CS472
Foundations of Artificial Intelligence

Final Exam

December 14, 2004
12-2:30pm

Name:

(Q exam takers should write their **Number** instead!!!)

Netid:

(Q exam takers should **leave this blank**!!!)

Instructions: You have 2.5 hours to complete this exam. The exam is a closed-book, closed-notes exam. Suggestion: 15 minutes per question = 2 hrs. The "machine learning I" question may take more like 20 minutes; but the "version space" question should take about 10.

#	description	score	max score
1	Logic and Knowledge-based systems	____	15
2	CSPs	----	15
3	State space search	____	15
4	Planning	____	15
5	Machine learning I	____	20
6	Game playing	____	15
7	Decision tree learning	____	15
8	Version spaces	____	10
Total score:		________ /	120

Logic and Knowledge-based Systems (15 points total)

1. Consider the following propositions:

   - I: Ivana is an apprentice.
   - J: Jennifer is an apprentice.
   - K: Kevin is an apprentice.
   - L: keLly is an apprentice.

   Using the letters given, express each of the following statements in *conjunctive normal form*, suitable for resolution theorem proving.

   (a) (2 pts) Ivana or Jennifer is an apprentice.

   (b) (2 pts) If keLly is an apprentice, then Jennifer is an apprentice.

   (c) (3 pts) If Kevin is an apprentice, then Ivana and keLly are not apprentices.

   Consider doing the same for the following statement: *At least two of the four people are apprentices.* (Do not spend long on this.)

   (a) (3 pts) How many clauses are needed to express a statement of the form *at least M out of the N people are apprentices*? Express your answer in terms of M and N.

2. Assume that you were asked to implement a proof procedure that can answer queries for any reasonably-sized (e.g. thousands of sentences) first-order logic knowledge base described in conjunctive normal form.

   - (2 pts) Would you suggest a proof procedure that uses forward- or backward-chaining? Briefly explain your answer.

   - (3 pts) Would you suggest a proof procedure that relies on (a) a very large set of logical equivalences to make inferences, (b) a single rule of inference, or (c) somewhere in between? Briefly explain your answer.

Constraint Satisfaction Problems (15 points total)

Consider the n-queens problem, which requires the placement of n queens on an nxn chess board such that no queen is on the same row, column, or diagonal as any other.

1. (9 pts) Formalize the **4-queens problem** as a constraint satisfaction problem. (You only need to exhaustively list the constraints for one of the types of constraints that you include in the CSP. Any additional types of constraints can be explained briefly in words rather than equations.)

2. (4 pts) Show the constraint graph for your CSP formalization from part 1.

3. (2 pts) Name the *uninformed* search algorithm that is typically employed to solve CSPs?

State Space Search: (15 points total)

1. (3 pts) Under what conditions does A* search produce the optimal solution?

2. (3 pts) Give three reasons why hill-climbing strategies might fail to find a usable solution.

3. (2 pts) An extension to hill climbing called *local beam search* involves retaining the k best choices in each iteration of the search.

   What value of k makes local beam search into the standard hill-climbing search?

   What search strategy expands nodes in the same order as a local beam search with $k = \infty$?

4. (4 pts) Explain iterative-deepening-A* search.

5. (3 pts) Why would one prefer to use IDA* over A*? Why would one prefer to use A* over IDA*?

Planning (and knowledge representation): (15 points total)

Consider the following first-order situation-calculus formulation of the monkey and bananas problem. The world contains a monkey, a box, and a bunch of bananas hanging from the ceiling; the monkey and box are initially on the floor. We'll use the notation *holds(?p, ?s)* to indicate proposition *?p* is true in situation *?s* (where the leading ? indicates a variable); and *result(?o, ?s)* indicates the new situation that results after applying operator *?o* in prior situation *?s*. The initial situation is denoted s0, and the three distinct locations in this world are labeled loc0, loc1, and loc2. The first six axioms formally describe the initial state of the world.

holds (at (monkey, loc0), s0)	*holds (on (monkey, floor), s0)*
holds (at (box, loc1), s0)	*holds (on (box, floor), s0)*
holds (at (bananas, loc2), s0)	*holds (on (bananas, ceiling), s0)*

The next four axioms describe the operators available to change the state of the world: *MOVE, PUSH, CLIMB, AND GRAB.*

holds(at(monkey, ?l1), ?s) $\wedge$ *connected(?l1,?l2)*
$\rightarrow$ *holds(at(monkey, ?l2), result(MOVE(monkey, ?l1, ?l2), ?s))*

holds(at(monkey, ?l1), ?s) $\wedge$ *holds(at(?x, ?l1), ?s)* $\wedge$ *differs(monkey,?x)* $\wedge$ *connected(?l1,?l2)*
$\rightarrow$ *holds(at(monkey, ?l2), result(PUSH(monkey, ?x, ?l1, ?l2), ?s))* $\wedge$
holds(at(?x, ?l2), result(PUSH(monkey, ?x, ?l1, ?l2), ?s))

holds(at(box, ?l1), ?s) $\wedge$ *holds(at(monkey, ?l1),?s)* $\wedge$ *holds(on(monkey, floor),?s)*
$\rightarrow$ *holds(on(monkey,ceiling), result(CLIMB(monkey, box, ?l), ?s))*

holds(on(monkey, ?x), ?s) $\wedge$ *holds(at(monkey,?l),?s)* $\wedge$
holds(on(bananas, ?x), ?s) $\wedge$ *holds(at(bananas, ?l),?s)*
$\rightarrow$ *holds(has(monkey,bananas), result(GRAB(monkey,bananas, ?l, ?x), ?s))*

Note that the special predicates *differs* and *connected* are situationless. You can assume *differs* is appropriately defined and that all locations are appropriately interconnected; thus, *connected (?l1, ?l2)* is true for all *?l1, ?l2* when *differs (?l1,?l2)* is true, i.e., a given location is never connected to itself.

1. (5 pts) We would like to use a theorem prover (or any deductive inference system) to derive plans with this set of axioms. Unfortunately, there is something lacking in our "domain theory," i.e. in our formulation of the monkeys and bananas world. Provide the common name for the problem that we will encounter and describe it briefly.

2. (5 pts) Explain how the domain theory might be extended (you need not actually do it) in a plausible way so that it can be shown that the monkey can get the bananas. What is unsatsifactory about the resulting set of axioms?

3. (5 pts) Are the planners from parts 1 and 2 *partial-order* planners? Briefly explain your answer.

Machine learning I: (20 points total)

1. (5 pts) Draw a perceptron that represents the **majority function** of n binary inputs (i.e. the perceptron should output a 1 if at least half the inputs are 1's and should output a 0 otherwise). Be sure to specify what kind of units you are using.

2. (6 pts) Construct by hand a neural network that computes the **XOR function** of two inputs. Be sure to specify what kind of units you are using.

3. Suppose that a training set contains only a single example, repeated 100 times. In 80 of the 100 cases, the single output value is 1; in the other 20, it is 0.

   - (3 pts) What will a **neural network** predict for this example, assuming that it has been trained using the backpropagation algorithm on the 100-example data set and reaches a global optimum?

   - (3 pts) What will a **decision tree** (trained using ID3) predict for this example after training on the 100-example training set? Be sure to state any assumptions that you make w.r.t. handling non-homogeneous leaves.

   - (3 pts) What will a **1-nearest neighbor** algorithm predict for this example after training on the 100-example training set? Be sure to state any assumptions that you make w.r.t. handling ties.

Game playing: (15 points total)

There's a simple game you can play with silver dollars. (A silver dollar is a coin worth \$1.) There are five (5) silver dollars on the table and you and your opponent take turns picking up 1, 2, or 3 coins until none is left. You get to keep each silver dollar you pick up. *But*, if you pick up the last coin, you have to pay \$2 to your opponent. The object of the game is to finish with the most money. Consider building a game tree to solve the above problem where each state in the game tree represents the number of coins left on the table.

1. (3 pts) What utility function should you use to evaluate terminal nodes?

2. (6 pts) Show the (complete) resulting minimax tree for this problem.

3. Your opponent courteously offers to let you go first.

   - (3 pts) If you accept, what is your first move? How much will you earn?

   - (3 pts) Should you accept? Briefly explain your answer.

Decision Trees: (15 points total)

	User Action	Author	Thread	Length	Where Read
1	skips	known	new	long	home
2	reads	unknown	new	short	school
3	skips	unknown	follow-up	long	school
4	skips	known	follow-up	long	home
5	reads	known	new	short	home
6	skips	known	follow-up	long	school
7	skips	unknown	follow-up	short	school
8	reads	unknown	new	short	school
9	skips	known	follow-up	long	home
10	skips	known	new	long	school
11	skips	unknown	follow-up	short	home
12	skips	known	new	long	school
13	reads	known	follow-up	short	home
14	reads	known	new	short	school
15	reads	known	new	short	home
16	reads	known	follow-up	short	school
17	reads	known	new	short	home
18	reads	unknown	new	short	school

The data set above was gathered by a web-based agent that observed a user deciding whether to *skip* or *read* a newsgroup article depending on whether the author was known or not, whether the article started a new thread or was a follow-up message, the length of the article, and whether it was read at home or at school.

1. (10 pts) Based on the above data, list the features in order of decreasing information gain (i.e. the first feature listed has the higheste information gain and would therefore become the root of a decision tree created by an ID3-style decision tree induction system). You do not need to show any calculations.

2. (5 pts) Assume now that the decision tree learner selects the first feature in the list that you created in part 1 as the root node of the tree. To continue building the tree, the decision tree learner would be called recursively to build the rest of the tree. List the training instances that would be included in the next recursive call associated with each branch of the root node.

Version Spaces: (10 points total)

This is a question about version spaces and Mitchell's candidate-elimination algorithm. Suppose the instance space consists of points on the x-y plane having integer-valued coordinates, and the hypothesis space consists of all rectangles in the plane having corners with integer coordinates. (Rectangles that are unbounded in one or both dimensions are also in the hypothesis space.) An instance is covered by a hypothesis if the instance is a point in, or on, the boundary of the corresponding rectangle.

Show the version space after each of the training instances below has been processed by the candidate-elimination (i.e. version space) algorithm. Draw each version space as a collection of rectangles in the x-y plane. (If any rectangles are not completely bounded, be sure to indicate this in some way in your picture.) Indicate the training points and label the hypotheses as to their membership in the G or S sets.

1. positive instance (-1, 3)

2. positive instance (1, 1)

3. negative instance (1, 5)