CS 188 — Introduction to AI — **Midterm 1**

Fall 1993 — Stuart Russell

You have 1 hour and 20 minutes. The exam is open-book, open-notes.
You will not necessarily finish all questions, so do your best ones first.
Write your answers in blue books. Hand them all in.

60 points total. Panic not.

1. (15 pts.) Definitions
Provide brief, *precise* definitions of the following:

(a) (3) Accessible environment

(b) (3) Truth table

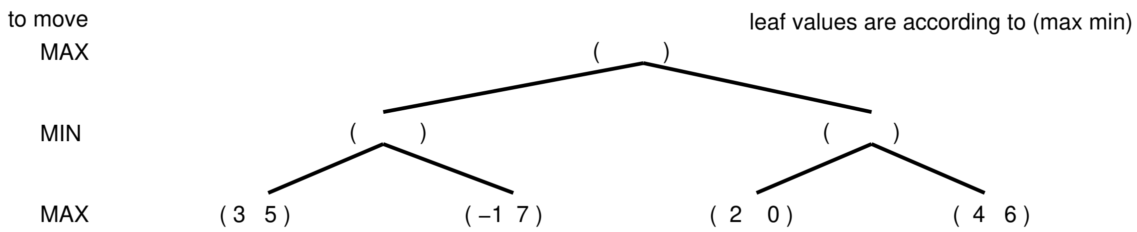
(c) (3) Evaluation function

(d) (3) Complete search algorithm

(e) (3) Sound inference rule

2. (12 pts.) Game-playing
In this question, we will consider the problem of search in a game tree when each player knows the evaluation function used by the other. The first change to the basic program is that `evaluate` will return a list of two values, indicating (say) the likelihood of a win for MAX according to MAX and MIN respectively.

(a) (4) Copy and complete the following game tree by filling in the backed-up value pairs for all remaining nodes. Mark on your tree the move that should be taken at the root.



(b) (4) The following code implements backing-up in a regular minimax algorithm:

```
(defun backed-up-value (side state depth limit)
  (if (= depth limit)
      (evaluate side state)
      (apply (if (oddp depth) #'min #'max)
             (mapcar #'(lambda (s) (backed-up-value (opponent side) s (1+ depth) limit))
                     (successors state)))))
```

Rewrite `backed-up-value` so that it works correctly with the new information. [You may wish to use the functions `the-biggest` and `the-smallest`. `(the-biggest f l)` returns the element of l with the largest value of the function f, eg `(the-biggest #'abs '(-2 4 -7))` = -7.]

(c) (4) (Open-ended) Can alpha-beta pruning be applied in any straightforward way with the two-valued evaluation function? If so, how? If not, why not?

3. (12 pts.) Simple knowledge representation
Translate each of the following English sentences into the language of standard first-order logic, stating the intended interpretation for any predicate, function or constant you use.

(a) (3) "All PCs are computers."

(b) (3) "If someone owns a PC, then there is some computer that they own."

(c) (3) "MaryBeth owns a PC."

(d) (3) "Anyone who owns a computer is a dweeb." [Write this as a first-order Horn clause.]

4. (9 pts.) Logical Inference

Consider the sentences in the previous question.

(a) (1) Does 3(a) logically entail 3(b)? (Yes or No)

(b) (1) Does 3(b) logically entail 3(a)? (Yes or No)

(c) (2) Apply Existential Elimination to 3(c).

(d) (5) Consider a knowledge base containing the sentences obtained in 3(a) 3(d) and 4(c). Show exactly how backward-chaining solves the query "Who is a dweeb?" Draw the proof tree, with unifiers, in the manner shown in chapter 9, p.183.

5. (12 pts.) Situation calculus

Assume that we have logical sentences in a KB describing the distances between adjacent towns in Romania. For example

$$Distance(Arad, Sibiu, 140) \qquad Distance(Sibiu, Fagaras, 99)$$

The state of the agent is given by its location and fuel level. It gets 50km/gallon.

(a) (5) Write axioms to define the $PathDistance$ predicate which describes the length of a path through a given list of cities, for example

$$PathDistance([Arad, Sibiu, Fagaras], 239)$$

[Hint: start with the base case when the path contains only one city. Then do the recursive case. You may use the $[x|l]$ notation defined in chapter 7 on p.128.]

(b) (5) Write a situation calculus axiom defining the positive effects of the action $FollowPath(p)$. Describe both the change in location and the change in fuel level. You must state the appropriate preconditions for following the path successfully, but need not describe what happens if those conditions are not met. You may use the function $Last(l)$ which denotes the last element of a list, as well as any arithmetic functions. [Note: this axiom does not need to be recursive. Just state the result of getting to the end of the path.]

(c) (2) Describe (in words) what sorts of frame axioms you would need, if any, for this world description.