Computer Science Department
Stanford University
Comprehensive Examination in Artificial Intelligence
Autumn 2008

My Comp Exam ID Number (formerly known as a MAGIC NUMBER)is ________________.

PLEASE READ THIS FIRST

a. You should write your answers for this part of the Comprehensive Examination **onthis exam**. Be sure to write yourComp Exam ID Numberon the cover of the exam.

b.You should be sure you have all the pages of this exam. There are 10 pages in this exam, including thiscover sheet.

c. This exam isOPEN BOOK. You may use notes, articles, or books.

d. You are allowed to use a basic calculator (a four function one is adequate, but it'sokay if it does logs, etc.).

e. But you are not allowed any help from people, or use of computers, PDAs, phones, Gameboy AdvanceSPs, or other deviceswith communication capabilities.

f. Show your work, sincePARTIAL CREDITwill be given for incomplete answers. Forexample, you can get credit for making a reasonable start on aproblem even if theidea or arithmetic does not work out. If you're really lucky, you might also get credit for realizing that certainapproaches are incorrect.

g. Points in this exam add up to 60. Points are allocated according to the numberof minutes we believe a student familiar with the material should take to answerthe questions. If you are somewhat less familiar with the material, a question mayeasily take you longer than the number of points it's worth. Therefore be careful:**IF YOU ARE TAKING TOO LONG ON A QUESTION, WRITE DOWN WHATEVER YOU HAVE AND MOVE ON.**

Question	Score	Possible
1 CSP		8
2 A*		8
3 Prob		12
4 ML		12
5 Vision		8
6 Robotics		8
7 NLP		4
Total		60

Q1. CSP algorithms (8 points). A Latin Square is an N x N array filled with colors, in which **no color appears more than once in any row or column** (there is no restriction along diagonals). Solving a particular Latin Square problem involves completing (or showing the impossibility of completing) a partially initialized square. We want to formulate finding a solution to the 4 x 4 Latin Square below as a CSP. The four colors that squares can be colored are red (r), green (g), blue (b), and yellow (y).

i. (1 point) What are the variables? (A statement is sufficient, you don't have to list them all)

A (categorical) variable s_{i,j} for each of the 16 squares

ii. (1 point) What are their possible values?

The colors: {r,g,b,y}

iii. (2 point) What are the constraints? (Indicate the nature of the constraints; you don't have to write them all out in any formal notation.)

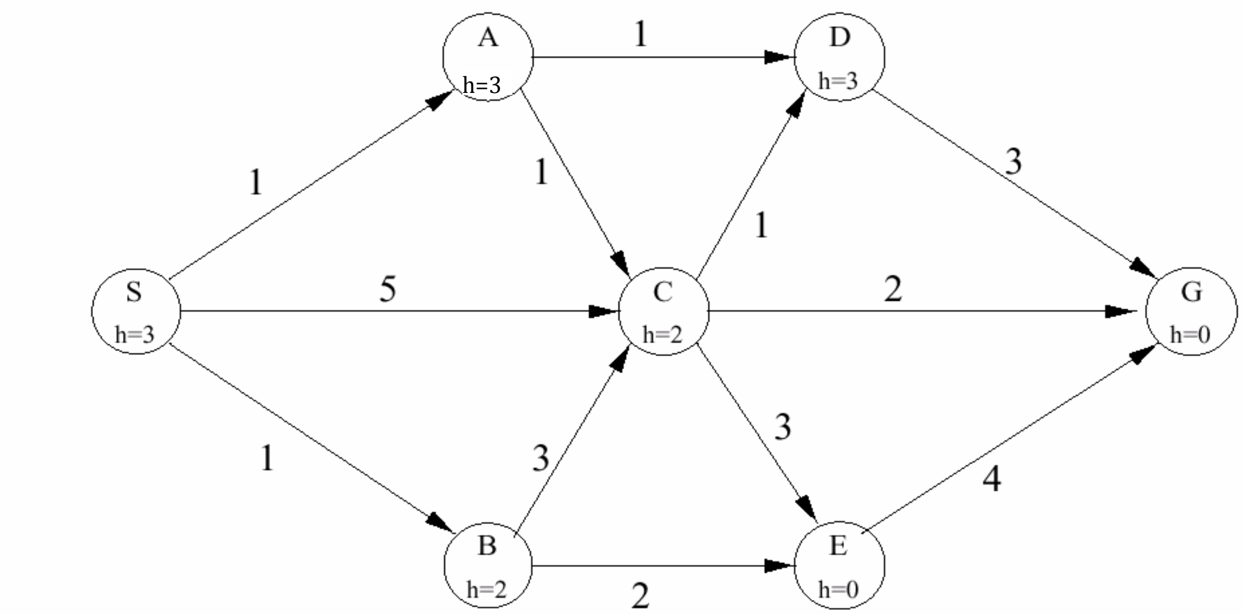
For i ne i' s_{i,j} ne s_{i',j}
For j ne j' s_{i,j} ne s_{i,j'}

iv. (4 points) Apply arc-consistency from the initial position to restrict the values of the variables as much as possible. Show your work on the Latin square below. Cross out values that are excluded by arc-consistency. Finish by indicating whether the problem has **no solution** (is inconsistent), has a solution that is **determined** by arc-consistency, or whether **search** is required to find the solution.

r	g	b	y
g	y	r g b y	r g b y
b	r g b y	r g b y	r g b y
r g b y	r g b y	r g b y	r g b y

Status of problem after applying arc consistency: Solution determined!

Q2. A* Search (8 points) Consider the search problem below.



Suppose we ask A* search to compute the least-cost path from S to G in this **graph**. The arrows indicate possible successor states, and the numbers on them indicate the path costs. The 'h=' numbers show the (admissible) heuristic values estimated for each state.

The A* search keeps a priority queue as it explores the graph. Fill in the following table with the contents of the priority queue after each iteration of the A* search algorithm loop (the best – lowest scoring – entry in the priority queue is written to the left). After the iteration that A* decides to finish and return, write "Finished". (That is, you are not expected to fill in all lines of the table. At some point the A* algorithm will terminate.) We've started the table for you.

After iteration	The priority queue contains (in order, low to high)
Initialization	(S, 3)
1	(B, 3), (A, 4), (C, 7)
2	(E, 3), (A, 4), (C, 6)
3	(A, 4), (C, 6), (G, 7)
4	(C, 4), (D, 5), (G, 7)
5	(G, 4), (D,5) [, (E, 5) if you don't record visited states)
6	Finish on popping (G, 4); path is (S, A, C, G), cost 4
7	
8	
9	
10	
11	
12	

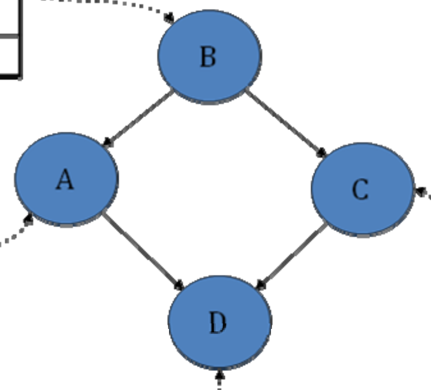
Q3. Probabilistic models (12 points). Consider the following Bayes net:

B is: there is a burglary at my house

B	P(b)
y	.01
n	.99

A is: my alarm goes off

A	P(a B y)	P(a B n)
y	.8	.01
n	.2	.99



C is a neighbor calls the police

C	P(c B y)	P(c B n)
y	.4	.1
n	.6	.9

D is: the police show up at my house

D	P(d A y,C y)	P(d A y,C n)	P(d A n,C y)	P(d A n,C n)
y	.99	.7	.9	.01
n	.01	.3	.1	.99

a. (1point) What is the prior probability of a burglary at my house?

0.01

b. (1point) Qualitatively, given this Bayes net, if the police show up at my house, willthe posterior probability of a burglary at my house be **greater** than or **less** than the priorprobability?

Greater

c. (9points)Work out theP(B=y|D=y)(asarealnumber)for this model. You might want to use the variable elimination algorithm (but this is not required). **Show your work.**

P(B=y | D=y)= P(B=y, D=y)/P(D=y)
= Sum_{A,C} P(A,C,B=y, D=y)/ Sum_{A,C,B} P(A,C,B,D=y)
= P(B=y) SUM_{A} P(A | B=y) SUM_{C} P(C | B=y) P(D=y | A, C) / …

Numerator =P(B=y)*[P(A=y | B)*(P(C=y | B)P(D=y | A,C)+P(C=n | B)P(D=y | A,C)) + …
= 0.01*[0.8*(0.4*0.99+0.6*0.7)+0.2*(0.4*0.9+0.6*0.01)]
= 0.01*[0.8*(0.396+0.42)+0.2*(0.36+0.006)
= 0.01*[0.8*0.816+0.2*0.366]
= 0.01*[0.6528+0.0732]
= 0.01*0.726
= 0.00726

[More space to show your work for 7c…]

P(B=n | D=y) = Sum_{A,C} P(A,C,B=n,D=y)
= 0.99*[0.01*(0.1*.99+.9*.7)+0.99*(.1*.9+.9*0.01)]
= 0.99*[0.01*(.099+0.63)+0.99*(0.09+0.009)]
= 0.99*[0.01*0.729+0.99*0.099]
= 0.99*[0.00729+0.09801]
= 0.99*[0.1053]
= 0.104247

P(B=y | D=y) = Numerator / (Numerator + P(B=n | D=y))
= 0.00726 / (0.00726 + 0.104247)
= 0.0651

d. (1 point) What is the likelihood ratio of a burglary given that the police have shown up versus the prior probability of a burglary?

LR = 0.0651 / 0.01 = 6.5 times as likely

Q4. Machine Learning (12 points).

a. (3 points) There are 4 choices for the value of the **Outlook** attribute. It can be: sunny, overcast, raining, or snowing.

   i. (2 points) We sampled the weather on 32 days in Pittsburgh, and it was sunny on 4, cloudy on 16, raining on 8 and snowy on 4. What is the entropy (called **information** in R&N) of the weather in Pittsburgh? (Calculate with $\log_2$, so as to produce an answer in base 2 bits.)

H(Pitt) = - [1/8 log2(1/8) + ½ log2(1/2) + ¼ log2(1/4) + 1/8 log2(1/8)]
= - [-3/8 + -1/2 + -2/4 + -3/8]
= 1.75 bits

   ii. (1 point) We also sampled the weather on 32 days in Stanford, and it was sunny on 16 and cloudy on 16. What is the entropy of the weather at Stanford?

H(Stan) = - [1/2 log2(1/2) + ½ log2(1/2)] = 1 bit

b. (9 points) The aim of the Play Tennis dataset from Tom Mitchell is to predict whether or not he would play tennis based on various attributes of the weather (the Day isn't an attribute, it's just a unique identifier):

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Consider building a (multivariate) Naïve Bayes classifier trained on the above data. The model parameters should be estimated without smoothing. Then use it to predict whether hewould play tennis for this new piece of test data:

Outlook = Sun, Temp = Cool, Humid = High, Wind = Strong

(i) (2 points) State the Naïve Bayes decision rule for this classifier, by giving an equation in terms of the probabilities of the attributes.

Choose argmax_{PT in {Yes, No}}
P(PT)P(O=Sun|PT)P(Temp=Cool|PT)P(Humid=High|PT)P(Wind=Strong|PT)

(ii) (6 points) Show your work in evaluating the answer, in particular working out the relative likelihood of playing tennis or not under the Naïve Bayes assumption.

P(PT=yes) = 9/14
P(Sun|yes) = 2/9, P(Cool|yes) = 3/9, P(High|yes) = 3/9, P(Strong|yes) = 3/9
Numerator of Bayes rule = 9/14 * 2/9 * (1/3)^3 = 1/(7 * 27) = 0.00529

P(PT=no) = 5/14
P(Sun|no) =3/5, P(Cool|no) = 1/5, P(High|no) = 4/5, P(Strong|no) = 3/5
Numerator of Bayes rule = 5/14 * (3/5)^2 * 4/5 * 1/5 = (3 * 6)/(7 * 125) = 0.0206

The denominators are the same and cancel, so the predicted probability of playing tennis is 0.00529/(0.00529 + 0.0206) = 0.204

(iii) (1 point) State what the classifier would predict.

PlayTennis = No

Q5. Vision (8 points)

a. (1 point) Suppose five points in a scene are collinear. Will they also be collinear in a (distortion-free) pinhole camera image of the scene?

Yes.

b. (1 point) Suppose a pinhole camera image contains five collinear points? Are the five points in the physical world corresponding to the points in the image necessarily collinear?

Generally no, because they might have different depths.

c. (6 points) Suppose in a Kalman filter, you have a one-dimensional Gaussian prior defined through mean μ and variance $\sigma^2 (X \sim N(\mu, \sigma^2))$. Then you take a measurement of the system state, which (for simplicity) is in the same coordinates as μ. Let's call the measurement z, and assume the measurement noise covariance is also σ^2 (the same σ^2 as above). That is $Z|X \sim N(X, \sigma^2)$.

   i. (3 points) What will the posterior mean be?

0.5 * (mu + z)

   ii. (3 points) What will the posterior covariance be?

1/2 sigma^2

Q6. Robotics (8 points)

a. (3 points) R&N talk about particle filters for mobile robot localization. Suppose you operate a robot in a building with a known (and correct) map, and you use range finders. Explain conditions under which the particle filter will break. That is, suppose you implement the basic particle filter as described in R&N. Under what conditions will this simply not work, even if its implementation is mathematically correct? We're interested in conceptual/mathematical answers, not answers like "if the linux kernel seg faults".

- perfect robot motion: then the initial particle set will have to have the exact robot pose among them, since we'll never add noise to the predicted motion. This happens with probability zero.
- perfect sensors: same problem: then each particle will get the weight zero, with probability one

Less satisfactory partial credit answer:
- wrong model of robot motion or wrong model of sensor noise

NOT a correct answer
- too few particles
- a symmetric environment (the particle filter should work just fine in estimating the posterior distribution, and nowhere in the question are we asking for an estimate of the robot's position - the particle filter estimates a posterior for this position, not the position).

b. (2 points) Why do we care that the configuration space of a robot is connected?

So that the robot can reach every possible configuration. A disconnected configuration space "traps" the robot in a subspace of all possible configurations.

c. (3 points) In a PID controller, what is the effect of the different terms - P,I, D? Can you give an example of each (just a phrase)?

PProportional control: reduces control error by providing negative force in proportion to control error

IIntegral control: provides adaptation in the face of long-lasting systematic error

DDerivative control: dampens system being controlled, prevents overshoot

Q7. NLP (4 points)

a. (1 point) Give an example of an adjective phrase that is longer than 1 word:

Lots of answers, e.g.,*extremely difficult*

b. (1 point) What is the time complexity of bottom-up parsing (the basic kind presented in R&N), in terms of the length of the sentence?

Exponential – O(2^n)

c. (1 point) What is the time complexity of chart parsing (in the length of the sentence)?

Cubic – O(n^3)

d. (1 point) What does the **fertility model** component of the IBM machine translation (MT) models model? (Say specifically how it relates to source or target language words, as appropriate.)

P(*n*|*w*)
The probability of each *target* language word *w* being translated by *n* words in the source.