Homework #7: Chapters 16, 18 - 20

The following exercises are due at the beginning of class on Friday, April 30. Note, this homework is continued on the reverse side of the paper.

1. *[20 pts.]* Consider the following variation of the Wumpus World agent. The agent has determined the probabilities of the outcomes of three different action sequences: A,B and C. The outcomes are defined by two Boolean propositions: *Gold* and *Die*. *Gold* is true if the actions lead to the gold. *Die* is true if the agent loses its life (e.g., by stepping into a pit or encountering the Wumpus). The agent's utility is -100 if the agent dies (regardless if it gets the gold or not), 50 if the agent gets the gold and lives, and 10 if the agent lives but does not get the gold. The probability of the various outcomes are given by the table below:

Action Sequence	**Gold**		**¬Gold**	
	Die	**¬Die**	**Die**	**¬Die**
A	0.6	0.05	0.2	0.15
B	0.05	0.1	0.05	0.8
C	0.25	0.6	0.1	0.05

   What is the expected utility of each action sequence? To maximize the chance for partial credit, be sure to show your work. If the agent follows the principle of maximum expected utility, which action sequence will it choose?

2. *[25 pts.]* Consider the training set in Table 1 on the reverse side of this paper.
   a) *[4 pts.]* Calculate the information requirement for the training set.
   b) *[16 pts.]* Calculate the information gain for each of the four attributes: *type*, *garage*, *bedrooms*, and *bathrooms*. **Hint:** If you don't have a calculator capable of doing base 2 logarithms, you can calculate them using the natural logarithm: $\log_2 x = \ln x / \ln 2$.
   c) *[5 pts.]* Based on your findings in part b), draw a partial decision tree that includes the attribute on which the first test should be performed and its immediate child nodes. You may use a question mark for the attribute of any nodes that cannot be completely classified by the first attribute test.

3. *[40 pts.]* Consider the training set in Table 1 again.
   a) *[8 pts.]* Convert the training set into a set of first-order logic description and classification sentences. Use the predicates *Type(x,t)*, *Garage(x,g)*, *Beds(x,b)* and *Baths(x,b)* in your description sentences and *Q(x)* for your goal predicate. Note that by using a binary predicate for garage (instead of a unary predicate), we are able to restrict our hypothesis space to only include candidate definitions that are positive conjunctive sentences.
   b) *[32 pts.]* Use current best-hypothesis search learning on the training set from problem #1 above. As was demonstrated in class, assume the algorithm is implemented as a depth-first search. The initial hypothesis should either be "True" or "False", and each node in your search tree should show the current hypothesis and how it classifies the current example (i.e., true positive, true negative, false positive or false negative). When a node

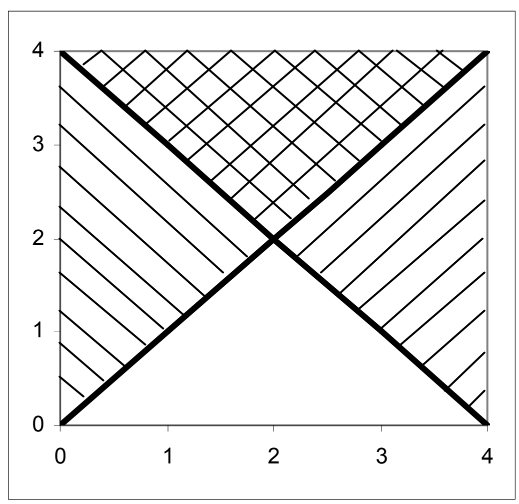
Example	Type	Garage	Bedrooms	Bathrooms	Goal Predicate
X_1	TownHouse	Yes	2	1.5	Yes
X_2	Condo	No	2	2	No
X_3	Apartment	Yes	2	1	No
X_4	TownHouse	Yes	4	2	No
X_5	TownHouse	Yes	3	1.5	Yes
X_6	Condo	Yes	1	1	No
X_7	TownHouse	No	3	1.5	No
X_8	Apartment	Yes	3	1	No

Table 1. Training set for problems #2 and #3.

is expanded, its children are determined by the consistency of its hypothesis. If it is consistent, then it has a single child with the same hypothesis considering the next example. If it is inconsistent, then the children's hypotheses are the immediate specializations (or generalizations) that are consistent with all examples to that point. Assume that the examples are received in the order given and that the hypothesis space only contains hypotheses whose candidate definitions are positive conjunctive sentences. When there is a choice of otherwise equivalent nodes to expand, always choose the one that adds the leftmost remaining condition from the attributes in the table.

4. *[15 pts.]* For this problem you will construct a series of neural networks, building up to a neural network that can recognize a function that is not linearly separable. All of the neural networks will have two inputs: x and y. Both x and y are real numbers and their values are restricted such that $0 \leq x \leq 4$ and $0 \leq y \leq 4$. Use threshold functions for all nodes in your neural networks and assume that the threshold function returns 1 when its input is ≥ 0 and returns 0 otherwise. Do not use a learning algorithm to develop the neural networks in this problem.

   a) Create a single layer feed-forward neural network (a perceptron) to recognize when the input y is greater than or equal to the input x (i.e. $y \geq x$). This corresponds to the area in the graph above the line from the lower left to upper right.
   b) Create a single layer feed-forward neural network (a perceptron) to recognize when the input y is greater than or equal to 4 minus the input x (i.e. $y \geq 4-x$). This corresponds to the area in the graph above the line from the upper left to lower right.
   c) Create a multilayer feed-forward neural network to recognize when y is $\geq$ both x and $4-x$. This corresponds to the double hash marked area in the graph. Obviously, this is not a linearly separable function.



Hints: Approach part c) by considering how to combine your answers from parts a) and b). Test your neural networks thoroughly to be sure that they perform as expected for various combinations of inputs in the range given (especially near boundaries of the function and input ranges)