CS472
Foundations of Artificial Intelligence

Prelim II
November 22, 2003

Name:

Netid:

Instructions: You have 50 minutes to complete this exam. The exam is a closed-book exam. There are 4 sets of questions.

#	topics	score		max score
1	reinforcement learning	____	/	25
2	version spaces	____	/	20
3	FOL and the resolution algorithm	____	/	20
4	decision trees and neural networks	____	/	35
	Total score:	______	/	100

1 Reinforcement Learning (25 points) Consider reinforcement learning in a known, accessible environment using the methods below:

1. **direct utility estimation** (also known as **naive updating** or **LMS updating**)
2. **dynamic programming**
3. **temporal difference learning**

1. (11 pts) Show the state utility update equation, U, for method (1), **direct utility estimation**. Be sure to define all symbols.

2. (8 pts) Show the state utility update equation(s), U, for method (2), **dynamic programming**. Be sure to define all symbols.

3. (6 pts) Which of the **three** methods above do **not** require a model of the environment?

2 Version Spaces (20 points) Assume that you want to be able to predict whether or not a student will do well on a prelim. In addition, assume that you have determined that success on a prelim depends on three factors: the *professor* giving the test, what you eat for *breakfast* on the day of the prelim, and the amount of *sleep* that you get the night before the prelim. Further assume that each of these features has a fixed set of possible values:

professor: {cardie, pingali, schneider}

breakfast: {cheerios, pop-tarts, bagel}

sleep: {lots, little, 8-hours}

Apply the *version space algorithm* (i.e. the candidate elimination algorithm) to the following examples to determine the concept description for a *successful prelim*. Each example is labeled either **positive** (i.e. you do well on the prelim) or **negative** (i.e. you do poorly on the prelim). Show the resulting *S* and *G* sets after each example:

1. **positive:** *<schneider,cheerios,8-hours>*
2. **negative:** *<cardie,bagel,lots>*
3. **positive:** *<pingali,bagel,8-hours>*

1. (4 pts) After example 1:

2. (5 pts) After example 2:

3. (5 pts) After example 3:

4. (2 pts) Based on your work, what advice would you give to a student taking a prelim?

5. (4 pts) List **two** critical inadequacies of the version space algorithm.

3 First-Order Logic and the Resolution Proof Procedure (25 points)

If a course is easy, some students are happy. If a course has a prelim, no students are happy.

1. (6 pts) Represent the above statements using *first-order logic*. Some useful predicates will be: *easy*, *happy*, *has-prelim*, *student*.

2. (9 pts) Convert the first-order logic statements above into a knowledge base in conjunctive normal form.

3. (10 pts) Given the above knowledge base, prove using *resolution by refutation* that, *if a course has a prelim, the course isn't easy*. (A refutation proof is a proof by contradiction. To obtain full credit, all steps must be labeled so that it's clear what's being resolved with what.)

   To help you out, you can assume that the **negation** of the statement to be proved results in the following sentences when converted to conjunctive normal form (where S2 is a skolem constant):

   (3a) has-prelim (S2)
   (3b) easy (S2)

4 Decision Trees and Neural Networks (30 points)

1. (5 pts) Consider the learning problem from the Version Space question and the following examples:

   (a) **positive:** <*schneider,cheerios,8-hours*>

   (b) **negative:** <*cardie,bagel,lots*>

   (c) **positive:** <*pingali,bagel,little*>

   Explain briefly the decision tree that would result if these examples were presented to a decision tree induction system like ID3. (No equations should be necessary.)

2. (5 pts) (True / False) Decision trees produced by the ID3 algorithm never test the same attribute twice along one path. (Briefly explain your answer.)

3. Consider a two-layer feedforward artificial neural network with two inputs *a* and *b*, one hidden unit *c*, and one output unit *d*. This network has five weights (w_{ac},w_{bc},w_{0c},w_{cd},w_{0d}), where w_{0x} represents the threshold weight for unit x. Assume that these **weights** are initialized to (.1, .1, .1, .1, .1). Assume a **learning rate** of 0.3. Also assume a **sigmoid** threshold function f.

   Use the following approximate values for f where necessary below:

x	$f(x)$
$-5.0 \leq x < -2.5$	0.001
$-2.5 \leq x < -0.05$	0.20
$-0.05 \leq x < 0$	0.49
0	0.5
$0 < x \leq 0.05$	0.51
$0.05 < x \leq 2.5$	0.80
$2.5 < x \leq 5.0$	0.999

   Consider the following **training example** for the network described above: $a = 1$, $b = 0$, $d = 1$.

   (a) (6 pts) Show the **output** for each node during the feedforward pass.

   (b) (9 pts) Show the **error** for each node as it would be computed using the Backpropagation algorithm.

   (c) (5 pts) What is the primary goal of the backpropagation algorithm?