CS 188 Fall 1997 | Introduction to AI Stuart Russell | Midterm

You have 1 hour and 20 minutes. The exam is open-book, open-notes.
100 points total.

You will not necessarily finish all questions, so do your best ones first.
Write your answers in blue books. Check you haven't skipped any by accident. Hand them all in. Panic not.

1. **(18 pts.) True/False**
   Decide if each of the following is true or false. If you are not sure you may wish to provide a *brief* explanation to follow your answer.

   (a) (3) It is possible to build a knowledge-based agent that is a pure reflex agent.

   (b) (3) Breadth-first search is complete if the state space has infinite depth but finite branching factor.

   (c) (3) Assume that a king can move one square in any direction on a chessboard (8 directions in all). Manhattan distance is then an admissible heuristic for the problem of moving the king from square A to square B.

   (d) (3) It is possible to write an *exact* evaluation function for chess.

   (e) (3) $(P \wedge \neg R) \Rightarrow (Q \Rightarrow R)$ can be converted into a Horn clause.

   (f) (3) $[\forall x\ P(x)] \vee [\forall x\ \neg P(x)]$ is a valid sentence.

2. **(12 pts.) Knowledge representation and inference**
   Translate each of the following English sentences into the language of standard first-order logic, including quantifiers. Use the predicates $French(x)$, $Chilean(x)$, $Wine(x)$, $>$, and the functions $Price(x)$ and $Quality(x)$.

   (a) (6) "All French wines cost more than all Chilean wines."

   (b) (6) "The best Chilean wine is better than some French wines."

3. **(18 pts.) Logical Inference**

   (a) (3) Explain how to use resolution to show that a given sentence α in first-order logic is true in all models.

   (b) (4) Can resolution be used in all cases to show that a first-order sentence α is true in at least one model?

   (c) (8) Show that the sentences
   $$\forall x\ [\forall y\ P(x,y)] \Rightarrow Q(x)$$
   and
   $$\forall x\ \exists y\ [P(x,y) \Rightarrow Q(x)]$$
   are logically equivalent by converting each into CNF. Show each step in the process.

   (d) (3) Give interpretations for the predicates P and Q such that the sentence in part (c) is (more or less) true *in the real world*.

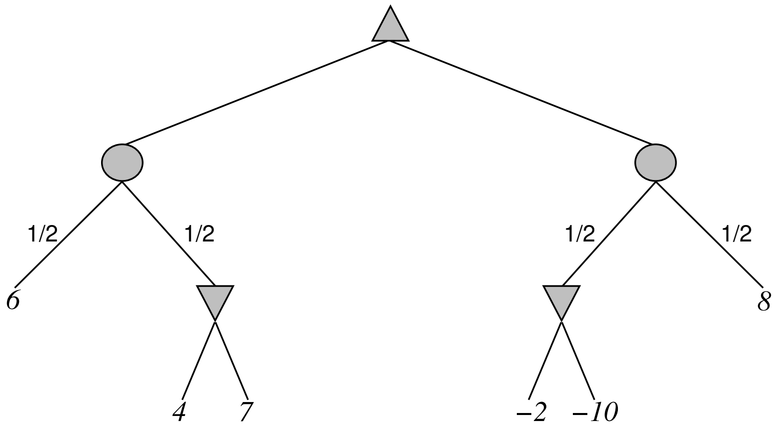
4. **(10 pts.) Situation calculus**
   Consider the following STRIPS operator, which describes the assignment operation from register y into register x (i.e., $x \leftarrow y$).

   $$Op(\text{ACTION}{:}Assign(x,y), \text{PRECOND}{:}Value(x,v_1) \wedge Value(y,v_2) \wedge Register(x) \wedge Register(y),$$
   $$\text{EFFECT}{:}Value(x,v_2) \wedge \neg Value(x,v_1))$$

   translate this information into situation calculus, EITHER as successor state axiom(s) OR as effect axioms plus frame axioms. You can assume that $Value$ is the only situation-dependent predicate.

5. (22 pts.) Game-playing

(a) (6) The following diagram shows a game tree with chance nodes and evaluations at the leaves. Copy the diagram and fill in the values of all internal nodes including the root, and indicate the best move at the root.

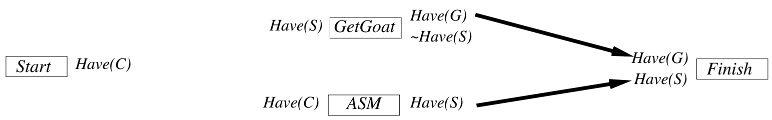


(b) (10) In this question we will consider a version of alpha–beta for game trees with chance nodes. Pruning in such trees is possible if there are known bounds on the values of leaf nodes. Suppose the tree above is evaluated in a left-to-right order, and that all leaf values fall in the range [+10,–10]. Explain in words how these bounds might enable some leaf nodes to be pruned. On your diagram, circle those leaves that need not be evaluated.

(c) (6) Consider a deterministic game where the two players are *not necessarily competing*—each simply tries to maximize his or her own utility. For any given terminal node, the utilities for each player are not necessarily related. (This is not the case in "standard" games, where, for example, if one player wins the other must lose.) Is it still possible to prune any nodes using (some version of) alpha–beta? If so, show a simple example tree that could be pruned; if not, explain precisely why not. [Hint: in this type of game, there can be leaves that make both players happy.]

6. (20 pts.) Planning

Examine the following incomplete partial plan carefully. The goal is to have a sheep and a goat. The *GetGoat* operator exchanges a sheep for a goat. The *ASM* (Automated Sheep Machine) operator yields a sheep if you have an ASM card. (For obvious reasons, one can only carry one sheep in one's wallet, so trying to get more than one sheep is a fruitless operation.)



(a) (3) Which conditions are open?

(b) (4) List all clobberings in the plan, giving both the clobbering step and the link being clobbered.

(c) (4) State which temporal orderings must be added to eliminate the clobberings.

(d) (6) Add causal link(s), step(s), and temporal ordering(s) to the plan in order to complete it. Draw the resulting plan.

(e) (3) Suppose you are executing the resulting plan. What sentences must be true just prior to executing the *GetGoat* operator in order for the plan to reach the goal?