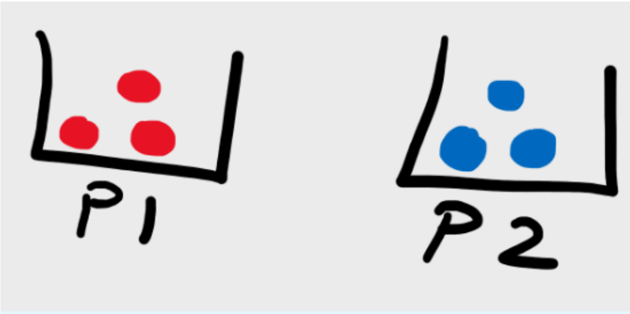
Artificial Intelligence I: Assignment 4

Name : *Saranya Balasubramaniyan*
Net Id: *sxb9644*
Course and section : *CSE 5360 - 900*

Task 1

Consider the following scenario:



Your task is to get all the blue marbles in P1 and red marbles in P2

The actions available are as follows:
You can move 2 marbles from P1 to P2 if they are each of different colors (one blue or one red)
You can move 2 marbles from P2 to P1 if they are of the same color (either you can move 2 red or you can move 2 blue)
Give the PDDL description to represent the above as a Planning Problem. Do not forget to also define and describe the predicates and constants that your are going to use.

Extra Credit (10 pts): Also, give a complete plan (using the actions described) for getting from the start to the goal state

Constants :
3 Red marbles are represented by constants : **R1, R2, R3**
3 blue marble are represented by constant s: **B1, B2, B3**
Containers : P1 and P2

Constants are R1, R2, R3, B1, B2, B3, P1 and P2

Predicates:
Contains(x, a) where x is the container and a represent marble. The predicate returns true if the specified container contains the marbles a.
SameColorAs(a,b) where a and b represent marbles. The predicate returns true of both a and b are of the same color, false otherwise.

State Description:
Initial state:

Contains(P1, R1) and Contains(P1, R2) and Contains(P1, R3) and Contains(P2, B1) and Contains(P2, B2) and Contains (P2, B3)

Goal state:

Contains(P1, B1) and Contains(P1, B2) and Contains(P1, B3) and Contains(P2, R1) and Contains(P2, R2) and Contains (P2, R3)

Action description:

MOVE_P1toP2(from, to, marble1, marble2)
PRECOND : Not(SameColorAs(marble1, marble2)) and Contains(from,marble1) and Contains(from,marble2)
EFFECT: Contains(to, marble1) and Contains(to, marble2) and not(Contains(from, marble1)) and not(Contains(from, marble2))

MOVE_P2toP1(from, to, marble1, marble2)
PRECOND : SameColorAs(marble1, marble2) and Contains(from,marble1) and Contains(from,marble2)
EFFECT: Contains(to, marble1) and Contains(to, marble2) and not(Contains(from, marble1)) and not(Contains(from, marble2))

Complete plan for getting from start to goal state

Steps	Action performed at each step	P1	P2
Initial State	Contains(P1, R1) and Contains(P1, R2) and Contains(P1, R3) and Contains(P2, B1) and Contains(P2, B2) and Contains (P2, B3) Is this the goal state? No	R1 R2 R3	B1 B2 B3
Step 1	**Action**: MOVE_P2toP1(P2, P1, B1, B2) **PRECOND :** SameColorAs(B1, B2) – true Contains(P2,B1) – true Contains(P2,B2) – true **EFFECT:** Contains(P1, B1) and Contains(P1, B2) and not(Contains(P2, B1)) and not(Contains(P2, B2)) **Current state:** Contains(P1, R1) and Contains(P1, R2) and Contains(P1, R3) and Contains(P1, B1) and Contains(P1, B2) and Contains (P2, B3) Is this the goal state? No	R1 R2 R3 B1 B2	B3
Step 2	**Action**: MOVE_P1toP2(P1, P2, R1, B1) **PRECOND :** Not(SameColorAs(R1, B1))– true Contains(P1,R1) – true Contains(P1,B1) – true **EFFECT:** Contains(P2, R1) and Contains(P2, B1) and not(Contains(P1, R1)) and not(Contains(P1, B1)) **Current state:** Contains(P1, R2) and Contains(P1, R3) and Contains(P1, B2) and Contains(P2, R1) and Contains(P2, B1) and Contains (P2, B3) Is this the goal state? No	R2 R3 B2	R1 B1 B3
Step 3	**Action**: MOVE_P1toP2(P1, P2, R2, B2) **PRECOND :** Not(SameColorAs(R2, B2))– true Contains(P1,R2) – true Contains(P1,B2) – true	R3	R1 R2 B1 B2 B3

	EFFECT: Contains(P2, R2) and Contains(P2, B2) and not(Contains(P1, R2)) and not(Contains(P1, B2)) **Current state:** Contains(P1, R3) and Contains(P2, R1) and Contains(P2, R2) and Contains(P2, B1) and Contains(P2, B2) and Contains (P2, B3) Is this the goal state? No		
Step 4	**Action**: MOVE_P2toP1(P2, P1, B1, B2) **PRECOND :** SameColorAs(B1, B2) – true Contains(P2,B1) – true Contains(P2,B2) – true **EFFECT:** Contains(P1, B1) and Contains(P1, B2) and not(Contains(P2, B1)) and not(Contains(P2, B2)) **Current state:** Contains(P1, R3) and Contains(P1, B1) and Contains(P1, B3) and Contains(P2, R1) and Contains(P2, R2) and Contains (P2, B3) Is this the goal state? No	R3 B1 B2	R1 R2 B3
Step 5	**Action**: MOVE_P1toP2(P1, P2, R3, B2) **PRECOND :** Not(SameColorAs(R3, B2))– true Contains(P1,R3) – true Contains(P1,B2) – true **EFFECT:** Contains(P2, R3) and Contains(P2, B2) and not(Contains(P1, R3)) and not(Contains(P1, B2)) **Current state:** Contains(P1, B1) and Contains(P2, R1) and Contains(P2, R2) and Contains(P2, R3) and Contains(P2, B2) and Contains (P2, B3) Is this the goal state? No	B1	R1 R2 R3 B2 B3
Step 6	**Action**: MOVE_P2toP1(P2, P1, B2, B3) **PRECOND :** SameColorAs(B2, B3) – true Contains(P2,B2) – true Contains(P2,B3) – true **EFFECT:** Contains(P1, B2) and Contains(P1, B3) and not(Contains(P2, B2)) and not(Contains(P2, B3)) **Current state:** Contains(P1, B1) and Contains(P1, B2) and Contains(P1, B3) and Contains(P2, R1) and Contains(P2, R2) and Contains (P2, R3) Is this the goal state? Yes Goal state achieved	B1 B2 B3	R1 R2 R3

Task 2

Suppose that we are using PDDL to describe facts and actions in a certain world called JUNGLE. In the JUNGLE world there are 4 predicates, each predicate takes at most 3 arguments, and there are 5 constants. Give a reasonably tight bound on the number of unique states in the JUNGLE world. Justify your answer

Number of Predicates : 4
Maximum number of arguments in the predicate: 3
Number of Constants : 5

Considering the fact that predicates usually result in true or false, we know that the number of unique states would be 2^n where n is the number of combinations that can be defined in the predicates.

The number of states can be determined based on the number of arguments each predicate takes. In this scenario, to get a tight bound in the JUNGLE world, the maximum number of states (upper bound) can be determined by assuming that the predicate takes the maximum possible arguments.
The minimum number of states can be determined by assuming that the predicate takes the minimum possible number of arguments.

Upper Bound/ maximum number of unique states in the JUNGLE world:

A single predicate can take a maximum of 3 arguments. So number of possible combinations for a single predicate is **5 * 5 * 5 = 125**
Since there are 4 predicates, there can be **4 * 125 = 500** possible combinations.

Since predicates result in true or false, the upper bound for JUNGLE world would be $\mathbf{2^{500}}$

Lower Bound/ minimum number of unique states in the JUNGLE world:

If the predicate takes the minimum number of predicates, which is 1, then the possible number of combinations is **5**.
Since there are 4 predicates, minimum number of predicates for available combinations is **4 * 5 = 20**

Since predicates result in true or false, the lower bound for JUNGLE world would be $\mathbf{2^{20}}$

Reasonably tight bound for the number of unique states in JUNGLE world is [2^{20} - 2^{500}]

Task 3

Consider the given joint probabilty distribution for a domain of two variables (Color, Vehicle) :

	Color = Red	Color = Green	Color = Blue
Vehicle = Car	0.0630	0.1080	0.1290
Vehicle = Van	0.0441	0.0756	0.0903
Vehicle = Truck	0.0504	0.0864	0.1032
Vehicle = SUV	0.0525	0.0900	0.1075

Part a: Calculate P (Color is not Green | Vehicle is Truck) by Inference by Enumeration

	Color = Red	Color = Green	Color = Blue
Vehicle = Car	0.0630	0.1080	0.1290
Vehicle = Van	0.0441	0.0756	0.0903
Vehicle = Truck	0.0504	0.0864	0.1032
Vehicle = SUV	0.0525	0.0900	0.1075

P(˥Green | Truck) = α <P(˥Green , Truck)   P(Green, Truck)>
= α<(0.0504+0.1032)   (0.0864)>
= α<0.1536   0.0864)

(0.1536/ x) +(0.0864/ x) = 1
0.24/ x = 1
x = 0.24
The value of α is 1/0.24 = 4.16667

P(˥Green | Truck) = α <P(˥Green , Truck)   P(Green, Truck)>
= 4.16667 <0.1536   0.0864>
=<0.64   0.36>

Through Inference by enumeration,
P(˥Green | Truck) = 0.64

Part b: Are Vehicle and Color totally independant from each other? Justify.

Vehicle and Color are totally independent from each other. An event that the Vehicle being a car or a truck or van or an SUV is not going to affect the color of the vehicle. Hence both vehicle and color are independent

To prove total Independence,
if P(A) = P(A|B), then events A and B are independent

Let us consider an example,
Vehicle Car as event A and Color Red as event B

Let us verify if P(Car) = P(Car|Red)

Calculating P(Car), Based on the joint distribution table,

	Color = Red	**Color = Green**	**Color = Blue**
Vehicle = Car	0.0630	0.1080	0.1290
Vehicle = Van	0.0441	0.0756	0.0903
Vehicle = Truck	0.0504	0.0864	0.1032
Vehicle = SUV	0.0525	0.0900	0.1075

P(Car) = 0.0630 + 01080 + 0.1290 = 0.3

Calculating P(Car|Red),

Based on conditional probability,
P(Car|Red) = P(Car ^ Red)/P(Red)

	Color = Red	**Color = Green**	**Color = Blue**
Vehicle = Car	0.0630	0.1080	0.1290
Vehicle = Van	0.0441	0.0756	0.0903
Vehicle = Truck	0.0504	0.0864	0.1032
Vehicle = SUV	0.0525	0.0900	0.1075

P(Car|Red) = 0.0630/ (0.0630 +0.0441 + 0.0504 +0.0525)
= 0.0630/0.21
= 0.3

Hence P(Car) = P(Car|Red) = 0.3

Therefore, with this illustration, we can prove that the events Vehicle and Color are totally independent events.

Task 4

In a certain probability problem, we have 12 variables: A, B1, B2, ..., B10, C.
Variable A has 8 possible values.
Each of variables B1, ..., B10 have 5 possible values. Each Bi is conditionally indepedent of all other 9 Bj variables (with j != i) given A.
Variable C has 6 possible values. Variable C is totally independant of all other variables in the domain.
Based on these facts:

Part a: How many numbers do you need to store in the joint distribution table of these 12 variables?

Since there are 12 variables, the joint distribution table would be 12 dimensional.

Possible values of A : **8**
Possible values of B_1 to B_{10} : $\mathbf{5^{10}}$
Possible values of C: **6**

Hence, to store in the joint distribution table, numbers needed would be $\mathbf{8 * 5^{10} * 6 = 468{,}750{,}000}$

Part b: What is the most space-efficient way (in terms of how many numbers you need to store) representation for the joint probability distribution of these 12 variables? How many numbers do you need to store in your solution?

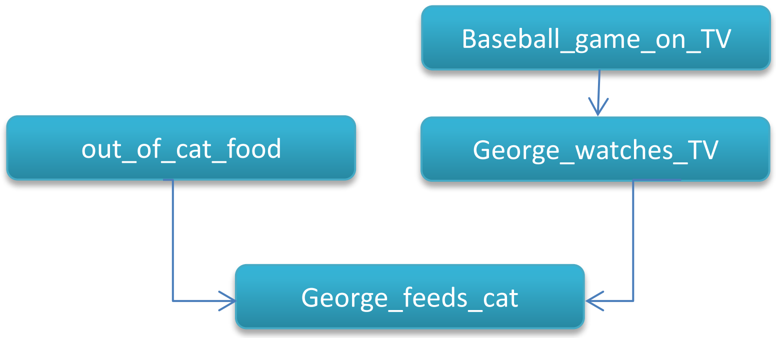
For variable A, with 8 possible values, values needed would be 8 – 1 = 7
For variable B_i with 5 possible values, that is conditionally dependent on A with 8 variables, values needed would be 8 * (5-1) = 32
So, for all the 10 B_i variables, the values needed would be 10 * 32 = 320
Since C is totally independent with 6 possible values, values needed for storage would be 6-1 =5

So total space needed would be 7 + 320 + 5 = **332**

Task 5

George doesn't watch much TV in the evening, unless there is a baseball game on. When there is baseball on TV, George is very likely to watch. George has a cat that he feeds most evenings, although he forgets every now and then. He's much more likely to forget when he's watching TV. He's also very unlikely to feed the cat if he has run out of cat food (although sometimes he gives the cat some of his own food). Design a Bayesian network for modeling the relations between these four events:

baseball_game_on_TV
George_watches_TV
out_of_cat_food
George_feeds_cat
Your task is to connect these nodes with arrows pointing from causes to effects. No programming is needed for this part, just include an electronic document (PDF, Word file, or OpenOffice document) showing your Bayesian network design.



Task 6

For the Bayesian network of previous task, the text file at this link contains training data from every evening of an entire year. Every line in this text file corresponds to an evening, and contains four numbers. Each number is a 0 or a 1. In more detail:

The first number is 0 if there is no baseball game on TV, and 1 if there is a baseball game on TV.
The second number is 0 if George does not watch TV, and 1 if George watches TV.
The third number is 0 if George is not out of cat food, and 1 if George is out of cat food.
The fourth number is 0 if George does not feed the cat, and 1 if George feeds the cat.
Based on the data in this file, determine the probability table for each node in the Bayesian network you have designed for Task 5. You need to include these four tables in the drawing that you produce for Task 5. You also need to submit the code/script that computes these probabilities.

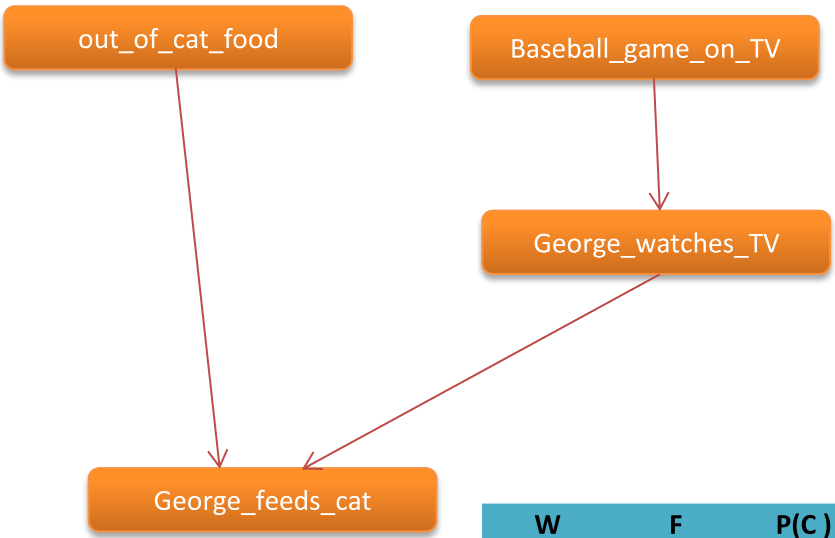
Attached is the excel, **assignment4_sxb9644_task6.xlsx** containing the detailed explanation on the calculations that resulted in the probability tables that are given in the solution below.

Also attached is the **ProbabilityTable.java** which contains the logic for retrieving these probability values from the given training data. While executing **ProbabilityTable.java,** the path of the sample training data has to be given as the input argument. The java code is capable of processing any input file of the provided format for this Bayesian network.

Events	Symbol
Baseball_game_on_TV	G
George_watches_TV	W
out_of_cat_food	F
George_feeds_cat	C

P(F)
0.1698

P(G)
0.3041



G	P(W)
T	0.9279
F	0.1181

W	F	P(C)
T	T	0.0416
T	F	0.7064
F	T	0.3157
F	F	0.9587

Task 7

Given the network obtained in the previous two tasks, calculate P (Baseball Game on TV | not(George Feeds Cat)) using Inference by Enumeration

Using the symbols and the probability tables from task 6,
To find: P(G| ˥C)
Variables that are not present : W and F

Through conditional probability, P(G| ˥C) = P(G, ˥C)/ P(˥C)

Calculating the numerator through inference by enumeration,
P(G, ˥C) = P(G, ˥C, W,F) + P(G, ˥C, W, ˥F) + P(G, ˥C, ˥W, F) + P(P(G, ˥C, ˥W, ˥f)

=[P(G) * P(W|G) * P(F) * P(˥C |W, F)]
+[P(G) * P(W|G) * P(˥F) * P(˥C |W, ˥F)]
+[P(G) * P(˥W|G) * P(F) * P(˥C |˥W, F)]
+[P(G) * P(˥W|G) * P(˥F) * P(˥C |˥W, ˥F)]

=[0.3041 * 0.9279 * 0.1698 * 0.9584]
+[0.3041 * 0.9279 * 0.8302 * 0.2936]
+[0.3041 * 0.0721 * 0.1698 * 0.6842]
+[0.3041 * 0.0721 * 0.8302 * 0.0413]

= 0.0459 + 0.0687 + 0.00257 + 0.0007
P(G, ˥C) =0.11787

Calculating the denominator through inference by enumeration,
P(˥C) = P(G, ˥C, W,F) + P(G, ˥C, W, ˥F) + P(G, ˥C, ˥W, F) + P(˥G, ˥C, W, F) + P(G, ˥C, ˥W, ˥F) + P(˥G, ˥C, ˥W, F) + P(˥G, ˥C, W, ˥F) + P(˥G, ˥C, ˥W, ˥F)

P(˥C) = [P(G) * P(W|G) * P(F) * P(˥C |W, F)]
+[P(G) * P(W|G) * P(˥F) * P(˥C |W, ˥F)]
+[P(G) * P(˥W|G) * P(F) * P(˥C |˥W, F)]
+[P(˥G) * P(W|˥G) * P(F) * P(˥C |W, F)]
+[P(G) * P(˥W|G) * P(˥F) * P(˥C |˥W, ˥ F)]
+[P(˥G) * P(˥W|˥G) * P(F) * P(˥C |˥W, F)]
+[P(˥G) * P(W|˥G) * P(˥F) * P(˥C |W, ˥F)]
+[P(˥G) * P(˥W|˥G) * P(˥F) * P(˥C |˥W, ˥F)]

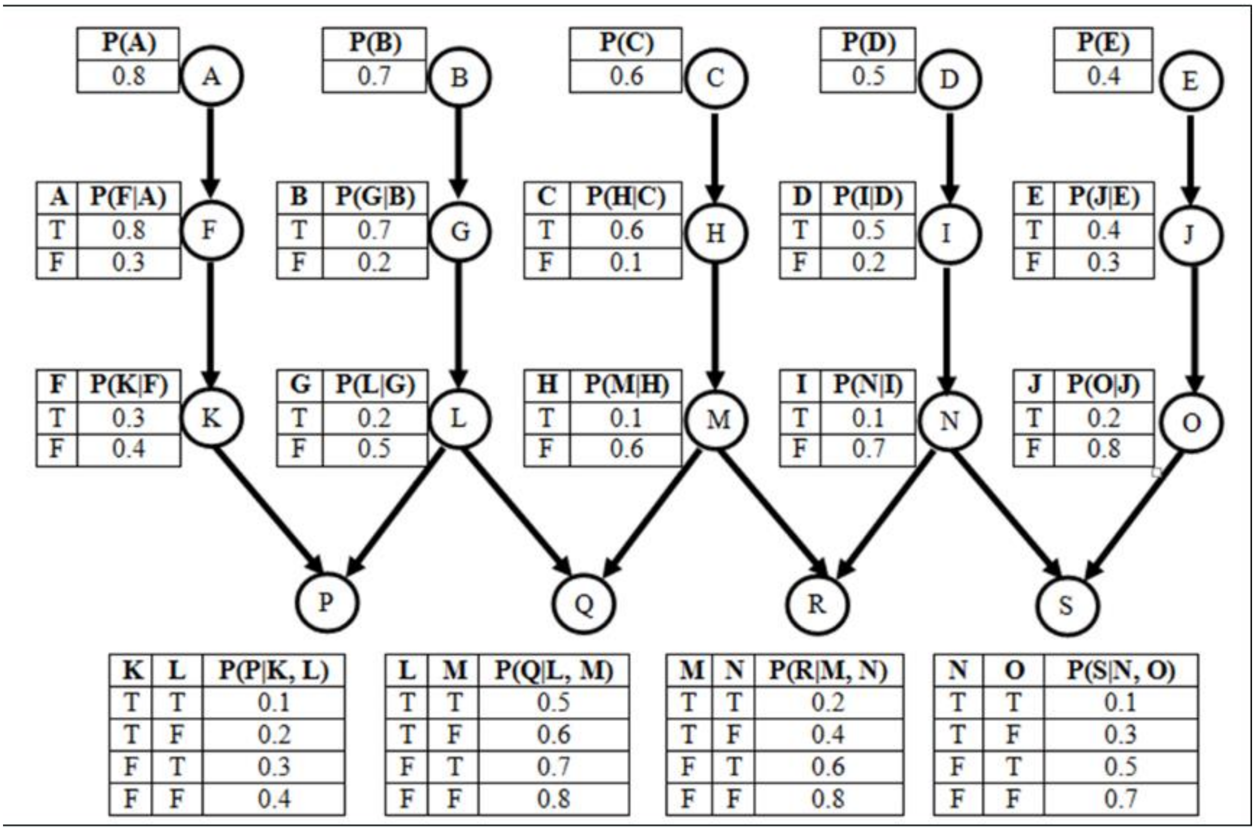
P(˥C) = [0.3041 * 0.9279 * 0.1698 * 0.9584]
+ [0.3041 * 0.9279 * 0.8302 * 0.2936]
+[0.3041 * 0.0721 * 0.1698 * 0.6843]
+[0.6959 * 0.1181 * 0.1698 * 0.9584]
+[0.3041 * 0.0721 * 0.8302 * 0.0413]
+[0.6959 * 0.8819 * 0.1698 * 0.6843]
+[0.6959 * 0.1181 * 0.8302 * 0.2936]
+[0.6959 * 0.8819 * 0.8302 * 0.0413]

P(˥C)= 0.0459 + 0.0687 + 0.0025 + 0.0133 + 0.0007 + 0.0713 + 0.0200 + 0.0210

P(˥C) =0.2434

P(G| ˥C) = 0.11787/0.2434 = 0.4842

Task 8



Part a: On the network shown in Figure 2, what is the Markov blanket of node N?

The set of nodes (parents, children, children's parents) is the Markov Blanket of a node.
Parents of node N : **I**
Children of node N : **R, S**
Parents of Node N's children : **M, O**

Part b: On the network shown in Figure 2, what is P(I, D)? (Note: You can use simplified calculations to calculate this as long as it is justified)

$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(Xi \mid parents(Xi))$
P(I,D) = P(I | Parents(I)) * P(D | Parent (D))
= P(I | D) * P(D)
= 0.5 * 0.5
= 0.25
P(I, D) = 0.25

Part d: On the network shown in Figure 2, what is P(M, not(C) | H)? (Note: You can use simplified calculations to calculate this as long as it is justified)

Based on conditional probability,

$$P(A_1, \ldots, A_k \mid B_1, \ldots, B_m) = \frac{P(A_1, \ldots, A_k, B_1, \ldots, B_m)}{P(B_1, \ldots, B_m)}$$

P(M, not (C)|H) = P (M, not(C), H) / P(H)

Calculating the numerator : P(M, not (C), H)

Implementing the below formula,

P(X_1,….., X_n) = $\prod_{i=1}^{n} \mathbf{P(Xi\ |parents(\ Xi))}$

P(M, not(C), H) = P(M | parent (M)) * P (not (C) | parent(C)) * P(H | parent(H))
= P(M | H) * P(not(C)) * P(H| not(C))
= 0.1 * (1-0.6) * 0.1
= 0.1 * 0.4 * 0.1
= 0.004

Calculating the denominator : P(H)

Implementing Markov blanket for node H,
Parent of H: C
Child of H : M
Other parents of M : None

Implementing inference by enumeration,

P(H) = P(C, H, M) + P(not(C),H, M) + P(C,H,not(M))+P(not(C),H,not(M))

P(H) = [P(C)*P(H |C) * P(M|H)] +
[P(˥C)*P(H | ˥C) * P(M|H)] +
[P(C)*P(H |C) * P(˥M|H)] +
[P(˥C)*P(H | ˥C) * P(˥M|H)] +

=[0.6 * 0.6 * 0.1] +
[0.4 * 0.1 * 0.1] +
[0.6 * 0.6 * 0.9] +
[0.4 * 0.1 * 0.9] +

= 0.036 + 0.004 + 0.324 + 0.036

= 0.4

Now, P(M, not (C)|H) = P (M, not(C), H) / P(H)
= 0.004 / 0.4
= 0.01

P(M, not (C)|H) = 0.01