GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

COMP 424 FINAL 2004

Some solutions here:
http://www.cs.mcgill.ca/~dprecup/courses/AI/sample_questions_answers.pdf

1.a)

If h is between 0 and 1 sqrt(h) will be more pessimistic than h
h^2 may be admissible if it is still <= h*. (not admissible because it overestimates when h >= 1) It is likely not though. sqrt(h) will be admissible.
If h^2 was still admissible it would be better than h, sqrt(h) would be worse.
(this is because we say h' dominates h if h'>=h always, which h^2 is.

b) 132 | 4421---------> 1321421
275| 1421 --------> 2754421

c) i. False
ii. True
iii. False
iv. True

d)

A = 8 allows B to be pruned as Min player will choose the left branch
A = 5 will not be pruned as we do not yet know if B will be greater than 7.

e)

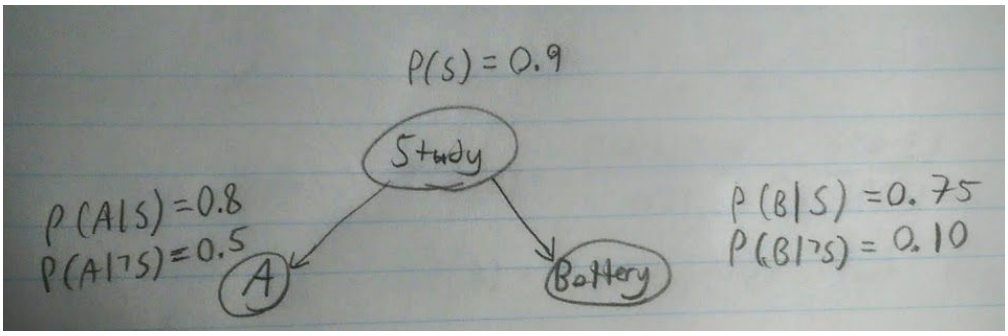
C=D=1 would result in Min player choosing the C D subtree at its turn, so we will prune E F because the Max player will choose the left tree from the root in this case.

2. a)i , ii, iii
b) iii. Linear perceptron can't separate XOR (make the graph), neural nets require at least one hidden node to represent XOR.
c) TopRegress because it has the best accuracy on the test data? Yea, ideally you want a robust model. That is shown with high test accuracy.

d) anyone have any insight on this? Write out derivative of mean square error with respect to each wi. Wk+1 = wk + alpha*(gradient)

3.
a)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

b) 

P(A|B) = P(A,B)/P(B)
= P(A)* P(B) / P(B)
= (0.8*0.9 + 0.5*0.1) = 0.77

Note that here A and B aren't independent since we don't know S. Therefore P(A,B) != P(A) * P(B).

This should be:
P(A|B) = P(A, B)/P(B)
= (P(A|S)*P(B|S)*P(S) + P(A|~S)*P(B|~S)*P(~S)) / (P(B|S)*P(S) + P(B|~S)*P(~S))
= 0.796 (+2)

c)

EU(S) = U(A) * P(A|S) - U(B)*P(B|S) = 10*0.8 - 5*0.75 = 4.25
EU(~S) = U(A) * P(A|~S) - U(B)*P(B|~S) = 10 * 0.5 - 5*0.1 = 4.5

So we are better off to not study.
If only that was true

4.

a)

Let:
F(x) = x is a citizen of Fredonia
S1(x, L) = x speaks language L
S2(x, D) = x speaks dialect D
Has(L, D) = language L has dialect D
D1,D2 = the 2 dialects of Fredonian

i.

$\exists l\ \forall x\quad F(x) => S1(x, l)$

ii. (pretty sure it's just at least 2 dialects in which case it's only first line)
$\exists\, d1, d2\ (d1\ != \ d2\ AND\ Has(Fredonian,\ d1)\ AND\ Has(Fredonian,\ d2))$

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

$\forall d \ (d != d1 \ AND \ d != d2 \ => \ \neg Has(Fredonian,\ d)\ )$

Iii.

$\forall x \quad F(x) \ => \ (S(x, D1) \ AND \ \neg S(x, D2)) \ OR \ (S(x, D2) \ AND \ \neg S(x, D1))$

b)

i. $\forall x \ \ [NOT \ F(x)] \ OR \ S1(x, L(x))$

ii. $(D1 \ != \ D2) \ AND \ Has(L(x),\ D1) \ AND \ Has(L(x),\ D2))$

iii.

$\forall x \ \ [NOT \ F(x)] \ OR \ \ (S2(x, D1) \ AND \ \neg S2(x, D2)) \ OR \ (S2(x, D2) \ AND \ \neg S2(x, D1))$

Pretty sure the above is wrong (there are no universal quantifiers in CNF)

True, this was copy pasted from the teachers answers (see link above)

Yeah this is skolemized but not in proper CNF

5. a) 4/9

b) what is the test set? Every data point?

6.

a)

P(L) = 2/5 = 0.4

P(L | D1) = 2/3 = 0.667

P(L|~D1) = 0/2 = 0

P(L| D2) = 2/2 = 1

P(L|~D2) = 0/3 = 0

P(L|D3) = 1/3 = 0.333

P(L|~D3) = 1/2 = 0.5

Aren't the values supposed to be of the form P(L | D1, D2, D3) ?

Shouldn't we compute P(D1), P(D2), P(D3) instead of P(L) ?

- **definition of naive bayes : P(L | D1,D2,D3) = alpha * P(L) * product of P(Di | L)**
  - **I.e. all "symptoms" are children of root (here L), independent of each other given the root**
  -

Agreed

P(L) = ⅖

P(D1 = 1|L=1) = 2/2 =1

P(D1 = 1| L=0) = ⅓

P(D2=1 | L=1) = 1

P(D2=1|L=0) = 0

P(D3=1 | L=1) = ½

P(D3=1|L=0) = ⅔

P(D1=1) = ⅗

P(D2=1)=⅖

P(D3=1)=⅗

b)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Only on Day 1:
P(L|D1,~D2,~D3) = P(L, D1, ~D2, ~D3) / P(D1,~D2,~D3)
= 0/((3/5)*(3/5)*(2/5)) = 0

On Days 1 and 3:
P(L|D1,~D2,D3) = P(L,D1,~D2,D3)/P(D1,~D2,D3)
= 0/((3/5)*(3/5)*(3/5)) = 0

With laplace smoothing this would be different perhaps? Both would have 0.5

Shouldn't this be:
P(L=1 | D1 = 1) = P(L=1, D1 = 1) / P(D1 = 1) = 2/3
P(L = 1| D1 = 1, D3 = 1) = (P(L = 1, D1 = 1, D3 = 1)/P(D1 =1, D3 = 1) = 3/5
Reads Shakespeare only on day 1 => doesn't read it on day 2 or day 3?

Naive means there's no conditional probability between D1, D2, D3. So for D1&D3 this makes sense. But for D1 only this shouldn't be correct. But that's what she did in the solutions… Who's right ?

Only on Day 1:
P(L|D1,~D2,~D3) = P(L, D1, ~D2, ~D3) / P(D1,~D2,~D3)
=
P(L)P(D1|L)(1-P(D2|L))(1-P(D3|L))/[P(L)P(D1|L)(1-P(D2|L))(1-P(D3|L) + P(~L)P(D1|~L)(1-P(D2|~L))(1-P(D3|~L)]

Agree with this formula, but 1-P(d2|L) is 0, which makes whole thing 0

c)

Since we only have 1 entry of (D2, D3), we would up the count of (D1,D2,D3) to 2, but this still gives us P(L|D1,D2,D3) = 1 as before, so it makes no difference.

That makes no sense to me. Shouldn't we use Soft EM or Hard EM to add incomplete data to the set ? Agreed x 4

7.

a)

WakeUp(S):
Preconditions: Asleep(S), In(Dorm)
Postconditions:
Add: ~Asleep(S), ~Charged(S)
Delete: Asleep(S)

Recharge(S):
Preconditions: ~Asleep(S), ~Charged(S), In(Dorm)
Postconditions:
Add: Charged(S)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Delete: ~Charged(S)

Go(S, Dorm, Exam):
Preconditions: ~Asleep(S), In(Dorm), ~In(Exam), Charged(S)
Postconditions:
Add: In(Exam), ~In(Dorm)
Delete: In(Dorm), ~In(Exam)

More generally:
G(S,X,Y):
Precondition: ~Asleep(S), In(X,S), ~In(Y,S), Charged(S)
Postconditions: ~Alseep(S), ~In(X,S), In(Y,S), ~FullyCharged(S)

b)

Obviously we must go in the order above. Cannot do anything while Asleep except WakeUp(S). After this we can only either go back to sleep or charge. After this we can go back to sleep or go to the exam.

Kinda confused what a planning graph is. Does anyone see this is the notes? (no - doubt it's in the exam) GraphPlan. The prof said we didn't cover this semester.

Question 8 MDP:

a) The optimal policy is to float on every state ~~except n-1 where you should reset. (answer based on intuition)~~ Wouldn't you want to collect the 10 reward at the end instead of resetting? If you reset, you have an infinite loop -> inifnite reward (with respect to discount factor).Yes but the thing is, it's 1/.5 = 2. So compare that to 10, you would prefer it any time. What's 1/.5 ? the discount factor is 0.5. It's because you have a discount rate that having 10 dollars now is better than a stream of infinite 1$. Yeah I see what you mean, we would have to run the algorithm on multiple iterations in both cases to actually have to optimal solution. Ohh got it, the **infinite sum of ½^k converges to 2.**
b) V*(n) = 10
V*(n) is the optimal value of starting at state n. So the optimal value is 10 because all you can do is do that one transition that gives you 10 and terminates.Why does the solution (which is on top of this page) have a different answer? (20) It seems like the solution implies that it does not terminate after it does the transition from state n to state n though (+1).

c) V*(k) = 1 + y + y^2 + … + y^{n-k}(10) = 1 + (1/2)*1 + (1/2)^2 * 1 …. + (1/2)^k * 10
Where y is the discount factor

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

$$V^*(k) = \sum_{x=0}^{n-k-1} \gamma^x + 10 * \gamma^{n-k}$$

~~Since we are less than n here we never get the 10 reward, so it would be:~~
~~V*(k) = 1 + (1/2)*1 + (1/2)^2 * 1 .... + (1/2)^k * 1~~
~~That's just my thinking though.~~
~~Although, if we aren't gonna make it to state n, we have more value by looping back as this will converge to 2 as explained in a, whereas for a finite k<n, we will approach 2 but not quite get there.~~
~~Ex: without looping V(1) = 1, V(2) = 1.5, V(3) = 1.75, ...~~
~~with looping V(k) = 2 forall k<n~~
~~**So I think the V*(k) = 2 for k<n**~~

d) ~~V_0(k) = 0 for every k~~
~~V_1(k) = 1 + yV_0(k+1) = 1 (since V_0(k+1)=0 (for every k until k-2)~~
~~V_2(k) = 1 + yV_1(k+1) = 1 +(1 + yV_0(k+2)) for each k until k-2~~
~~= 1 + ½(1) = 1.5~~
~~V_1(n-1) = 1~~
~~V_2(n-1) = 1 + ½(10) = 15~~
~~V_1(n) = 10 + **yV_0(n+1) -> not defined ? (I think we can assume V(z)=0 if z is not in [1,...,n])**~~
~~V_2(n) = **not defined ?** = 10~~

From Solutions:
V1(n) = 10
V1(n−k) = 1 ∀k > 0.
V2(n) = 10~~+10 ∗ 1/2 = 15~~ or should it be 10 since it terminates after +10? Yes that's correct. The solutions are wrong, she doesn't take into account the fact that it terminates after one cycle.
V2(n−1) = 1+1/2 ∗V1(n) = 6
V2(n−k) = 1+1/2 ∗V1(n−k +1) = 1+1/2 = 3/2 ∀k > 1

e) We'll do both **TD** and **Monte Carlo**

TD:
Let learning rate alpha = 0.1
For 1st trajectory: V(n-1) = V(n-1) + alpha(1 + yV(n) - V(n-1)) = 0.1
V(n-1) = V(n-1) + alpha(1 + yV(n) - V(n-1)) = 0.1 (should minus)

and: V(n) = V(n) + alpha(10 + y[**V(n+1) = 0 ?**] - V(n)) = 1
[V(n+1)=0 or or just don't learn from terminal nodes]
For 2nd trajectory: V(n-2) = V(n-2) + alpha(1 + y[V(n-1) = **0.1**] - V(n-2))
Isn't V(n-1) = 0.1? YES!
= 0 + 0.1(1 + 0.05 - 0) = 0.105

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

V(n-1) = [V(n-1)=.1] + alpha(1 + y[V(n) = 1] - [V(n-1) = 0.1])
= .1 + .1(1 + 1/2 - .1)
= .1 + .1(1.4) = .1 + .14 = 0.24
V(n) = V(n) + alpha(1 + y[**V(n+1) = 0 ?**] - V(n))
= 1 + .1(10 + y0 - 1) = 1.9

Monte Carlo:

For trajectory 1:
V1(n) = 10
V1(n-1) = 1 + y10 = 6

For trajectory 2:
V2(n) = (10 + 10)/2 = 10
V2(n-1) = (6 + 6)/2 = 6
V1(n-2) = 1 + ½ + 2.5 = 4 Why is it 4.5 in the solutions → she must've made a typo Agree with 4 x 2

<u>Question 9</u>

The fuck is dis (maybe use MDPs?)
Second MDP
Something about evaluating utilities of each product and trying to maximize total value in inventory? MDP I think...

I feel like a bayes net with learned parameters would be more appropriate

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

COMP 424 FINAL 2005

Question 1:
(a) The representation ar
e the values of A,B,C,D
Initial state is A=B=C=D=0
(b)
4 branching to the left until A=1, then one branching on the right to get B=1 and boom the solution.

Yes but depth is 4 so you will switch A back and forth until your depth is 4.

The states are the assignments of the variables represented in the tuple {A,B,C,D}
Therefore we expand as follow:
{0,0,0,0}, {1,0,0,0}, **{0,0,0,0}**, {1,0,0,0} (depth 4 unsatisfied, start expanding on bold state) {0,1,0,0}, {0,0,1,0}, {0,0,0,1}, **{1,1,0,0}**

> **doesn't {0,1,0,0} satisfy the solution? Also you stopped at a depth of 3 (root = depth of 0), it should be:** {0,0,0,0}, {1,0,0,0}, {0,0,0,0}, **{1,0,0,0}**, ← thanks! {0,0,0,0} (backtrack to bold), {1,1,0,0}

I did it like so:
{0,0,0,0}
changeA
{1,0,0,0}
changeB
{1,1,0,0}
changeC changeD
{1,1,1,0} {1,1,0,1} <- satisfied
changeD
{1,1,1,1}
No more operators, go back up

(c) satisfied disjunctions -> cost so far. not-satisfied disjunctions -> cost to go And how does this answer the question?

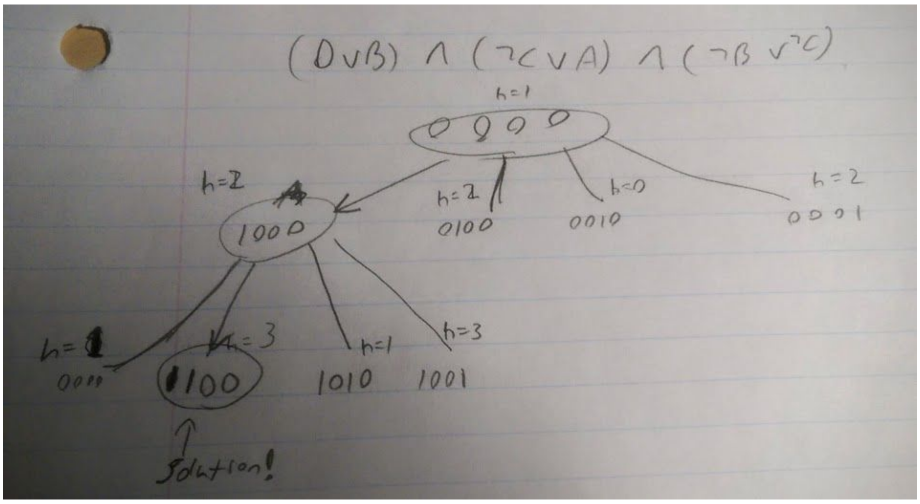
I think its not admissible because a heuristic is supposed to indicate the cost to get to the solution. (so it is not supposed to grow when we get closer to the solution). Also note that h is admissible iff h <= h* where h* is the actual shortest path to finishing. h* is the min number of changes left to make, and clearly h = satisfied disjunctions can be greater than this, ie at the end we have 0 changes left to make but 3 satisfied disjunctions. (+1 this makes sense, h counts the number of

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

disjunctions we have already satisfied. However, when we reach the goal h = 3! Which is not good as a requirement for admissibility is that h(goal) = 0)

We may be looking at this the wrong way though as 1-h is ofc admissible. In d when we do best first I assume "best" would mean highest number of satisfied disjunctions, because picking the option with lowest makes no sense.

(d) This question uses the heuristic in C), so maybe it should be admissible …



(0,1,0,0) is a solution, because second branch has h=3. So don't expand 1000.

<u>Question 2:</u>
a) A=B=5, C=10
b) E=4, F=4

<u>Question 3:</u>
(a)
P(D) = probability of a random guy having the disease = 1/1000
P(T|D) = probability of test being positive when having the disease = 0.99
P(not T|D) = 0.01

P(T|not D) = 0.01
P(not T| not D) = 0.99

Using Bayes' theorem: P(D|T) = (P(T|D)P(D))/P(T) = 0.99 * 0.001 / P(T)
P(T) = P(T|D)P(D) + P(T| not D)*P(not D)
= 0.99*0.001 + 0.01*0.999 = 0.01

P(D|T) = (0.99 * 0.001)/0.01 = 0.099

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

So 10% of chances of having the disease if the test is positive.

More exactly
P(T) = P(T|D)P(D) + P(T| not D)*P(not D) = 99/100*1/1000 + 1/100 * 999/1000 = 99/100,000 + 999/100,000 = **1,098/100,000**
The question is asking to find P(D) tho, its 1/(1+0.00999), you have to find P(D|T+)

P(D=1 | T=1) = P(D=1, T=1) / P(T=1)
P(D=1, T=1) = P(D=1) P(T=1 | D=1) = 1/1000 * 99/100
P(T=1) = P(D=1) P(T=1 | D=1) + P(D=0) P(T=1 | D=0)
= 1/1000 * 99/100 + 999/1000 * 1/100

P(D=1 | T=1) = (1/1000 * 99/100) / (1/1000 * 99/100 + 999/1000 * 1/100)
= (99/100,000) / (99/100,000 + 999/100,000)
= 99/100,000 / 1098/100,000
= 99/1098

(b)
The probability that john has the disease = P(D|T) * P(D|not T)
Isn't P(D | T1, ~T2) what we're looking for? the two tests are independent GIVEN the presence or absence of disease
Where P(D| not T) = (P(not T|D) * P(D))/P(not T)
= (0.01 * 0.001)/0.901
= ~ 0 (it is really low)

^^ Note sure if this is the right answer
P(D | T1, ~T2) = P(T1, ~T2 |D) / P(T1, ~T2)
Since the outcomes of the tests are independent given the disease: P(T1, ~T2) = sum_d P(T1 | d) * P(~T2 | d) *P(d) { conditioning on D}

P(T1, ~T2 | D) = P(T1 | D) * P(~T2 | D)

By working it out: i get P(D | T1, ~T2) =0.001 ← this is a 0.1% chance that John has the disease.

(c)
It is more probable that the first test fucked up
Yes, either test works out wrong, and he is safe and cured :)

Question 4:

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

(a) This is a joke
Set A = take COMP 424
Set B = cry a lot
Set C = work a lot
Set D = fail
Set E = be tired at the exam

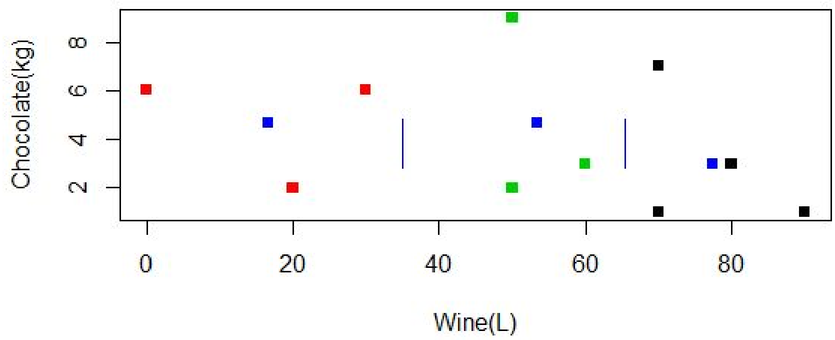
LOL

(b)

(i) If **A is unknown**, therefore B and E are **dependent**
(ii) If **D is unknown and A is known,** then B and E are **independent.**
(iii) If **D is known and A is unknown,** then B and E are **dependent**.

Question 5:
(a) ANN3 because it has the lowest test set validation error? Is that all we use to determine which is best?
(c) Put 1 center on subject (F -> 1,S->9,B->7)
Blue: Centers, blue lines- voronoi tessellation

NOT SUPPOSED TO BE ABLE TO COMPUTE K-MEAN (SEE SLIDES)



Question 6:
WITHOUT CASH
(a) The numbers:
- The R1 takes 15 minutes when no traffic and 60 minutes when traffic, we want to minimize the time taken to go to work, so we want to maximize it's negative values (-15, -60)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Let P(T) = probability of having traffic.
Utility of U(r1) = P(not T)*U(R1|not T) + P(T)U(R1|T)
= 0.8(-15) + 0.2(-60)) = -24
Utility of U(r2) = P(T)*U(R2|T) + P(not T)P(R2| not T)
= -30
A modest suggestion: U(r1|~T)=15, U(r1|T)=0, U(r1)=12

(b) VPI: (0.8*(-15) + 0.2(-30)) - (-24) = 6.
Since it's 6 minutes, we can compute the cash value that.
6/60 * 20 = 2$

WITH CASH

(a) Assuming the person is paid 20$ an hour (Is that in the text ? It's in the question, I assumed that the person is risk neutral therefore the monetary value corresponds to the utility itself. Doesn't it just come down to minimizing the time it takes to go to work).

WITH LOTTERIES:

Setting up lotteries:
L(Route1) = [95, 0.8; 80, 0.2]
L(Route2) = [90, 1]
MEU(R*) = max_r U(L(r)) = max (95*0.8 + 80*0.2, 90) = 92 ~~(R1 better route)

WITH UTILITIES:

You forgot the time (15 mins) taken to travel back to campus. So we have 4.75-TravelTime hours. Crossed out and replaced with correct answers.
U(R1|T) = 20*(4.75-1) = 75, U(R1|~T) = 20*(4.75-0.25) = 90
U(R2) = 20*4.25 = 85
Using utilities: EU(R1) = P(T)*U(R1|T) + P(not T)U(R1| not T)
~~= 0.2 * 80 + 0.8 * 95~~
~~= 92~~
= 0.2*75 + 0.8*90 = 87
EU(R2) = P(T)*U(R2|T) + P(not T)U(R2| not T)
~~= 0.2 * 90 + 0.8 * 90~~
~~= 90~~
= 85
Therefore MEU(nothing) = max(U(R1), U(R2)) = U(R1) = 87

(b)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

With perfect info:
MEU = U(R1|~T)P(~T) + U(R2|T)P(T)
= 90*0.8 + 85 * 0.2 = 89

VPI = MEU(info) - MEU(noInfo) = 89 - 87 = 2

~~Clarifying the 2$.~~
~~MEU(T) = P(T) * U(R2|T) + P(not T)U(R1|T)~~
~~= 0.2 * 90 + 0.2 * 95~~
~~= 94~~

~~Value = MEU(T) - MEU(nothing) = 94 - 92 = 2$~~

~~I got $2 with above reasoning. (+2)~~

Question 7:

(a) Assuming that the rewards are yielded for a given state upon taking an action from that state:
(i) V(s1) = 1 + (0.9)[0.5V(s1) + 0.5V(s3)]
(ii) V(s2) = 2 + (0.9)[0.5V(s2) + 0.5V(s3)]
(iii) V(s3) = 1 + (0.9)V(s1)

(b) π* = { S1 -> B, S2 -> A, S3 -> A }

(c) V(s_t) ← V(s_t) + α[r_t + γV(s_t+1) - V(s_t)]

State/action pair s_t, a_t, r_t, s_{t-1}	**V(S1)**	**V(S2)**	**V(S3)**
S1, A, 1, S3	= 0 + 0.1(1 + 0.9 *0 - 0) = 0.1	0	0
S3, B, 1, S3	0.1	0	= 0 + 0.1(1 + 0.9*0 - 0) = 0.1
S3, A, 1 , S1	0.1	0	= 0.1 + 0.1(1 + 0.9*0.1 - 0.1) = 0.199

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

S1, B, 1, S2	= 0.1 + 0.1 (1 + 0.9*0 - 0.1) = 0.1 + 0.1(0.9) = 0.19	0	0.199
S2, A, 2, S2	0.19	= 0 + 0.1(2 + 0.9*0 - 0) = 0.1 * 2 = 0.2	0.199

Question 9:

(a)

(i) [simple] gradient descent is insufficient over spaces with local max/minima - use some form of randomized descent (simulated annealing)

(ii) gradient descent is insufficient in continuous search spaces of high dimension, where it is unfeasible to test the value of each neighbor configuration (such a situation forces you to use some randomized neighbor selection process which may result in suboptimal actions)

(b) NOT COVERED

Another limitation is that you need large amounts of data to train an NN which is not possible to obtain in most cases. We can solve this by trying algorithms that learn faster

Question 10:

(a) False - by exploiting conditional independence inference can be far cheaper than 2^n, making it practical in the real world

(b) True. Here, P(~A|~B) + P(A|B) = 1 since A and B are independent. Therefore, P(~A|~B) = P(~A) and P(A|B) = P(A). P(~A) + P(A) = 1.

(f) True

(i) True - by definition, any finite MDP must have an optimal policy. Bellman proved it (from the slides).

CAREFUL: this is False. Finite MDP must have an optimal value function, not optimal policy (think of Assignment #5)

- NO: an optimal value function implies that there exists an optimal policy yielded from V*. assignment 5 was asking about the **uniqueness** of the

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

optimal policy, which very well may not be unique. That's not what this question was asking

(j) False - why would the reward being negative have an impact? (yep, just add a constant that makes all of the rewards positive -> will give same output) They both converge… so it would be true. But do they converge to the same estimate? Why not ? Why would they? They're different methods that use different formula

(n) False - utility is a long-term of reward that takes into account different possible outcomes, reward is a simple function of state and action (Reward - immediate, return - long run)
Utility = sum of rewards along the way

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

COMP 424 FINAL 2006

Question 1:

a) This heuristic is admissible because log(1 + h(x)) =< h(x). H is a better heuristic because its estimate is closer to the true cost. (also due to domination) Can h(x) be < -1, so log(1 + h(x)) is undefined? Also, if g(x) = log(1 + h(x)) is a better admissible heuristic, then I'd argue f(x) = log(1 + g(x)) is even better, etc, etc… which doesn't seem to make sense. Heuristics are non negative. So it can't be < 0.
b) 1101010, 00010011
c) Two leftmost arrows are selected as moves, node 5 is pruned
d) A = 5, B = 6, C = 3, D = 4 for no pruning. A = 5, B = 4, C = ANY, D = ANY, for pruning. **The values here seems wrong for the first part. The max value is 10 at the moment and having a 5 and 6 value would lead the left branch to be pruned.t**
A=1,B=11 will work, A<10 and B<10 will casue C,D to be pruned
e) Yes, because the heuristic will be more precise at depth 4 than at depth 2. Not necessarily though.

Question 2:

a)

Shouldnt this be:
i) ∀a,b,m friends(a, b) -> (goingTo(a, m) <-> goingTo(b, m))

i) ∀a,b,m friends(a, b) ^ goingTo(a, m) → goingTo(b, m)
ii) ∀a,b,c (friends(a, b) ^ friends(b, c)) → friends(a, c)
iii) friends(amy, bert) ^ friends(bert, cleo)
iv) goingTo(amy, star wars)

My answers:

(i) Forall x,y,m AreFriends(x,y) ⇒ GoToMove(x,m) AND GoToMove(y,m)
(ii) AreFriends(x,y) AND AreFriends(y,z) ⇒ AreFriends(x,z)
(iii) AreFriends(A,B), AreFriends(B,C)
(iv) GoToMove(A, Starwars)

b) Proof to match with green text:
First simplify the knowledge base:

1. NOT AreFreinds(x,y) OR (GoToMovie(x,m) AND GoToMovie(y,m))
2. NOT AreFriends(x,y) OR NOT AreFriends(y,z) OR AreFriends(x,z)
3. AreFriends(A,B)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

4. AreFriends(B,C)
5. GoToMovie(A, Starwars)

Using CNF Resoltion and GMP unless stated:
Using 4 and 2 (x=A, y=B, z=C)
6. NOT AreFriends(A,B) OR AreFriends(A,C)
Using 3 and 6 :
7. AreFriends(A,C)
Using 7 & 1 (x=A,y=C,M=Starwars)
8. GoToMovie(A, Starwars) AND GoToMovie(C, Starwars)
Using And Elimination
9. GoToMovie(C, Starwars)

(Could also do this by adding NOT GoToMovie(C,STarwars) to the KB and deriving a contraditction)

MP iii) into ii)
    v) friends(amy, bert) ^ friends(bert, cleo) → friends(amy, cleo).
MP v) and iv) into i)
    vi) friends(amy, cleo) ^ goingTo(amy, star wars) → goingTo(cleo, star wars))

Question 3:
NOT COVERED

Question 4:
a = max_arg{x_n}
b = min_arg{x_n}

See
http://math.stackexchange.com/questions/49543/maximum-estimator-method-more-known-as-mle-of-a-uniform-distribution

Clarifying:
L(theta|x) = product (from i to n) of 1/(b-a) = (b-a)^-n
So we want to minimize (b-a) to maximize 1/(b-a)^n
Take a = max_arg{x_n} and b = min_arg{x_n}, this maximize the likelihood on the set of values x_1, x_2, .. x_n

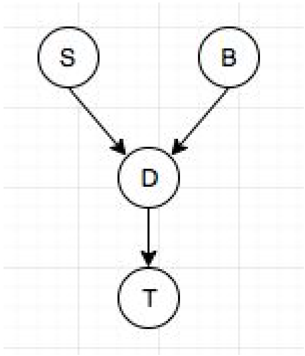
Question 5:
NOT COVERED

Question 6:

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Question 7:

(a)



Let S = sugar
Let B = butter
Let D (in net) = G (in calcs) = cake is delicious
Let T = high tip

From the text:

- P(B) = 0.1
- P(S) = 0.2
- P(G|S,B) = 0.8
- P(G|S,~B) = 0.5
- P(G|~S, B) = 0.5
- P(G|~S,~B) = 0.25
- P(T|G) = 0.5
- P(T|~G) = 0.25

(b)

P(T) = P(T|G)*P(G) + P(T|~G)*P(~G)

P(G)
= P(G|B,S)P(B)P(S) + P(G|~B,S)*P(~B)*P(S) + P(G|B,~S)*P(B)*P(~S) + P(G|~B,~S)*P(~B)*P(~S)
= 0.8 * 0.1 * 0.2 + 0.5 * 0.9 * 0.2 + 0.5 * 0.1 * 0.8 + 0.25 * 0.9 * 0.8
= 0.326
P(~G) = 1 - 0.326 = 0.674

Therefore:
P(T) = 0.5 * 0.326 + 0.25 * 0.674 = 0.3315

Customers leave high tips 33% of the time.

(c)

Let X_i be the probability that customer i leave a high tip (p = 0.3315)
Let X = X_1 + X_2 + … X_20

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Esperance (<-- you mean expectation) of a bernoulli variable:
E(X) = 20 * E(X_1) = 20 * 0.3315 = 6.63 ~ 7 (rounding to closest integer)

So ~7 customers will leave a tip.

(d)
P(T|B,S) = P(T|G)*P(G|B,S) + P(T|~G)*P(~G|B,S)
= 0.5 * 0.8 + 0.25 * 0.2
= 0.45
This time, customers leave a high tip 45% of the time.

(e)
MEU(nothing) = P(T) * U(T) + P(~T) * U(~T)
= 0.3315 * 10 + (1-0.3315)*1
= 3.9835

So MEU(nothing) ~ 4 why would you round these numbers?

MEU(high sugar and high butter) = P(T|B,S)*U(T) + P(~T|B,S)*U(~T)
= 0.45 * 10 + 0.55 * 1
= 5.05

So MEU(high sugar and high butter) ~ 5

Value of adding S and B = MEU(high S and high B) - MEU(nothing) = 1$

The expected gain of such a practice is 1$ per customer.

<u>Question 8:</u>
(a)
Let B = having a bone chip
Let O = having operation
P(B) = 0.4

<u>From the text:</u>
U(O|B) = -10
U(O|~B) = -10
U(~O|B) = -100
U(~O|~B) = 0

EU(operation) = P(B)*U(O|B) + P(~B)*U(O|~B)
= 0.4 * -10 + 0.6*-10

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

= -10

EU(no operation) = P(B)*U(~O|B) + P(~B)*U(~O|~B)
= 0.4 * -100 + 0.6 * 0
= -40

MEU(no info) = EU(operation) = -10

So the best choice is to get the operation

(b)

MEU(B) = P(B) * U(best action when having bone chip) + P(~B) * U(best action when no bone chip)
= P(B) * U(O|B) + P(~B)*U(~O|~B) -> this is obvious
= 0.4 * -10 + 0.6 * 0
= -4

Therefore the value of the information is

Value = MEU(B) - MEU(nothing) = -4 - (-10) = 10 - 4 = 6

(c)

(d)
From the MEU definition:

MEU(T) = P(T) * U(best action|T) + P(~T) * U(best action |~T)

Where P(T) = P(T|B)P(B) + P(T|~B)P(~B)
= 0.4x + 0.6y

P(~T) = P(~T|B)P(B) + P(~T|~B)P(~B)
= (1-x) * 0.4 + (1-y) * 0.6
= 1 - 0.4x - 0.6y

It is clear that the best action when test is positive is to get an operation and vice-versa (we can show it with computation of EU(no op|T) and EU(operation|~T)).
EU(operation|T) = P(B|T) * U(O|B) + P(~B|T) * U(O|~B)
= -10 * (P(B|T) + P(~B|T))
= -10 * 1

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

= -10

EU(no op | ~T) = P(B|~T) * U(O|B) + P(~B|~T) * U(~O|~B)
= P(B|~T) * -10 + P(~B|~T) * -10 * 10
= -10 * (P(B|~T) + P(~B|~T) * 10)
= -10 * OH FUCK IT

Question 9:

Can someone double-check this ? Looks good to me

(a)

Climb (child, stool):
Pre: In(child, r) ^ Height(child, low) ^ In(stool, r)
Post: add: Height(child, high) del: Height(child, low)

Grab (child, object):
Pre: In(child, r) ^ In(object, r) ^ Height(child, h) ^ Height(object, h)
Post: add: Holds(child, object)

Release (child, object):
Pre: Holds(child, object)
Post: del: Holds(child, object)

Go(child, a, b):
Pre: In(child, a)
Post: add: In(child, b) del: In(child, a)

(b)

Initial: In(child, A) ^ In(stool, B) ^ In(tree, C) ^ In(candy, C) ^ Height(child, low) ^ Height(candy, high)
Goal: Hold(child, candy)

(c)

Go(child, A, B) → Grab(child, stool) → Go(child, B, C) → Release(child, stool) → Climb(child, stool) → Grab(child, candy)

Question 10:

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

(a)

Policy: S2 -> stay
S1 -> switch

(b)

WITH MATRICES

$$\begin{pmatrix} v1 \\ v2 \end{pmatrix} = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - ½ * \begin{pmatrix} 0.2 & 0.8 \\ 0 & 1 \end{pmatrix} \right)^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The equation above is V = (I - yT)^(-1)*R ?

$$\begin{aligned} V &= (1-\gamma T)^{-1}R \\ &= \left(\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} - \frac{1}{2}\begin{pmatrix}0.2 & 0.8\\ 0 & 1\end{pmatrix}\right)^{-1}\begin{pmatrix}0\\1\end{pmatrix} \\ &= \begin{pmatrix}0.9 & -0.4\\ 0 & 0.5\end{pmatrix}^{-1}\begin{pmatrix}0\\1\end{pmatrix} \\ &= \frac{1}{0.45}\begin{pmatrix}0.5 & .4\\ 0 & 0.9\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} \\ &= \begin{pmatrix}\frac{8}{9}\\ 2\end{pmatrix} = \begin{pmatrix}0.888...\\ 2\end{pmatrix} \end{aligned}$$

solve matrix get v1 = 0.889
v2 = 2

matrix method has wrong answer for v1

Yeah just saw that, supposed to be ~0.88

WITHOUT MATRICES

V(S1) = R(S1,switch)+y*(T(s1,s2)*V(S2)+T(s1,s1)*V(s1))
= 0 + ½ *(0.8*V(S2)+0.2*V(S1))
= ½ * 0.8 * V(S2) + ½ * 0.2 * V(S1)
= 0.4 * V(S2) + 0.1 * V(S1)
= 0.889

V(S2) = R(S2, stay)+y*V(S2)
= 1 + ½ * V(S2)
= 2

IT HOLDS.

(c)

V0(S1) = 0

V0(S2) = 0

V1(S1) = R(S1, switch) + y(0.8 * V0(S2) + 0.2*V0(S1))
= 0

V1(S2) = R(S2, stay) + y(V0(S2))

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

= 1

V2(S1) = R(S1, switch) + y(0.8*V1(S2)+0.2*V1(S1))
= 0 + y(0.8 * 1 + 0.2*0)
= 0 + 0.5 * 0.8
= 0.4
V2(S2) = R(S2, stay) + zxyz`(V1(S2))
= 1 + 0.5 * (1)
= 1.5~
(If we continue like this, it won't converge, is that right?)

(d)

V(S1) = V(S1) + alpha(R + V(S2) - V(S1))
= 0 + 0.1(0 + 0 - 0) = 0
V(S2) = V(S2) + alpha(R + V(S2) - V(S2))
= 0 + 0.1(1 + 0 - 0)
= 0.1
2006
<u>9Question 11:</u>
Quite likely to not get questions like this since we are going to be writing our solutions on a midterm like answer sheet.

a) Split sky into grid . State={any cell}, Obs.={blips in that cell}. T=airplane speed? E=radio wave
   Time to bullshit like no tomorrow LMAO
b)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

COMP 424 FINAL 2010

http://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-final-2010.pdf

We're doing this on the fly, comment in red if anything looks wrong!

Question 1

(a)

a*h is admissible when a <= 1. Otherwise it overestimates and is not admissible.

In A* the same condition applies since the heuristic must be admissible for it to be used in A*. If a <= 1, a*h could be used - but it should not since it would be less optimal.

(b)

If k was sufficient to solve the game, you would get the same result with k+1. If the maximum depth of the game is higher than k, then k+1 depth would be guaranteed to generate a better player since there is more potential game states being examined.

(c)

No, because there's a possibility that at a certain level a certain successor will have a low score, but that after a few depth iterations it will prove to be a better solution.

Question 2

Variables: Every cell of the game that is uncovered, Every cell of the game that is unknown
Domain: 0 through 8 for known cells, and -1 for unknown cells

The constraint:
For each cell of the game that is uncovered there is "value of the cell" neighbors with a bomb.
We should not uncover a cell with a bomb.

Question 3

(a)

i. $\forall r$ Robot(r) $\rightarrow$ Smart(r)

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Ii. ∃r Robot(r) → Smart(r)
iii. Robot(Robbie)
iv. ∀r, o Robot(r) ^ Owns(o, r) → Nice(r, o)
v. Owns(Bob, Robbie)
vi. ∃o, r Owns(o, r) → Smart(o)
Should this be for all r, since a owner a multiple robot is always smart → where do you see this ? If Owns(o,r), then o is smart of all Owns(o,r), since it is the same person. vi) says that "Some people who own robots are smart". And this FO statement says "There exists a person O and a robot R such that O owns R, such that O is smart". Any better way to do it ?
Aren't we supposed to use ∃o, r Owns(o, r) ^ Smart(o) with ∃ ? ← I think this is right? Also is ii. not ∃r Robot(r) ^ Smart(r)?

(b)

AI of iii), v)

vii. Robot(Robbie) ^ Owns(Bob, Robbie)

MP vii) into iv)

viii. Robot(Robbie) ^ Owns(Bob, Robbie) → Nice(Robbie, Bob)

(c)

This can't be proven, since vi) only says that some people owning robots are smart. As such we have no proof that Bob is one of them.

Question 4
NOT COVERED

Question 5
NOT COVERED

Question 6
NOT COVERED

Question 7
P(A) = 0.1, P(~A) = 0.9, P(B) = 0.5, P(~B) = 0.5

(a) P(at least 1) = P(A) * P(B) + P(A) * P(~B) + P(~A) * P(B) = 0.55
= P(A) + P(~A)*P(B) if factorized

(b) P(exactly 1) = P(A) * P(~B) + P(~A) * P(B) = 0.5

GOOD LUCK EVERYONE PEACE & LOVE
^ THANKS YO
PAIN IS TEMPORARY
GPA IS FOREVER

Question 8

(a)

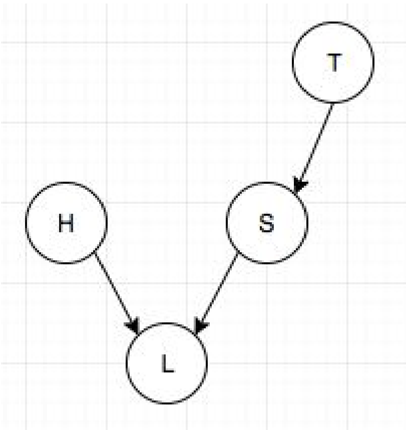
(possible answer)

Set H = they are happy
Set S = they like the sci-fi
Let NS = they like the non sci-fi

$P(H) = 0.6$
$P(S|H) = P(NS|H) = 0.8$
$P(S|\sim H) = 0.8$
$P(NS|\sim H) = 0.5$ <- in the text

Happy
/ \
/ \
Like the Sci fi
Like the non-sci-fi

(**other possible answer)**



$P(L|H) = 0.8$
$P(L|S, \sim H) = 0.8$
$P(L|\sim S, \sim H) = 0.5$
$P(L|S,H) = 0.8$
$P(L|\sim S,\sim H) = 0.5$

$P(H) = 0.6$
$P(S) = 0.5$

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

(b)

(first answer)

P(they liked the movie)

= P(sci-fi show) * (P(S|H)P(H) + P(S|~H)*P(~H)) + P(no sci-fi show) * (P(NS|H)P(H) + P(NS|~H)P(~H))

= 0.5 * (0.8 * 0.6 + 0.8 * 0.4) + 0.5 * (0.8 * 0.6 + 0.5 * 0.4)

= 0.74

(other answer)

P(L)

= P(sci-fi-show) * (P(L|H, S)P(H) + P(L|~H,S)P(~H))

+ P(no-sci-fi) * (P(L|H,~S)P(~H) + P(L|~H, ~S)P(~H))

= 0.5 * (0.8 * 0.6 + 0.8 * 0.4)

+ 0.5 * (0.8 * 0.6 + 0.5 * 0.4)

= 0.74

(c)

7.4 ~ 7

They will enjoy it for 7 nights on average.

(d)

Modification to the bayes network:

(first answer)

Impossible to incorporate.

(other answer)

Add a node B (buying popcorn) under S.

We can’t without information on the other customers. Maybe they consume high amounts of popcorn during non-sci-fi movies. We also have no data on the proportion of students vs non-students.

GOOD LUCK EVERYONE PEACE & LOVE

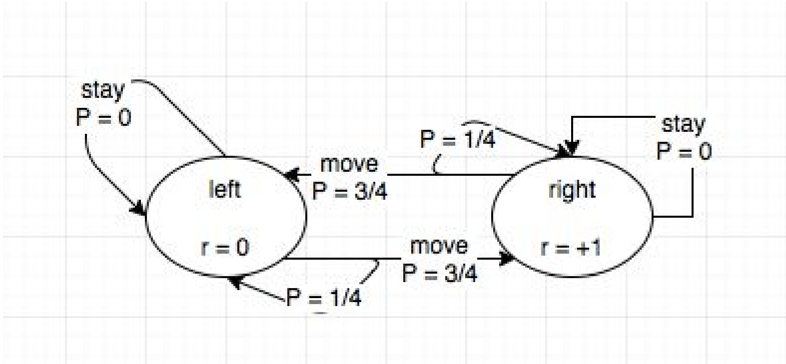
^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

Question 9

(a)



Did you mean P(stay) = 1 for both cases ? Yes! I didn't save the graph though. Working too fast lol. So just assume it's 1. Ok! Just checking.

(b)

π*(s) = { left → move, right → stay }

(c)

V(left) = r(left, move) + y * [T(left, right) * V(right) + T(left, left) * V(left)]
= 0 + 1/2 * [3/4 * V(right) + 1/4 * V(left)]
= 0 + 1/2 * [3/4 * 2 + 1/4 * V(left)] <- why is V(right) = 2?
= 6/7

V(right) = r(right, stay) + y * V(right)
= 1 + 1/2 * V(right) <- isnt V(right) suppose to be 1 so the answer is 1.5? No it's the optimal not, after one iteration, it is 2 right nvm got it
= 2

(d)

↓↓ can a TD-update pro double-check this ?

L, stay, 0, L	V(L) = V(L) + 0.1(0 + yV(L) - V(L)) = 0	V(R) = 0
L, stay, 0, L	V(L) = V(L) + 0.1(0 + yV(L) - V(L)) = 0	V(R) = 0
L, stay, 0, L	V(L) = V(L) + 0.1(0 + yV(L) - V(L)) = 0	V(R) = 0
L, move, 0, R	V(L) = V(L) + 0.1(0 + yV(R) - V(L)) = 0	V(R) = 0
R, stay, 1, R	V(L) = 0	V(R) = V(R) + 0.1(1 + yV(R) - V(R)) = 0.1
R, move, 1, L	V(L) = 0	V(R) = V(R) + 0.1(1 + yV(L) - V(R)) = 0.19

(+1)

GOOD LUCK EVERYONE PEACE & LOVE

^ THANKS YO

PAIN IS TEMPORARY

GPA IS FOREVER

Gotcha bro You forgot the discount factor every time, okkk never mind there is no discount factor (wtf ?) Yep that's good. Lol. Thanks for the review. There is a discount factor specified in c) (½) so that must be it. But wherever it's used the V(s) = 0 so it's a wash anywayhttp://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-final-2010.pdf

(e) Is the starting policy pi(s,l):stay? I think that's what they mean. Weird question.

Question 10