CS 188 Introduction to AI
Spring 1996 Stuart Russell

Midterm

You have 1 hour and 20 minutes. The exam is open-book, open-notes.
80 points total (1 point per minute).

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 some explanation to follow your answer.

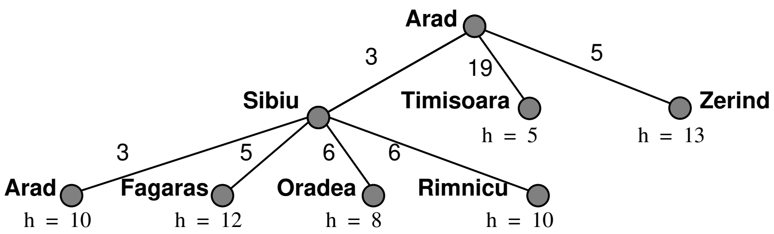
(a) (3) Breadth-first is an optimal search algorithm.
(b) (3) Truth tables can be used to establish the truth or falsehood of any propositional sentence.
(c) (3) Forward chaining is complete for first-order logic.
(d) (3) $\exists x \forall y \; x = y$ is a satisfiable sentence.
(e) (3) Minimax and alpha-beta can sometimes return different results.
(f) (3) Simple reflex agents cope well with inaccessible environments.

2. (14 pts.) Search

(a) (4) In a map-colouring problem, the aim is to colour a map using a given set of colours so that no two adjacent countries are the same colour. Give a precise formulation of map-colouring as a search problem.
(b) (4) Provide a rigorous critique of each step of the following argument, which appeared in a recent submission to the European Conference on AI:

> "Given two admissible heuristics h_1 and h_2 where $h_1(n) \geq h_2(n)$ for all nodes n, it is obvious that A* using h_1 will be more efficient than A* using h_2. Now suppose I am given an admissible heuristic h_2. If one can find a constant c such that the heuristic $h_1(n) = h_2(n) + c$ is still admissible, then searching with h_1 is better than searching with h_2."

(c) (6) The following diagram shows a partially expanded search tree. Each arc is labelled with the corresponding step cost, and the leaves are labelled with the h value.



i. Which leaf will be expanded next by a greedy search?
ii. Which leaf will be expanded next by a uniform-cost search?
iii. Which leaf will be expanded next by an A* search?

3. (8 pts.) Game-playing
Draw the *smallest possible* game tree on which alpha-beta will prune at least one leaf node. Make sure to label the leaves with values, and circle the leaf (or leaves) that will be pruned.

4. (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) "No one at UCB listens to KBLX."

(b) (3) "No one is at both UCB and Stanford."

(c) (3) "If Joe knows someone at Stanford then Joe knows someone who is not at UCB."

(d) (3) "Everyone at Stanford listens to the same radio stations."

5. (13 pts.) Logical Inference

(a) (3) Give a unifier for each of the following pairs of sentences, if possible, or write "None" if not.

i.	$Between(1, x, 2)$	$Between(y, F(y), 2)$
ii.	$> (x, y)$	$< (y, x)$
iii.	$Related(x, Father(x))$	$Related(Father(y), y)$

(b) (2) Let S be the sentence $\exists x\ 1 + x = 1$. Let S' be the result of applying Existential Elimination to S. Write down S'.

(c) (2) Suppose that S and S' are now part of our knowledge base, and that we now apply Existential Elimination to S again to obtain S''. Write down S''.

(d) (4) If Existential Elimination is a sound rule, then both S' and S'' follow from S. Yet S' and S'' taken together seem to say there are two numbers x such that $1 + x = 1$, which seems to be different from the intention of the original sentence. Can you resolve this apparent paradox?

(e) (2) Let C_1 be the clause $\neg At(Father(x), Stanford) \vee At(x, Stanford)$, and let C_2 be the clause $\neg At(y, Stanford) \vee Owns(y, BMW) \vee \neg Happy(y)$. Write down the result of applying resolution to C_1 and C_2.

6. (15 pts.) Planning

In Stanford's amazing new VRISC machine, there are only three registers, called A, B, and C. Initially, they contain 1, 2, and 3 respectively. There is only one instruction, called $Assign(x, y)$, which copies the value contained in register y into register x. The only predicate we need to describe the situation is $Value(x, v)$, which says that register x contains value v. We would like to use a partial-order planning algorithm to construct a plan to switch the contents of A and B.

(a) (3) Using the pictorial notation for STRIPS operators, show the *Assign* operator with its precondition(s) and effect(s).

(b) (3) Leaving about 3 inches of empty space in the middle, draw the initial plan for this problem, again using the standard pictorial notation.

(c) (3) Now the planner decides to achieve the goal condition that B should contain a 1. Add an *Assign* step to your diagram so that this condition is achieved. *Make sure to bind variables as appropriate.*

(d) (3) Now suppose that the planner decides to achieve the precondition of this *Assign* step by connecting it to the *Start* step. Update your diagram to reflect this, marking any changes in bindings.

(e) (3) Now suppose the planner decides to achieve the other goal condition, namely that A should contain a 2. Explain in words (with changes to your diagram if that makes it easier to explain) the sequence of events that occur when it adds a new *Assign* step to achieve this, up to the point where the addition of the step with all its ramifications is complete.