Pr(F1=1|C=0)\Pr(F2=1|C=0)\Pr(F3=1|C=0)\Pr(C=0)\\
&\quad = \alpha \times 1/3 \times 1/3 \times 2/3 \times 1/2 = 1/27 \times \alpha
\end{aligned}$$

$\alpha = \frac{1}{1/3+1/27} = 2.7$
$\Pr(C=1|F1=1, F2=1, F3=1) = 2.7 \times 1/3 = 0.9$
$\Pr(C=0|F1=1, F2=1, F3=1) = 2.7 \times 1/27 = 0.1$

So Naïve Bayes assigns Class 1 to case (1,1,1), which is the same as the decision tree.

23. **[Decision Tree]** Assume that you are given the set of labeled *training examples* below, where each of three features has three possible values: a, b or c. You choose to learn a decision tree from these data.

	F1	F2	F3	Output
ex1	c	b	b	+
ex2	a	a	c	+
ex3	b	c	c	+
ex4	b	c	a	-
ex5	a	b	c	-
ex6	c	a	b	-

What score would the *information gain* calculation assign to feature F1, when deciding which feature to use as the <u>root</u> node for a decision tree being built? **Be sure to show all your work**.

SOLUTION:
$IG(S, F1) = H(S) - \frac{2}{6}H(S_{F1=a}) - \frac{2}{6}H(S_{F1=b}) - \frac{2}{6}H(S_{F1=c})$
where $H(S) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$
$H(S_{F1=a}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$
$H(S_{F1=b}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$
$H(S_{F1=c}) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$
$IG(S, F1) = 1 - \frac{2}{6} \times 1 - \frac{2}{6} \times 1 - \frac{2}{6} \times 1 = 0$

24. **[Naïve Bayes]** Consider a traffic-monitoring agent trying decide whether the stop-lights at an intersection are working or not ($W = w$ or $\neg w$). The agent observes two variables, the East-West light's color EW, which is either red or green, and the North-South light's color NS, which is also either red or green. When the lights are functioning, exactly one of the lights is green, while when the lights are broken, both are red (and flashing, but the agent cannot perceive this distinction). The agent has several observations as training data:

NS	EW	W
r	g	w
g	r	w
r	r	$\neg w$
g	r	w
r	g	w
r	g	w
g	r	w

(a) Assume the agent uses a Naïve Bayes classifier to make its prediction. What are the parameters of the model?
SOLUTION:
$Pr(W), Pr(NS|W), Pr(EW|W)$

(b) What is the Maximum Likelihood Estimator (MLE) for the parameters?
SOLUTION:
$Pr(W = w) = 6/7$, $Pr(W = \neg w) = 1 - 6/7 = 1/7$
$Pr(NS = r|W = w) = 3/6 = 1/2$, $Pr(NS = g|W = w) = 1 - 1/2 = 1/2$
$Pr(NS = r|W = \neg w) = 1$, $Pr(NS = g|W = \neg w) = 1 - 1 = 0$
$Pr(EW = r|W = w) = 3/6 = 1/2$, $Pr(NS = g|W = w) = 1 - 1/2 = 1/2$
$Pr(EW = r|W = \neg w) = 1$, $Pr(EW = g|W = \neg w) = 1 - 1 = 0$

(c) Assume the agent observes that both lights are red. What is the probability that the lights are working according to the agent's model?
SOLUTION:
$Pr(W = w|NS = r, EW = r)$
$= \alpha \times Pr(NS = r|W = w) \times Pr(EW = r|W = w) \times Pr(W = w)$
$= \alpha \times 1/2 \times 1/2 \times 6/7 = \alpha \times 3/14$
$Pr(W = \neg w|NS = r, EW = r)$
$= \alpha \times Pr(NS = r|W = \neg w) \times Pr(EW = r|W = \neg w) \times Pr(W = \neg w)$
$= \alpha \times 1 \times 1 \times 1/7 = \alpha \times 1/7$
$\therefore \alpha = 14/5, Pr(W = w|NS = r, EW = r) = 3/5 = 0.6$

25. **[K-nearest-neighbour]** Assume we have the age, loan amount and pay result of some customer from the bank. We need to predict the pay result of a new loan applicant given only age and loan amount. The data are shown below:

Age	Loan Amount (K)	Pay Status	Distance to new customer
20	20	Paid	$\sqrt{(20-42)^2 + (20-140)^2} = 122$
25	45	Paid	$\sqrt{(25-42)^2 + (45-140)^2} = 96.51$
22	95	Default	$\sqrt{(22-42)^2 + (95-140)^2} = 49.24$
35	53	Paid	$\sqrt{(35-42)^2 + (53-140)^2} = 87.28$
35	**120**	**Paid**	$\boldsymbol{\sqrt{(35-42)^2 + (120-140)^2} = 21.19}$
33	**150**	**Default**	$\boldsymbol{\sqrt{(33-42)^2 + (150-140)^2} = 13.45}$
40	54	Default	$\sqrt{(40-42)^2 + (54-140)^2} = 86.02$
45	75	Paid	$\sqrt{(45-42)^2 + (75-140)^2} = 65.07$
52	20	Paid	$\sqrt{(52-42)^2 + (20-140)^2} = 120.42$
49	220	Default	$\sqrt{(49-42)^2 + (220-140)^2} = 80.31$
60	**100**	**Default**	$\boldsymbol{\sqrt{(60-42)^2 + (100-140)^2} = 43.86}$

A new bank customer with age 42 comes in and applies for a loan AUD$140K, how would the 3-nearest-neighbour algorithm predict this customer? Paid or Default? **Be sure to show all your work**.

SOLUTION: According to the 3 nearest neighbours (bold-fonted in the table), the algorithm will predict this new customer Default the loan.

26. **[K-means clustering]** Consider the k-means clustering algorithm when answering the following questions.

(a) What are the two steps performed at each iteration of the clustering algorithm?
SOLUTION:
Step 1. Reassign data points to their closest cluster (centre).
Step 2. Recompute the centre (mean) of each cluster.

(b) Is the algorithm guaranteed to converge (stop) at some point? And if so, does it converge to a global optimum or a local optimum?

SOLUTION:

Yes, convergence is guaranteed, but to a *local* optimum. Other techniques are required to achieve global optimum.

(c) Choosing the "right number" of clusters for k-means clustering can be difficult. One strategy would be to choose the number of clusters that minimises the sum of the (squared) distances between data points and their cluster centres. Would this be a good strategy? Why?

SOLUTION:

No, it is not a good strategy, since the "optimal" number of clusters (i.e., the one that minimizes the sum of the distances to the cluster means) would always be n, the number of datapoints.

(d) Before running a clustering algorithm it is important to preprocess your data. What preprocessing is most important and why?

SOLUTION:

Must rescale your features/dimensions to have the same range, otherwise the distance measure will be controlled by the dimension with the largest scale.

27. [**K-means clustering**] Assume we have following data with two attributes, apply the K-means clustering algorithm where K=2 and initialized cluster centroid C1=(1.25, 1.25) and C2=(1.75,4.75). You should use Euclidean distance to perform your calculations. **Be sure to show all your work**.

Instance	D1	D2
X1	1.0	1.5
X2	1.0	4.5
X3	2.0	1.5
X4	2.0	3.5
X5	3.0	2.5
X6	5.0	6.0
X7	4.0	5.5
X8	6.0	1.5

SOLUTION:

	C1	1.25	1.25		
	C2	1.75	4.75		
Iteration 1					
Instance	D1	D2	Distance to C1	Distance to C2	Cluster
X1	1	1.5	0.353553391	3.335416016	C1
X2	1	4.5	3.259601203	0.790569415	C2
X3	2	1.5	0.790569415	3.259601203	C1
X4	2	3.5	2.371708245	1.274754878	C2
X5	3	2.5	2.150581317	2.573907535	C1
X6	5	6	6.051859218	3.482097069	C2
X7	4	5.5	5.062114183	2.371708245	C2
X8	6	1.5	4.756574398	5.35023364	C1
Clusters={X1, X3, X5, X8}, {X2, X4, X6, X7}					
Update centroid	C1	3	1.75		
	C2	3	4.875		
Iteration 2					
Instance	D1	D2	Distance to C1	Distance to C2	Cluster
X1	1	1.5	2.015564437	3.923088707	C1
X2	1	4.5	3.400367627	2.034852575	C2
X3	2	1.5	1.030776406	3.52003196	C1
X4	2	3.5	2.015564437	1.700183814	C2
X5	3	2.5	0.75	2.375	C1
X6	5	6	4.697073557	2.294694969	C2
X7	4	5.5	3.881043674	1.179247642	C2
X8	6	1.5	3.010398645	4.515597967	C1
Clusters={X1, X3, X5, X8}, {X2, X4, X6, X7}					
Clusters are the same for iteration 1 and 2, STOP					