CS 188 Fall 1993 — Introduction to AI — Stuart Russell — Midterm 1 Solutions

1. (15 pts.) Definitions

(a) (3) *Accessible environment*: an environment is accessible to an agent if the agent's percepts provide information on the complete state.

(b) (3) *Truth table*: a truth table for a propositional logic expression shows the truth/falsehood of the expression in all possible assignments of truth/falsehood to the propositional symbols in the expression.

(c) (3) *Evaluation function*: a function that estimates the utility of a state in a game tree.

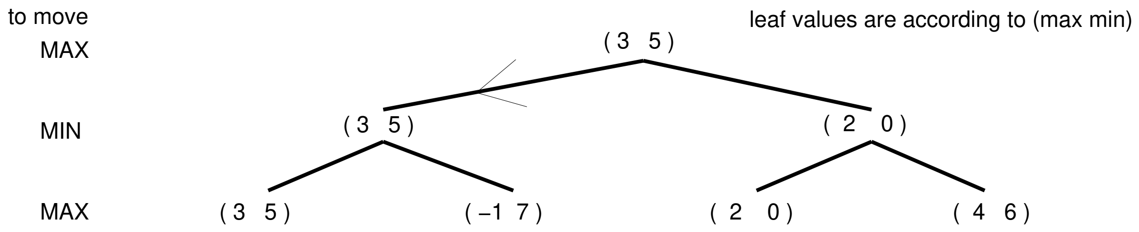
(d) (3) *Complete search algorithm*: an algorithm guaranteed to find a solution if one exists at a finite depth.

(e) (3) *Sound inference rule*: a pattern of inference that is guaranteed to generate true conclusions if given true premises.

2. (12 pts.) Game-playing

The main thing here is to understand who is making the decision at each node. If Max is deciding, the first value will be maximized. If Min is deciding, the second value will be minimized.

(a) (4)



(b) (4) We just need to replace `#'max` and `#'max` by the appropriate decision function:

```
(defun backed-up-value (side state depth limit)
  (if (= depth limit)
      (evaluate side state)
      (apply (if (oddp depth)
                 #'(lambda (pairs) (the-smallest #'second pairs))
                 #'(lambda (pairs) (the-biggest #'first pairs)))
             (mapcar #'(lambda (s) (backed-up-value (opponent side) s (1+ depth) limit))
                     (successors state)))))
```

(c) (4) Looking at the tree above, we see that if we were just using the values according to Max, the rightmost leaf would be pruned because 2 is less than 3. But because Min is actually going to make the decision, we have to check it. For example, it might have turned out to be (4 -1), in which case Min would choose it, giving Max a value of 4 and changing the best move. Thus, at least for two-ply trees, there appears to be no way to prune because the two sets of values are uncorrelated.

3. (12 pts.) Simple knowledge representation

$PC(x)$ means x is a PC. $Computer(x)$ means x is a computer. $Owns(x,y)$ means x owns y. $MaryBeth$ means MaryBeth. $Dweeb(x)$ means x is a dweeb. We won't worry about person predicates.

(a) (3) "All PCs are computers." $\forall x\ PC(x) \Rightarrow Computer(x)$

(b) (3) "If someone owns a PC, then there is some computer that they own."
$\forall x, y\ Owns(x,y) \land PC(y) \Rightarrow \exists z Owns(x,z) \land Computer(z)$

(c) (3) “MaryBeth owns a PC.” $\exists x Owns(MaryBeth, x) \wedge PC(x)$

(d) (3) “Anyone who owns a computer is a dweeb.” $\forall x, y\ Owns(x, y) \wedge Computer(y) \Rightarrow Dweeb(x)$
It is also possible to write $forall x\ (\exists y\ Owns(x, y) \wedge Computer(y)) \Rightarrow Dweeb(x)$
but this wouldn't be a Horn clause since the LHS is not a conjunction of atomic sentences.

4. (9 pts.) Logical Inference
Consider the sentences in the previous question.

(a) (1) Yes

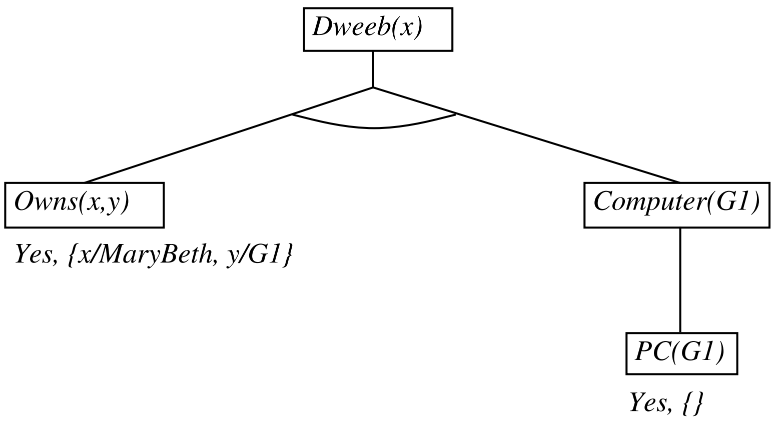
(b) (1) No

(c) (2) $Owns(MaryBeth, G1) \wedge PC(G1)$

(d) (5) The knowledge base contains

$$\begin{array}{l}
\forall x\ PC(x) \Rightarrow Computer(x) \\
\forall x, y\ Owns(x, y) \wedge Computer(y) \Rightarrow Dweeb(x) \\
Owns(MaryBeth, G1) \\
PC(G1)
\end{array}$$

The inference process generates the following proof tree:



5. (12 pts.) Situation calculus

(a) (5)

$$\begin{array}{l}
\forall x\ PathDistance([x], 0) \\
\forall x, y, l, d_1, d_2\ Distance(x, y, d_1) \wedge PathDistance([y|l], d_2) \Rightarrow PathDistance([x, y|l], d_1 + d_2)
\end{array}$$

(b) (5) The preconditions are that the agent must be at the start of the path and have enough gas to get to the end of the path. The effects are that the agent is at the end of the path and the fuel level drops by the amount of gas needed for the path distance:

$$\begin{array}{l}
\forall x, l, y, f, d, p, s At(x, s) \wedge FuelLevel(f, s) \wedge PathDistance(p, d) \wedge f > (d/50) \wedge y = Last(p) \\
\qquad \Rightarrow At(y, Result(FollowPath(p), s)) \wedge FuelLevel(f - d/50, Result(FollowPath(p), s))
\end{array}$$

(c) (2) Since the only action changes both situation-dependent predicates, there is no need for a frame axiom to say what doesn't change. However, if you feel the need to add new predicates such as $CahOnHand$ and say that the trip doesn't change those, then that's OK too.