# CS486 Practice Questions

## 0.1 Question

**[20 points]** Suppose we want to write a program for the following task: the program will be given a table of numbers, such as

| x1 | x2 | x3 | x4 | y |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 0 |
| 2 | 3 | 5 | 1 | 10 |
| 1 | 7 | 4 | 2 | 9 |

and an expression such as

```
y = (f1 (f2 x1 x2)
        (f3 x3 x4))
```

where the x's are variables that hold real numbers, and the f's are functions. We want to find a set of definitions for these functions so that the expression as a whole computes the results given in the table, e.g. so that if x1 is 1, x2 is 1, x3 is 2, and x4 is 3, the value of the whole expression is 0. Each f must be defined to be one of +, *, -, or /. E.g., a solution for the specific problem given above is

```
f1 = +, f2 = *, f3 = -
```

a. Formulate this puzzle in terms of State Space Search. Include an informal description of each of the following:
   - A representation for states.
     partial assignment of ops to fs
   - A set of operators, their state transitions and the applicability condition (if any) for each to avoid resulting in an illegal state.
     add an assignment of an op to an f. if last op assigned, dont assign it so that a line of the table is wrong
   - A goal test.
     fs have ops

## 0.2 Question

2. **[25 Points]** Suppose you are building a knowledge-based system to plan the seating at a dinner party. You want to use the system to prove statements of the form $OK(P1, P2)$ (or $\neg OK(P1, P2)$) meaning it is (or is not) OK for person $P1$ to sit next to person $P2$. Suppose you give the system the following axioms:

   $(\forall x, y) Dislikes(x, y) \rightarrow \neg OK(x, y)$

   $(\forall x, y) Male(x) \wedge Male(y) \rightarrow \neg OK(x, y)$

   $(\forall x, y) Female(x) \wedge Female(y) \rightarrow \neg OK(x, y)$

   $(\forall x, y) \neg Dislikes(x, y) \wedge Male(x) \wedge Female(y) \rightarrow OK(x, y)$.

   $(\forall x, y) \neg Dislikes(x, y) \wedge Female(x) \wedge Male(y) \rightarrow OK(x, y)$.

   $Male(John) \wedge Female(Susan) \wedge Male(David) \wedge Female(Jane)$

   $Dislikes(Susan, Dave) \wedge Dislikes(Jane, John)$

   (a) Do the axioms entail $OK(Susan, John)$? Explain.
   No, to conclude this we would have to have ¬ Dislikes(Susan, John)

   (b) Do the axioms entail $\neg OK(Susan, John)$? Explain.
   No, to conclude this we would have to have Dislikes(Susan, John)

   (c) Do the axioms entail $OK(Susan, John)$ under the Closed World Assumption? Explain.
   Oops - we did not cover the Closed World Assumption. Ignore this question

   (e) Which of the answers (a,b and/or c) will change if we rewrite all the axioms by replacing each atom $Dislikes(\ldots)$ with $\neg Likes(\ldots)$? Explain.
   None (except c which we are ignoring)

   (a) **[30 Points]** Four criteria were defined for comparing search strategies: completeness, optimality, time complexity, and space complexity. Which one (1) of these four criteria **best** explains why:

      i. Iterative-Deepening search is usually preferred over Breadth-First search.
      Space complexity

      ii. Iterative-Deepening search is usually preferred over Depth-First search.
      completeness

      iii. Alpha-beta Minimax search is usually preferred over pure Minimax search.
      Time complexity

      iv. A* search is usually preferred over Greedy search.
      optimality

      v. Algorithm X might be preferred over A* search.
      space complexity

      vi. Bidirectional search is usually preferred over Breadth-First search.

   time complexity

## 0.3 Question

[10 points] Convert each of the following to Conjunctive Normal Form.

- $\forall d\ (dog(d) \Rightarrow \exists p\ (person(p) \wedge owns(p,d)))$
  $\forall d\ (\neg dog(d) \vee \exists p\ (person(p) \wedge owns(p,d)))$
  $\forall d\ (\neg dog(d) \vee (person(s(d)) \wedge owns(s(d),d)))$
  $(\neg dog(d) \vee person(s(d))) \wedge (\neg dog(d) \vee owns(s(d),d))$
- $\exists d\ (dog(d) \wedge \forall f\ (dogfood(f) \Rightarrow eats(d,f)))$
  $\exists d\ (dog(d) \wedge \forall f\ (\neg dogfood(f) \vee eats(d,f)))$
  $(dog(s) \wedge \forall f\ (\neg dogfood(f) \vee eats(s,f)))$
  $(dog(s) \wedge (\neg dogfood(f) \vee eats(s,f)))$
  $(dog(s) \wedge \neg dogfood(f)) \vee (dog(s) \wedge eats(s,f))$

## 0.4 Question

Consider the following ways in which one might represent the assertion "everybody loves somebody sometime" in first order logic.

(i) $\forall x\, \exists y\, \exists t\ loves(x,y,t)$
(ii) $\exists t\, \forall x\, \exists y\ loves(x,y,t)$
(iii) $\exists y\, \forall x\, \exists t\ loves(x,y,t)$
(iv) $\exists y\, \exists t\, \forall x\ loves(x,y,t)$

(a) (1%) For each pair of sentences $\alpha$ and $\beta$ from this list, state which of the following holds:

1. $\alpha \models \beta$ and $\beta \not\models \alpha$
2. $\alpha \not\models \beta$ and $\beta \models \alpha$
3. $\alpha \not\models \beta$ and $\beta \not\models \alpha$
4. $\alpha \models \beta$ and $\beta \models \alpha$.

(That is, for each of the six possible pairs, indicate one of these four possible relations. Note that $\models$ assumes there is at least one object in the domain.)

(b) (1%) Give a set of people and time points, which defines a domain set $D$, and a specification of the relation "loves" over $\langle$*person, person, time point*$\rangle$ triples, $R$, such that $\langle D,R\rangle \models$ (i) but $\langle D,R\rangle \not\models$ (iv).

## 0.5 Question

[20 points] Given the following clauses as the knowledge base, show a proof by resolution refutation that cheese(Moon):

$\neg$ large(x) $\vee$ $\neg$ green(x) $\vee$ cheese(x)
large(x) $\vee$ $\neg$ p(y) $\vee$ $\neg$ orbit(x, y)
$\neg$ rock(x) $\vee$ green(x)
p(Earth)
orbit(Moon, Earth)
orbit(Earth, Sun)
rock(Moon)

$\neg$ large(x) $\vee$ $\neg$ green(x) $\vee$ cheese(x), $\neg$ cheese(moon) $\rightarrow$ $\neg$ large(moon) $\vee$ $\neg$ green(moon)

$\neg$ large(moon) $\vee$ $\neg$ green(moon), $\neg$ rock(x) $\vee$ green(x)$\rightarrow$ $\neg$ rock(moon) $\vee$ $\neg$ large(moon)
$\neg$ rock(moon) $\vee$ $\neg$ large(moon), large(x) $\vee$ $\neg$ p(y) $\vee$ $\neg$ orbit(x, y)$\rightarrow$ $\neg$ rock(moon) $\vee$ $\neg$ p(y) $\vee$ $\neg$ orbit(moon, y)
$\neg$ rock(moon) $\vee$ $\neg$ p(y) $\vee$ $\neg$ orbit(moon, y),rock(moon) $\rightarrow$ $\neg$ p(y) $\vee$ $\neg$ orbit(moon, y)
$\neg$ p(y) $\vee$ $\neg$ orbit(moon, y),p(Earth) $\rightarrow$ $\neg$ orbit(moon,earth)
$\neg$ orbit(moon,earth), orbit(moon, earth) $\rightarrow$ false

## 0.6 Question

**[25 points]** For each of the following statements, express it in First Order Predicate Calculus or explain why this cannot be done.

- Every professor teaches at least one class.
  $\forall p\ professor(p) => \exists c\ class(c) \wedge teaches(p, c)$
- Every student likes at least one class that he takes.
  $\forall s\ student(s) => \exists c\ class(c) \wedge takes(s, c) \wedge likes(s, c)$
- Freshmen typically take the class CS111.
  Cannot be done - can't express "typically" in FOL
- No class has both freshmen and seniors taking it.
  $\neg\exists c\ class(c) \wedge \exists f\ \ freshman(f) \wedge takes(f, c)) \wedge \exists s\ senior(s) \wedge takes(s, c)$
- Every student who is taking a class taught by Prof. Steinberg is also taking a class taught by Prof. Kaplan.
  $\forall s\ (student(s) \wedge \exists c\ course(c) \wedge teaches(Steinberg, c) \wedge takes(s, c)) => \exists c'\ course(c') \wedge teaches(Kaplan, c') \wedge takes(s, c')$
- Everything that is true for all birds is true for all eagles.
  Can't do in FOL; requires quantiers on predicates.

## 0.7 Question

Imagine you are playing "Let's Make a Deal" with Monty Hall. There are three doors labelled 1, 2 and 3, with nothing behind two of the doors, but a prize worth $9,999 behind one of them (you don't know which). Here is how the game is played: You pick a door $a = 1, 2$ or 3. Monty then opens one of the *other* doors $c \neq a$ which does *not* contain the prize. You are then left with two doors—the one you originally picked, $a$, and the one that Monty left closed, $b$. Monty offers you a final choice: you can stay with your first choice $a$ or you can switch to the second door $b$. The question is, should you switch? You will answer this by determining the value of the information that Monty is giving you by revealing $c$ and leaving $b$.

Let $S$ be a random variable that indicates which door contains the prize. Our initial beliefs are $\mathrm{P}(S{=}1) = \mathrm{P}(S{=}2) = \mathrm{P}(S{=}3) = \frac{1}{3}$. Your actions are to pick a door $a = 1, 2$ or 3. The reward function is $R(S,a) = \begin{cases} \$9,999 & \text{if } S = a \\ 0 & \text{if } S \neq a \end{cases}$

(a) (1%) What is the expected reward of your initial choice $a$?
(Obviously, $\mathrm{P}(S{=}a) = \frac{1}{3}$ for any $a$.)

After you have picked a door $a$ Monty then reveals one of the other doors $c$ that does not contain the prize; leaving the alternate door $b$. Note that Monty's choice of $c$ (and $b$) depends on $a$ and $S$. That is, Monty knows which door $S$ contains the prize and what your initial choice $a$ was, and he uses this information to select $c$ and $b$. Thus, $c$ and $b$ are actually functions of $a$ and $S$. Therefore, we really should write $c$ and $b$ as

$$\begin{aligned} c(a,S) &= s' \quad \text{for some } s' {\neq} S \\ b(a,S) &= \begin{cases} S & \text{if } S \neq a \\ s' & \text{for some } s' {\neq} S, \text{ if } S = a \end{cases} \end{aligned}$$

(b) (1%) What is the expected reward of choosing the second door $b(a,S)$?
To determine this, use the general fact that

$$\begin{aligned} \mathrm{P}\left(S{=}b(a,S)\right) &= \mathrm{P}\left(S{=}b(a,S)|S{=}a\right)\ \mathrm{P}(S{=}a) \\ &+\ \mathrm{P}\left(S{=}b(a,S)|S{\neq}a\right)\ \mathrm{P}(S{\neq}a) \end{aligned}$$

(c) (1%) What is the value of the information that Monty is giving you by revealing door $c$? (That is, how much should you be willing to pay Monty for this information?) Determine the answer by considering the improvement (if any) in your expected optimal reward given the information *over* your expected optimal reward without the information. (That is, just take the difference between these two quantities.)

(d) (0%) Should you switch?

## 0.8 Question

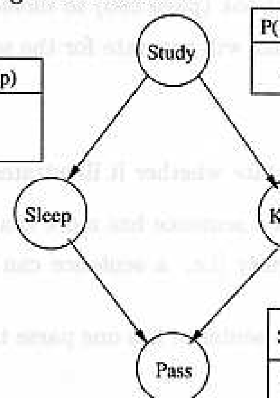
10. A certain test is available for detecting the presence of a viral infection, although the test is not 100% accurate. Assume that 1/4 of the population is infected, and the other 3/4 is not. Assume that 90% of those infected test positive and 80% of uninfected persons test negative. Given that the test is positive, what is the probability that the person is infected?

$I$ a person is infected, $T$ - test comes out positive. We need to evaluate $P(I|T)$. By the normalized Bayes' Rule

$$P(I|T) = \frac{P(T|I)P(I)}{P(T|I)P(I) + P(T|\neg I)P(\neg I)} = \frac{.9 \times .25}{.9 \times .25 + .2 \times .75} = \frac{.225}{.375} = 0.6 = 60\%$$

## 0.9 Question

You are given the following belief network:




| P(Study) |
|---|
| .7 |

| Study | P(Sleep) |
|---|---|
| T | .3 |
| F | .8 |

| Study | P(Know) |
|---|---|
| T | .8 |
| F | .4 |

| Sleep | Know | P(Pass) |
|---|---|---|
| T | T | .9 |
| T | F | .6 |
| F | T | .5 |
| F | F | .1 |

Consider the dilemma of whether to stay up late to study for the exam, and compute

(a) the probability $P(Study, Sleep, Know, \neg Pass)$

(b) the probability $P(Study, Sleep, \neg Know, Pass)$

(c) the probability of $P(Study, Sleep, Pass)$

Show your reasoning.

## 0.10 Question

Consider the set of compound propositions $A = \{p \to q,\ T \to p \vee r,\ r \to s,\ s \to \perp\}$. You will conduct three searches for a satisfying truth assignment for $A$ (*i.e.*, an assignment of truth values to the atomic propositions $p, q, r, s$ that makes every rule in $A$ evaluate to $T$).

(a) (1%) First perform a simple backtracking search: That is, conduct a depth first search in partial truth assignments to $p, q, r, s$ (assigning a value to $p$ first, and then $q$, *etc.*, trying the value $F$ before $T$), until a satisfying truth assignment is found, or every plausible assignment has been considered. Note that you should backtrack immediately if any rule in $A$ becomes falsified.

Show the primitive assignment steps. How many primitive assignments did you make?

(b) (1%) Perform a backtracking search with pruning (unit propagation): That is, conduct a depth first search in partial truth assignments to $p, q, r, s$, but immediately assign a satisfying value to any variable that becomes the sole unassigned variable in a rule. (The other assignment value can be ignored because it will never lead to a satisfying assignment for the rule unless earlier assignments are undone.) Backtrack whenever a rule becomes falsified.

Show the primitive assignment steps. How many primitive assignments did you make?

(c) (1%) Perform a greedy local search starting with the assignment $p = F$, $q = F$, $r = F$, $s = F$. In particular, given a current complete assignment, consider each of its four neighboring assignments (obtained by flipping each one of the current variable values independently). If a satisfying assignment to the set $A$ is found, halt. Otherwise, go to the neighboring assignment that satisfies the most rules in $A$, and repeat. (Note: break ties arbitrarily, but always move to a new neighboring assignment.) Stop if you haven't found a satisfying assignment within four steps.

Show the complete assignments considered. How many did you consider?

## 0.11 Question

Suppose that in your local nuclear power station there is an alarm that senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Consider the variables A (alarm sounds), FA (alarm is faulty), FG (gauge is faulty), G (gauge reading), and T (actual temperature).

(a) Construct an appropriate belief network, given that the gauge is more likely to fail when the core temperature gets too high. Explain breifly your reasoning.

(b) Is your network a polytree? If not, why not? [Hint: do not take this question to imply it should be a polytree!]

## 0.12 Question

Consider the class of linear functions defined by a weight vector $\mathbf{w}$, $H = \{h_{\mathbf{w}} : \mathbf{w} \in I\!R^n\}$, where the prediction made by $h_{\mathbf{w}}$ on an input vector $\mathbf{x}$ is given by $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. Assume you are given training data

$$X, \mathbf{y} \;=\; \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{t1} & \dots & x_{tn} \end{bmatrix}, \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix}$$

In class we saw that the weight vector $\mathbf{w}$ that minimized the sum of squared prediction error

$$SSE \;=\; \sum_{i=1}^{t} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2$$

was given by the weight vector $\mathbf{w}$ that satisfied $X^\top X \mathbf{w} = X^\top \mathbf{y}$.

One problem with the *SSE* training objective is that is does not penalize large weight values, which can lead to over-fitting. In practice, it is usually a much better idea to minimize a combination of prediction error and weight size

$$REG \;=\; \sum_{i=1}^{t} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

where $\lambda$ is just a fixed trade-off parameter.

(a) (1%) Calculate $\frac{\partial REG}{\partial w_k}$ for this alternative training criterion.

(b) (2%) The weight vector $\mathbf{w}$ that minimizes *REG* must satisfy

$$\begin{bmatrix} \frac{\partial REG}{\partial w_1} \\ \vdots \\ \frac{\partial REG}{\partial w_n} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Rewrite this system of equations in matrix form (similar to $X^\top X \mathbf{w} = X^\top \mathbf{y}$). *Hint:* The matrix form is very similar to the previous case, but with one new term. To express this new term, use the identity matrix $I$.

Therefore, just as for *SSE*, the *REG* criterion can be minimized by solving a linear system.

## 0.13 Question

Suppose you are given a choice between (i) a guaranteed \$1,000,000 or (ii) a bet on a coin toss where you receive \$2,000,000 if a head turns up and \$10 is a tail turns up.

(a) What is the expected monetary value of the two choices?

(b) What would you choose? Explain your reasoning.

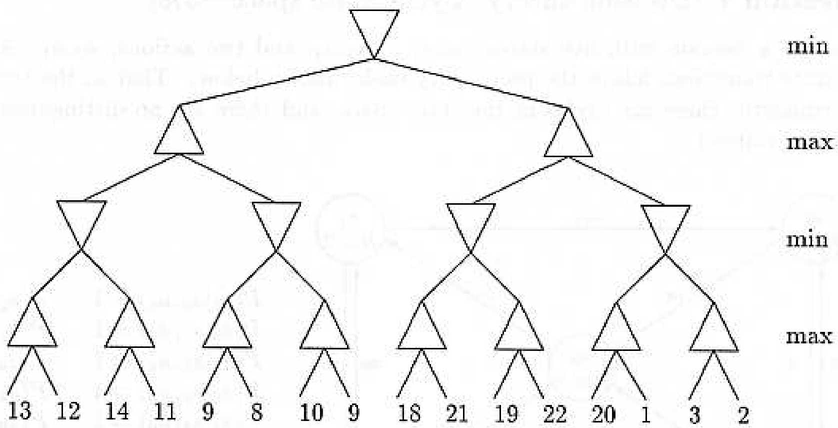
## 0.14 Question

(a) Suppose we need to compute $P(H|E_1, E_2)$ and we have no information about conditional independence. Which of the following set of numbers is sufficient to do the calculations:
$P(E_1, E_2), P(H), P(E_1|H), P(E_2|H)$
$P(E_1, E_2), P(H), P(E_1, E_2|H)$
$P(H), P(E_1|H), P(E_2|H)$

(b) Now suppose you know that:
$P(E_1|H, E_2) = P(E_1|H)$
Will your answer to the previous question change?

## 0.15 Question

Consider the following game tree.




(a) (1%) Compute the minimax values for each node, and show the optimal action choices for each node.

(b) (2%) Show the expansion steps that $\alpha$-$\beta$ pruning makes on this game tree. Assume that nodes are expanded left to right. Clearly indicate which subtrees are not explored.

## 0.16 Question

**[20 points]** The Farmer, Fox, Goose, and Grain problem is described as follows: A farmer wants to move himself, a silver fox, a fat goose, and some tasty grain across a river, from the west side to the east side. Unfortunately, his boat is so small he can take at most one of his possessions across on any trip. Worse yet, an unattended fox will eat a goose, and an unattended goose will eat grain, so the farmer must not leave the fox alone with the goose or the goose alone with the grain. What is he to do?

(a) Formulate the Farmer, Fox, Goose, and Grain problem in terms of State Space Search. Include an informal description of each of the following:

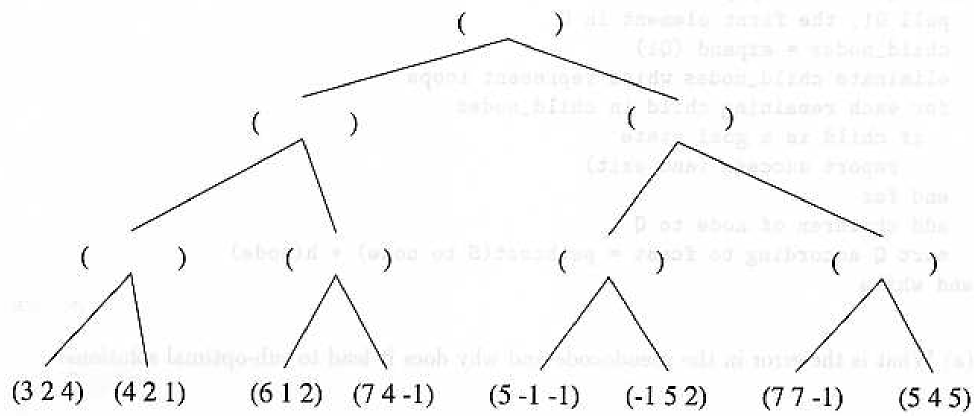
- A representation for states.
  A state is a tuple of 4 booleans, one for each of farmer, fox, goose, and grain, which is true if they are on the east side of the river and false otherwise
- A set of operators, their state transitions and the applicability condition for each to avoid resulting in an illegal state (one in which something gets eaten). How many operators are needed?
  Note: You need not describe every single operator in detail but your sample should be representative.
  There are 4 operators: the farmer crosses alone or with one of his three possessions. In each case the precondition is that the fox and goose booleans not be the same unless one of them is being taken with the farmer, and the same for the goose and the grain. E.g., for farmer crosses alone, the precondition is goose not equal fox and goose not equal grain.
- A goal test.
  All booleans are true
- A non-trivial admissible heuristic evaluation function.
  The number of false booleans divided by 2.

**You should provide the best solutions you can think of.**

## 0.17 Question

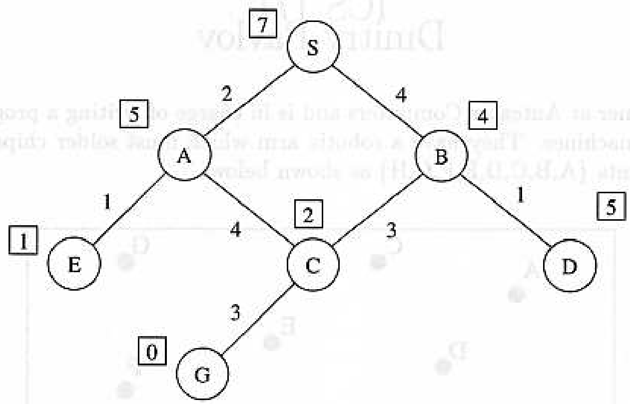
Consider the problem of search in a three player game. You may assume that there are no alliances. The evaluation function now has three values $(x, y, z)$, so that $x$ is the value of the state for player 1, $y$ is the value for player 2 and $z$ is the value for player 3. Player 1 moves first (at the root), followed by player 2 and then 3.

Extend minimax search to 3 player games and complete the following game tree by filling in the backed up value triples for all nodes. Indicate the move chosen by player 1 by circling the corresponding state.



## 0.18 Question

Shown below is a graph representing a map navigation problem.




The path cost is shown by the number on the links; the heuristic evaluation is shown by the number in the box. Assume that during search

- S is the start state and G is the goal state
- when placing nodes on the queue use alphabetical ordering to break ties
- assume that we never generate child nodes that appear as ancestors of the current node in the search tree

(a) What is the order that breadth first search will expand the nodes?

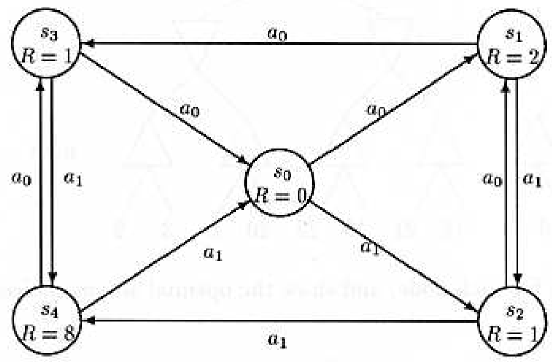
(b) What is the order that iterative deepening search will expand the nodes?

(d) What is the order that A* search will expand the nodes?

(e) What is the order that IDA* search will expand the nodes?

## 0.19 Question

Consider a domain with five states, $s_0, s_1, s_2, s_3, s_4$, and two actions, $a_0, a_1$. Assume that the state transitions follow the probability tables shown below. (That is, the transitions are deterministic, there are cycles in the state space, and there are no distinguished initial or terminal states.)




$P(s_1|s_0, a_0) = 1$ $\quad$ $P(s_2|s_0, a_1) = 1$
$P(s_3|s_1, a_0) = 1$ $\quad$ $P(s_2|s_1, a_1) = 1$
$P(s_1|s_2, a_0) = 1$ $\quad$ $P(s_4|s_2, a_1) = 1$
$P(s_0|s_3, a_0) = 1$ $\quad$ $P(s_4|s_3, a_1) = 1$
$P(s_3|s_4, a_0) = 1$ $\quad$ $P(s_0|s_4, a_1) = 1$

Use a discount factor of $\gamma = 0.75$, and consider an initial policy $\pi_0$ given by

| $s$ | $\pi_0(s)$ |
|---|---|
| $s_0$ | $a_0$ |
| $s_1$ | $a_0$ |
| $s_2$ | $a_1$ |
| $s_3$ | $a_1$ |
| $s_4$ | $a_0$ |

(a) (2%) Compute the *value function* for the fixed initial policy $\pi_0$:

$$V(s, \pi_0) = R(s) + \gamma \sum_{s'} V(s', \pi_0) P(s'|s, \pi_0(s))$$

Note that this reduces to solving: $\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ 8 \end{bmatrix} + \frac{3}{4} \begin{bmatrix} v_1 \\ v_3 \\ v_4 \\ v_4 \\ v_3 \end{bmatrix}$

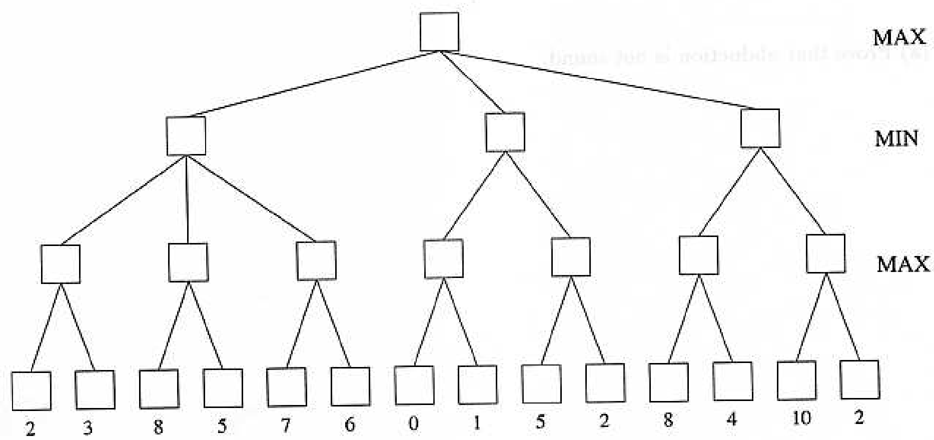
(b) (1%) Determine a better policy $\pi_1$ by making locally optimal updates to $\pi_0$. That is, for each state $s$, compute

$$\pi_1(s) \quad = \quad \arg\max_a \; R(s) + \gamma \sum_{s'} V(s', \pi_0) P(s'|s, a).$$

(c) (1%) What is the value function, $V(s, \pi_1)$, for the new policy $\pi_1$?
Is this the optimal achievable policy?

## 0.20 Question

Consider the following search tree where the evaluation function values are shown below each leaf node.




(a) Indicate the nodes that are pruned in minimax search with alpha-beta pruning by placing an X in the corresponding box. By pruning we mean that alpha-beta does not calculate or back up a value for that node. Assume that the root node is the maximizing player and that the search proceeds in a left to right fashion.

(b) Circle the state that represents the move chosen by MAX.

## 0.21 Question

Consider the following English statements:

1. Anyone who passes this exam and wins the lottery is happy.
2. Anyone who studies hard passes this exam.
3. Anyone who is lucky passes this exam.
4. John did not study hard for this exam but he is lucky.
5. Anyone who is lucky wins the lottery.
6. If John passes this exam then so will Mary.

(a) Translate the English into first order predicate logic. Use the predicate $pass(x)$ to mean that "$x$ passes this exam".

(b) Convert the statements into conjunctive normal form.

(c) Prove that John is happy using resolution with refutation (contradiction).

## 0.22 Question

You are working at NASA and are in charge of designing the navigation and exploration system for the Mars Rover II. The rover is supposed to explore the terrain of Mars and find interesting rocks. You are simulating the performance of the rover and have programmed in it the following utility(action,state) matrix:

| | | State | | |
|---|---|---|---|---|
| | | $(R, \neg D)$ | $(\neg R, \neg D)$ | $D$ |
| Action | $M$ | 30 | -1 | -100 |
| | $\neg M$ | -1 | -1 | -1 |

There are two actions: $M$ move forward and $\neg M$ do not move. There are three possible states: $(R, \neg D)$ interesting rock and no dangers ahead, $(\neg R, \neg D)$ no interesting rocks and no dangers ahead, and finally $D$ dangers ahead (in this case it doesn't matter if there are interesting rocks because the rover will be damaged if it moves).

(a) Calculate the expected utility of the actions moving and not moving given that $P(R, \neg D) = 0.5$, $P(\neg R, \neg D) = 0.4$ and $P(D) = 0.1$

(b) Suppose that we are unsure of the probabilities assigned to $P(\neg R, \neg D)$ and $P(D)$. How much can $P(D)$ change before the action with the maximum expected utility changes? You can assume that $P(R, \neg D)$ is fixed and is equal to 0.5, and that $P(\neg R, \neg D) = 0.5 - P(D)$.

## 0.23 Question

A lie detector test is known to be 80% reliable when the person is guilty (i.e., if a person is guilty, then there is an 80% chance they will "fail" the test). A lie detector test is 95% reliable when the person is innocent (i.e. if a person is innocent, then there is a 95% chance they will "pass" the test). A suspect is chosen from a group of people of whom 1% are guilty.

(a) Let $G$ stand for guilty ($\neg G$ for innocent) and $L$ stand for failing the lie detector test ($\neg L$ for passes the lie detector test). Find the following probabilities:

$P(G) =$

$P(L) =$

$P(L|G) =$

$P(\neg L|G) =$

$P(L|\neg G) =$

$P(\neg L|\neg G) =$

(b) The suspect has failed the lie detector test. What is the probability that the suspect is innocent (i.e., compute $P(\neg G|L)$)?

## 0.24 Question

1. Determine whether the following argument is logically correct by representing each sentence as a proposition and checking whether the conclusion is logically implied by the conjunction of the assumptions.

"If Alice wins the nomination, she will be happy, and if she is happy, she is not a good campaigner. But if she loses the nomination, she will lose the confidence of the party. She is not a good campaigner if she loses the confidence of the party. If she is not a good campaigner, then she should resign from the party. Therefore, she should resign from the party." (Introduce propositional variables $N, H, C, P, R$).

$N$ - A. wins the nomination; $H$ - A. is happy; $C$ - A. is a good campaigner; $P$ - A. loses the confidence of the party; $R$ - A. should resign.

*KB*: 1. $N \Rightarrow H$
2. $H \Rightarrow \neg C$
3. $\neg N \Rightarrow P$
4. $P \Rightarrow \neg C$
5. $\neg C \Rightarrow R$

*Goal*: $R$.

Possible ways of handling this problem are: truth tables (model checking), Hilbert style proving, resolution proving. Truth tables are bore, resolution will reappear below many times. Let us do this by the conventional propositional logic, Hilbert style. We will use reasoning by cases of sort:

$$N, KB \vdash R \text{ and } \neg N, KB \vdash R, \text{ therefore } KB \vdash R.$$

An obvious justification of this rule comes from the Deduction Theorem, axiom 8 $(N \Rightarrow R) \Rightarrow ((\neg N \Rightarrow R) \Rightarrow (N \vee \neg N) \Rightarrow R)$ and the law of excluded middle $N \vee \neg N$.

6. $N$, a hypothesis
7. $H$, by MP, from 1,6
8. $\neg C$, by MP, from 2,7
9. $R$, by MP, from 5,8.

Likewise,

10. $\neg N$, a hypothesis
11. $P$, by MP, from 3, 10
12. $\neg C$, by MO, from 4, 11
13. $R$, by MP, from 5,12.

## 0.25 Question

A cryptarithmetic problem is a simple constraint satisfaction problem where one must assign numerals to letters in order to satisfy a given arithmetic relation, subject to the constraint that each letter be assigned a unique numeral. For example, given the problem

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

the task is to find an assignment $D = 7$, $E = 5$, $Y = 2$, $N = 6$, $R = 8$, $O = 0$, $S = 9$, $M = 1$ where each letter is assigned a unique numeral and the resulting arithmetic relation $9567 + 1085 = 10652$ is satisfied.

In this question you will use a simple backtrack search to solve the following cryptarithmetic problem:

$$\begin{array}{cccc} & & N & O \\ & G & U & N \\ + & & N & O \\ \hline H & U & N & T \end{array}$$

To make the problem easier, you will employ an ordering over variables O, N, T, U, G, H, and an ordering over numerals 9, 8, 7, 6, 5, 4, 3, 2, 1, 0. Also, notice that once O and N are assigned, the assignments to T, U, G, H are all forced (if one is to satisfy the arithmetic constraints). Therefore, to speed up the search, you only have to consider assignments over O and N, determine the values of T, U, G, H (in order), and immediately backtrack if any of the following constraints are violated: (i) two of the variables are assigned the same numeral; (ii) it is impossible to assign a single numeral to a variable and yet satisfy the arithmetic constraints; and (iii) the variable U gets assigned any numeral other than 0. (It turns out that U has to be assigned 0 or there is no way to consistently assign numerals to G and H.)

(a) (3%) Trace through the following tables in order (*in your exam book!*), showing which numerals get assigned before a constraint gets violated, and stopping at the first successful assignment.

| O | N | T | U | G | H |
|---|---|---|---|---|---|
| 9 | 8 | | | | |
| 9 | 7 | | | | |
| 9 | 6 | | | | |
| 9 | 5 | | | | |
| 9 | 4 | | | | |
| 9 | 3 | | | | |
| 9 | 2 | | | | |
| 9 | 1 | | | | |
| 9 | 0 | | | | |

| O | N | T | U | G | H |
|---|---|---|---|---|---|
| 8 | 9 | | | | |
| 8 | 7 | | | | |
| 8 | 6 | | | | |
| 8 | 5 | | | | |
| 8 | 4 | | | | |
| 8 | 3 | | | | |
| 8 | 2 | | | | |
| 8 | 1 | | | | |
| 8 | 0 | | | | |

| O | N | T | U | G | H |
|---|---|---|---|---|---|
| 7 | 9 | | | | |
| 7 | 8 | | | | |
| 7 | 6 | | | | |
| 7 | 5 | | | | |
| 7 | 4 | | | | |
| 7 | 3 | | | | |
| 7 | 2 | | | | |
| 7 | 1 | | | | |
| 7 | 0 | | | | |

(b) (1%) How many partial variable assignments (rows) did you have to consider to find a solution? Using the same variable and numeral ordering, how many rows would have to be considered if you simply enumerated every complete variable assignment until a solution is found?

## 0.26 Question

```
      A  h=5
     / |   \ 1
 5 /  6|     D h=3
 /     |     |   \
B h=2  |     \ 4  \ 2
 |    C h=4   \   |
 \     |        F h=3
 5\   1|      /
   \   |     /4
       G h=1
       /
    1 /
     /
    E h=1
    |
  2 |
    |
 Goal h=0
```

a. Show step by step how A∗ expands nodes. Do not use the pathmax equation. For each node expanded during the search show its f, g, and h values. If a node is reached on multiple paths show its f and g values each time the node is reached, and indicate any change of its parent node.

b. Show the solution path found.

c. Would the result be different if you were to use the pathmax equation? why? would the process of expanding nodes be different if you were to use the pathmax equation?

## 0.27 Question

Determine whether searching backwards from the goal state or forward from the initial state would be preferable when solving the following problems. For each problem specify its initial and goal states. Justify your answer.

a. find a path in a maze;

b. interpret sonar readings underwater to recognize a large submarine from a small one;

c. verify if a person is your cousin by checking that you both have the same common ancestor Paul Revere;

d. verify if you and Bill have a common ancestor no more than 4 generations back.

## 0.28 Question

Suppose you are given an upper bound C to the cost of the optimal solution for a search problem. You decide to modify A$*$ as follows: every time a node n with f(n)>C is generated, the node is immediately discarded.

a. Is your new algorithm guaranteed to find the optimal solution? explain your answer and specify any condition or assumption you make.

b. What happens if you use a non admissible heuristics?