Introduction to Artificial Intelligence
4005-750-01, 20112 (Winter 2011-2012)

Final Examination, March 2, 2012
Instructor: Richard Zanibbi, Duration: 120 Minutes

Name: _____

Instructions

- **The exam questions are worth a total of 100 points.**
- Place any coats or bags at the front of the exam room.
- Students are permitted one letter-sized sheet of paper with notes on **one** side in the exam room.
- You may use pencil or pen, and write on the backs of pages in the booklets.
- If you require clarification of a question or an additional booklet, please raise your hand.
- **Please close the door behind you quietly if you leave before the end of the examination.**

Questions

1. **True/False (6 points)**
   You may provide a brief explanation with your answers for partial credit.
   (a) (T / F) Resolution theorem proving is more powerful than forward or backward chaining in propositional logic, because it is complete.
   (b) (T / F) Alpha-beta pruning allows the minimax action for a game tree to be computed more quickly than with the 'plain' minimax algorithm, but may result in a suboptimal action being selected.
   (c) (T / F) Backpropogation for neural networks is a form of incremental search
   (d) (T / F) In practice, the WalkSAT algorithm can sometimes determine whether a propositional sentence is satisfiable more quickly than the DPLL algorithm, even though it is a local search that includes random transitions.
   (e) (T / F) Propositional logic represents the world as a series of propositions without internal structure, whereas predicate logic represents the world using a set of objects, for which logical predicates are used to assert the truth/falsity of relationships between objects in the domain.
   (f) (T / F) The perceptron learning rule is guaranteed to converge to a hypothesis with zero training error for data in two or fewer dimensions.

2. **Definitions (12 points: 6 points each)**
   Define **two** of the following.
   (a) Model checking
   (b) Entailment
   (c) A* search
   (d) Inductive (machine) learning
   (e) Markov Chain Monte Carlo sampling for Bayesian Networks

3. **Probability and Bayesian Networks (16 points)**

Below is a joint probability distribution for three binary variables (X, Y, and Z).

	z		$\neg z$	
	y	$\neg y$	y	$\neg y$
x	1/8	3/16	1/8	3/16
$\neg x$	1/8	1/16	1/16	1/8

(a) (4 points) Identify which pairs of variables are (absolutely) *independent* of one another in the table above, and explain why they are independent.

(b) (8 points) Draw the following Bayesian network with 4 binary variables: one variable is independent of the other 3 variables, and 2 of the remaining variables are independent given the final variable.

(c) (4 points) For the last question, how many *independent* entries are there in the full joint probability distribution table? How many are there in the Bayesian network that you drew?

4. **Logic (16 points)**

   (a) (8 points) Trace the execution of the Prolog program below for the query: `breeze(4,3)`, showing the unification of variables as each rule is applied. You may organize the execution using a tree.

```
breeze(3,2).
not(breeze(2,3)).
pit(4,2).
breeze(X,Y) :- A is X-1, pit(A,Y).
breeze(X,Y) :- A is X+1, pit(A,Y).
breeze(X,Y) :- B is Y-1, pit(X,B).
breeze(X,Y) :- B is Y+1, pit(X,B).
```

(b) (8 points) Convert the following sentences to clausal form, and then prove a (a is true) using resolution (Hint: recall that resolution proofs are proofs by contradiction).

$$b \wedge c \rightarrow a$$
$$b$$
$$d \wedge e \rightarrow c$$
$$e \vee f$$
$$d \wedge \neg f$$

5. **Machine Learning (18 points)**
   (a) (6 points) Are neural networks classifiers, regressors, or both? Explain your answer.
   (b) (8 points) Linear models are used extensively for classification and regression. Suppose that we have a linear function $y = 2x_1 + x_2 + 4$.
      i. (4 points) Show where the function has $y = 0$, and draw the weight vector represented in the expression.
      ii. (4 points) Explain how this function can be used to define a linear classifier known as a *perceptron*.
   (c) (4 points) Which types of data are 1) decision trees and 2) neural networks best suited to modeling?

6. **Decision Trees (16 points)**
   (a) (6 points) Assume that we are given a binary variable W representing whether it will rain or shine on a given day in Rochester. Assume further that we have a probability distribution $\mathbb{P}(W)$ for that variable.
      i. (4 points) Provide a formula showing how to compute the *entropy* for W.
      ii. (2 points) Which distribution of $\mathbb{P}(W)$ will maximize entropy?
   (b) (4 points) A known limitation of the types of decision trees that we studied in class is that they tend to *over-fit* the training data. Explain what is meant by over-fitting, and why this tends to occur using the (naive) decision tree induction algorithm.
   (c) (6 points) Describe or provide pseudo code detailing how the decision tree induction algorithm works.

7. **Neural Networks (16 points)**

Imagine that we are constructing a neural network to classify pictures of dogs, skyscrapers, and coffee mugs.

(a) (4 points) Suppose that using two image features (attributes) results in the classes being *linearly separable* in the input space. Draw a diagram for the smallest neural network with the simplest activation functions that will correctly classify the data.

(b) (8 points) Suppose that we are training a linear regressor (a single node using the sigmoid activation function) with two inputs (x_1, x_2) and corresponding connection weights (w_1, w_2); there is also a connection weight for a bias input, w_0. The values are $(w_0, w_1, w_2) = (-0.5, 2, -1.5)$. The output produced for input (1,1) is 0.5, with a target value (i.e. correct output) of 1.0.

i. (2 points) What is the value of the derivative of the output?

ii. (6 points) Compute the new weight for w_2 obtained using gradient descent. Identify all the terms in your update computation.

(c) (4 points) Suppose we start considering images with more noise (e.g. more variation in lighting, 'dogs' produced by leaving lights on in skyscrapers at night), and the classes are no longer linearly separable in our feature space. To accommodate this, we add a hidden layer to the network.

Explain how to backpropogate errors at the output nodes to one of the hidden nodes during training, so that 'blame' for error at the output is correctly assigned to nodes in the hidden layer.