CSE 471/598: Intro to AI (Fall 2016)

Final exam
(Instructor: Subbarao Kambhampati)

DUE: Thursday 12/8 12noon.
(in hardcopy)

Name:______________________________________

Answer all of the following questions. You are allowed to consult the text book, your notes as well as class webpage and the class videos.

*But you are not allowed to consult with *anyone* other than me (the instructor), nor are you allowed to troll the cyberspace in search of answers.* Violations will be strictly prosecuted.

*****Please read and sign the following declaration**

"I certify, under penalty of academic dishonesty, that I have read and followed all the instructions regarding the examination."

Signature:____________________________________

****PLEASE WRITE LEGIBLY AND STAPLE ANSWERS IN THE CORRECT ORDER. (You don't have to type the answers). **You have to write your answers in the space provided.**

I Painting lessons [Planning] [15]	
II Python, models & Rao's happiness [Logic][12]	
III Arizona Flu Epidemic [Bayes Nets][18]	
IV Newman's Parties [Learning] [15]	
V Three-faced Dice [Likelihood setup] [5]	

Qn I

Consider the following "painting problem." We have a can of red paint (c1), a can of blue paint (c2), two paint brushes (r1, r2), and four unpainted blocks

(b1, b2, b3, b4). We want to make b1 and b2 red and make b3 and b4 blue. Here is a classical formulation of the problem:

s_0 = {can(c1), can(c2), color(c1,red), color(c2,blue), brush(r1), brush(r2), dry(r1), dry(r2), block(b1), block(b2), dry(b1), dry(b2), block(b3), block(b4), dry(b3), dry(b4)}

g = {color(b1,red), color(b2,red), color(b3,blue), color(b4,blue)}

dip-brush(r, c, k)

precond: brush(r), can(c), color(c, k)

effects: $\neg$dry(r), canpaint(r, k)

paint(b, r, k)

precond: block(b), brush(r), canpaint(r, k)

effects: $\neg$dry(b), color(b, k), $\neg$canpaint(r, k)

Part 1: [4] Answer the following questions about the model:

A. [1] In the **paint** operator, what is the purpose of effect **~canpaint(r,k)**?

B. [2] Is the model ensuring that we don't mix paints? (I.e., we don't dip a brush of one color in the can of a different color?). If not, can you think of way to modify the model (add new precondtions/effects as needed)

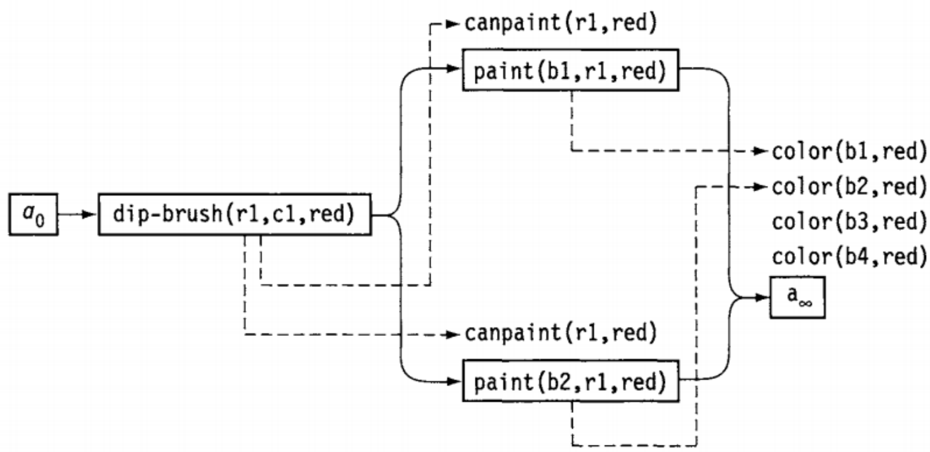
C. [1] Write down a plan (a sequence of actions of the form Dip and Paint—with appropriate parameters) that you think will solve this problem (**according to the model I gave**). Note that this is just to see if you understand the domain enough to write a plan yourself.

[For the following use the model I gave—ignoring any changes you proposed in Part 1.B]

Part 2. [3] Show one level of the progression search (i.e., starting from S0, show the first sets of states that progression would generate)

Part 3.[3] Show one level of the regression search (i.e. starting from g, show the first set of states that regression search would generate).

Now, consider the following partially ordered plan for the problem. The solid arrows are the precedence (ordering) relations. The dashed arrows show the causal links. For example the link from dip-brush to canpaint(b1,r1,red) shows how that precondition of paint(b1,r1,red) is supported).



Part 4: [2] For each causal link in the plan above, identify if it has a "threat" (and what the threat is), and explain the options for resolving the threat.

Part 5: [3] Do you think the plan here can be extended to solve the problem (by adding steps)? If so, sketch how (you don't need to show the search tree; just show the final plan).

Qn II Part A (Inference and Theorem Proving)
[3] Here is a piece of "Python logic"

```
Things made of wood float on water
Mary floats on water
---------
We can infer that Mary is made of wood
```

Using truth table method, show that this inference is **"unsound"** (you can use just two propositional symbols "**MW**" (to stand for made of wood) and "**FW**" (to stand for floats on water)

Qn.II part B [3] (Models)

[Part A]

Consider a propositional language containing four symbols A, B, C and D that can be used to describe the world that the agent is in. The agent starts with no knowledge—and can thus be in any of the 2^4 worlds.

Write down how many worlds are consistent with its knowledge as the following information is added *successively.*

KB ← KB + (A v B)

Worlds =

KB ← KB + (A -> C) + (B -> C)

Worlds =

KB ← KB + ~C

Worlds =

Qn II Part C:[6 pt]
Consider the following

> *If Rao is happy, he sets hard but interesting exams.*
> *If Rao is not happy, he sets long and boring exams.*
> *Prove: This exam is either hard but interesting or long and boring*

(**1**) Encode it into propositional logic (you need just 3 propositional symbols— **RH** for rao's happiness, **HI** for hard but interesting and **LB** for long and boring). (**2**) Convert the sentences (including the appropriately converted goal sentence) into clausal form. (**3**) Show a resolution refutation proof tree that proves the theorem. (**4**)Could this theorem have been proved using just modus ponens inference rule? [Make sure you clearly point out each part of your answer]

s

Qn III **[Arizona Flu Epidemic][]** Suppose we are interested in helping a hospital in *Phoenix* area in using bayes networks to help diagnose flu. The following is the knowledge about the domain that I just gleaned (a fancy word for "made-up").

Presence of rhino virus in the body typically (with probability 0.6) leads to what is known as dispotensia (a weakening of T-cell resistance). Dispotensia might often lead to a pathological condition called Flu (with probability 0.8). Flu itself can lead to a variety of symptoms including: Runny nose (probability 0.8), body aches(probability 0.6), fever (probability 0.5). In phoenix area, during winter, rhino viruses are typically present in humans (with probability 0.7). Finally, dispotensia may be there even when there is no rhino virus (probability 0.3). Flu can be caused even in the absence of dispotensia(probability 0.1). Runny nose may be there in the absense of flu (prob 0.1), body aches may be there in the absence of flu (0.3), and fever could be present in the absense of flu (0.3).

Part A.[4+1+1] (1) Draw a bayes network for representing this knowledge. Show the conditional probability tables at each node. (2) How many probabilities did we avoid gathering because we used the bayes network topology. (3) Is inference on this kind of network polynomial or exponential in the size of the network?

Part B.[2] Dispotensia is not always caused by rhino virus. Sometimes rhino virus may not cause dispotensia at all, if certain conditions don't pre-exist in the patient. But the theory regarding this is not yet complete. Some times, dispotensia may be caused by things other than rhino virus. What, if any, changes do we need to make to the bayes net above if we want to incorporate this knowledge?

<u>Now, suppose a patient walks into the hospital. She hasn't yet been checked for anything</u>.

Part C.[2] What is the probability that this patient is one sick puppy (ie., has rhino virus, dyspotensia, flu, runny nose, body ache *and* fever).

Part D.[5] Now you actually examine the patient and find out that she has a **runny nose,** but **no body aches** and **no fever.** What is the probability that she has **Flu**? (You can use the enumeration method (Russell & Norvig 14.4.1)for computing the probability).

Part F. [3] Suppose we decide to do inference by sampling techniques—specifically likelihood weighting. Suppose we are trying to compute probability that Rhino Virus is true given that the patient has no fever and no body aches. Suppose we generate a sample from the network using likelihood weighting. Assume we sample the network in the order of Rhino Virus, Dyspotensia, Flu, Runny Nose, Body Aches and Fever. Suppose our samples return *True, False, False, True, True, False* (take only the samples you need). What is the complete sample and what is its weight?

Qn IV [15] [Newman's Parties] Continuing your success at learning the pattern behind Costanza's party's from the homework, you have now decided to apply the Machine Learning Technology to Newman's parties.

Now, Newman doesn't call Seinfeld anyway (remember how he wanted to categorically exclude Seinfeld from his Millenium party?). So, for him, it is just a question of calling some subset of George, Elaine and Kramer.

Here are some past parties—with information about who showed up and whether the party was a "success" according to Newman. [Assume that 1 denotes that the person is present and 0 denotes that the person didn't show up. Similarly, in the last column, 1 denotes a successful party while zero denotes an unsuccessful one.).

	Kramer	George	Elaine	Success
P1	1	0	0	0
P2	0	1	0	0
P3	1	1	0	1
P4	0	0	0	1
P5	1	0	1	1
P6	0	1	1	1
P7	1	1	1	1
P8	0	0	1	1

[1] Suppose we assume that the concept is expressible as a Boolean function, what is the size of Newman's hypothesis space?

[5] Apply the decision tree learning algorithm to this problem. Show the information gain calculations at each stage and the full decision tree that is learned.

Now consider learning a Naïve Bayes Classifier for this data using maximum likelihood estimation. There is one difference however—**we don't know whether the parties P7 and P8 were successful (see "?" marks below).**

	Kramer	George	Elaine	Success
P1	1	0	0	0
P2	0	1	0	0
P3	1	1	0	1
P4	0	0	0	1
P5	1	0	1	1
P6	0	1	1	1
P7	1	1	1	**?**
P8	0	0	1	**?**

(1) [5] Show the structure of the naïve bayes network, and the parameters you estimate for it using the 6 examples. (**You don't need to use laplacian smoothing**).

(2) [2] Using the NBC you learn in the previous part, predict the **class distribution** of P7 and P8 (you need to find the probability that the party is a success and is not a success)

(3) [2] Now that we have class data for all 8 examples (including the two fractionally labeled ones), re-compute the parameters of the network.

Qn V.[5] This makes you go through the steps of setting up the likelihood function, and finding parameters that maximize it.

Suppose you are playing with a 3-faced dice. When thrown it can land in one of three positions numbered 1, 2 or 3 (see the picture)

You don't know up front what the probabilities of each position *p1, p2, and p3* are. Suppose you throw the dice many times and get <u>n</u> samples of training data 11223133122… etc. (the sequence is just illustrative).

When we analyze the outcomes, we notice that the dice comes out 1 *n1* times,2 *n2* times and 3 n3 times.

We want to "learn" the values of the parameters *p1, p2 and p3* that maximize the likelihood of the observed data.

Set up the expression for likelihood in terms of the parameters p1, p2 and p3 (note: write p3 as (1 – p1 – p2)). Find expressions for the parameters using MLE by maximizing the likelihood (hint: you would want to take the log of the likelihood expression, and finding its partial derivatives and equating them to zero, and solving for the parameters.