Name:

Massachusetts Institute of Technology

6.034 Artificial Intelligence

Final Examination **Date:** **6034 Item**

Please indicate the section you attend, or if you do not attend a section, put a tick mark on the last line.

Time	Room	Instructor	TA	Tick
10–11	34-303	Randy Davis	——	
10–11	34-302	Tomás Lozano-Pérez	Gina Levow	
11–12	36-156	Randy Davis	——	
11–12	34-302	Tomás Lozano-Pérez	Gina Levow	
12–1	34-303	Ken Yip	Hyung Chang	
1–2	34-302	Ken Yip	Hyung Chang	
1–2	34-301	Peter Szolovits	Andreas Argiriou	
2–3	34-304	Peter Szolovits	Andreas Argiriou	
2–3	34-301	Polly Pook	Jeff Breidenbach	
3–4	34-303	Polly Pook	Jeff Breidenbach	
—	—	—		

Problem 1	
Problem 2	
Problem 3	
Problem 4	
Problem 5	
Problem 6	
Total	

Special Notes

- On the first two examinations, many students lost many points because of misinterpretations caused by failing to read carefully. **Many even missed warnings printed in bold face.**
- Avoid getting stuck on particular problems. You will probably get a higher grade if you do some work on all problems, rather than better work on fewer problems.

Problem 1 Rules (18 Points)

Please look at the set of rules on the next to final page of the exam. You may want to rip out the last page and have it handy while you work on this problem.

You are also given the collection of assertions shown below and told that on every cycle the rule interpreter:

1. collects all rules that are ready to fire, and
2. selects one to fire based on rule ordering (i.e., the order in which the rules appear in the rule base)

Note that a rule is never selected unless it adds a new assertion.

Assertions:

```
Fuzzy bears live young
Fuzzy does not lay eggs
Fuzzy gives milk
Fuzzy has pointed teeth
Fuzzy has forward-looking eyes
Fuzzy has claws
Fuzzy has whiskers
Fuzzy is tawny colored
Fuzzy has stripes
```

Part A (4 Points)

Simulate the action of the rule interpreter on the assertions given by filling in the table below with the relevant rule names (eg R5) and by indicating any new assertions added to the assertion list. We have supplied more than enough lines in the table.

Step	Ready to Fire	Selected Rule	Assertion Added by Selected Rule
1			
2			
3			
4			
5			
6			

Part B (8 Points)

Now you are given a backward chaining interpreter and told that the system starts with the hypothesis that the animal is a tiger. You are also told that the interpreter selects the rules in the order they appear in the rule set, and works through them depth first. Similarly, you are told that the interpreter works on the clauses in a given rule in the order that the clauses occur in that rule.

You are told as well that the system starts with no initial assertions in the database, but that once a question is answered (or a rule fired), the answer (or inference) is recorded in the database and used by the system (rather than asking the question again).

You are to assume that the user answers questions by looking at the assertion list provided on the first page of this problem: any question corresponding to one of those assertions is to be answered yes; other questions are to be answered no.

As mentioned in class, a backward chaining interpreter produces an AND/OR tree. Simulate the behavior of this interpreter by completing the tree below (we have done the first step for you to show you what we mean) and by showing the order in which the depth-first search backward chaining would ask questions of the user (as before we have given you more than enough lines in the form).

```
                  tiger
                    |

     carnivore  feline  tawny  stripes
```

Questions asked of the user:

1)
2)
3)
4)
5)
6)
7)
8)
9)
10)
11)
12)

Part C (4 Points)

Now you are told the backward chainer starts with the same hypothesis, but proceeds via breadth first search through the AND-OR tree. Show the sequence of questions that this backward chainer would ask of the user.

Questions asked of the user:

1)
2)
3)
4)
5)
6)
7)
8)
9)
10)
11)
12)

Part D (2 Points)

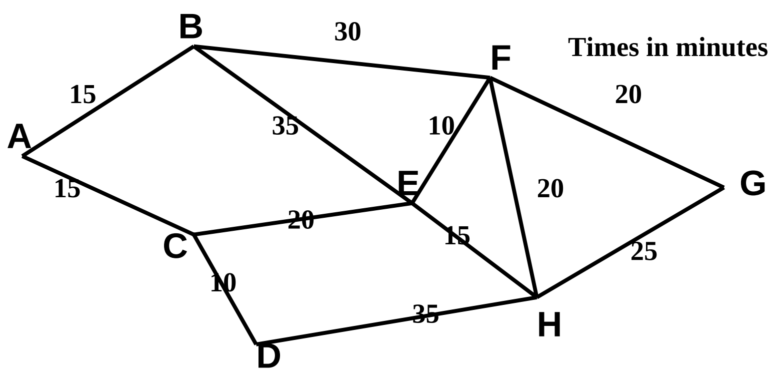
Imagine that you were a *user* rather than the implementor of this system. Review the sequence of questions asked and determine which of these approaches—depth first or breadth first—you think a user would prefer the system to employ. Circle each of the following that support your view:

1. Depth first, because questions tend to stay relevant to particular subproblems, rather than jumping around.
2. Depth first, because the answers will be more reliable.
3. Breadth first, because answers will be produced faster.
4. Breadth first, because questions tend to stay relevant to particular subproblems, rather than jumping around.

Problem 2 Constraints (19 points)

Consider a graph which represents a connected set of train tracks; the nodes in the graph represent stations and the links represent track segments. Each track segment is labeled by the maximum time for a train to traverse the segment. Assume that for safety reasons each track segment can be occupied by only one train at a time. At the stations any number of trains can stop and wait safely.

You are given a set of desired train trips. Each trip is specified by a sequence of stations to be traversed. Each trip has a specified maximum final time, that is, the time by which the train must reach the final station. You have to find a schedule for each trip, that is, the time at which each specified train should leave each station on the trip. Trains can depart from a station anytime after midnight at 5 minute intervals.



Sample trips:
Trip 1: A B E F G   max final time: 1:30pm
Trip 2: G F E C   max final time: 1:20pm

The following questions all concern how to formulate this problem as a Constraint Satisfaction Problem (CSP).

Part A (2 Points)

What are the variables in this CSP? Choose the best one from the list below:

1. all the stations
2. all the track segments
3. the stations for each trip
4. the track segments for each trip
5. all the trips

Part B (2 Points)

Give an example of a variable drawn from the figure above.

Part C (2 Points)

What are the domains of variable values in this CSP? Choose the best one from the list below:

1. all the possible times a train can depart a station
2. all the possible times a train can depart a station, that are earlier than the maximum final time of a trip.
3. all sequences whose length is the number of stations in the trip and whose entries are possible times a train can depart the station.
4. all sequences whose length is the number of track segment in the trip and whose entries are possible times a train can depart the station.
5. all the possible durations of a complete trip.

Part D (2 Points)

Give an example of a variable domain drawn from the figure above.

Part E (4 Points)

What are all the kinds of constraints in this CSP? For each type of constraint, (a) write a sentence that describes the purpose of the constraint and (c) give an example of each kind of constraint drawn from the Figure above. Use a different box for each kind of constraint; we have given you more than enough boxes.

1.

2.

3.

4.

Part F (2 Points)

Draw a constraint graph for the example in the figure above. In this graph, the nodes represent variables and the links represent the constraints. Indicate what type of constraint is indicated by each link (use the number of the constraint box above).

Part G (5 Points)

Now we switch to considering algorithms for GENERAL CSP – not just for the example above. Next to each of the algorithms below, check ALL the properties below that hold for that algorithm. The properties are explained in more detail below the table:

ALGORITHMS:

Algorithm	ALL	ONE	UNL	LGE	INI
Backtracking					
AC-3					
Backtracking + Forward Check					
Backtracking + AC-3					
Min-conflict Hill Climb (with conflict weights)					

We assume that the Min-conflict Hill Climb algorithm continues to run as long as there are conflicts in the current assignment.

PROPERTIES:

1. **ALL**: In principle, guaranteed to find ALL the answers to a CSP.
2. **ONE**: In principle, guaranteed to find AT LEAST ONE answer to a CSP, if any exist.
3. **UNL**: Unlikely by itself to find a unique solution to a CSP.
4. **LGE**: best choice when the size of the variable domains is very large.
5. **INI**: Requires an initial assignment for all the variables.

Problem 3 Frames and Language for Profit (12 Points)

The MIT Office for Grubbing Money (MOGM, pronounced "mug 'em") has hired you to develop a computer system to help keep track of news accounts of major donations to educational organizations. Using the ideas from Chapters 9 and 10 of the text on *frame systems* and *thematic role frames*, you plan to create a representation and a set of programs that will help MOGM identify likely potential donors to MIT.

Assume that George (remember him from 6.001?) has built a low-level language processing facility that retrieves newswire stories from the Internet, parses them into their syntactic elements, picks out the thematic roles and calls your procedure `process` with the following arguments, which correspond to those roles:

(`process` *verb agent thematic-object recipient purpose*)

An example of such a sentence is

> An anonymous foundation gave the Boston-based Center for Political Decency an endowment of $20,000,000 to promote civic debate.

which would result in the following call:

```
(process 'give 'anon-foundation '$20000000 'CPD 'promote-debate)
```

George's program classifies all phrases appropriately into a "world taxonomy" such as the one shown in Figure 1. Thus, `$20000000` *isa* `money`, `anon-foundation` *isa* `foundation`, `CPD` *isa* `non-profit`, and `promote-debate` *isa* `act`. Not all of these are in the figure (which is a **partial** taxonomy) but that does not matter.

Naturally, there are many other sentences parsed by George's program, some of which are relevant to your task and others of which are not. Some examples and their translations follow:

> Mo Vaughn gave Roger Clemens a punch in the nose.

```
(process 'give 'Mo 'punch-in-nose 'Roger nil)
```

> Bill Gates gave Harvard $15,000,000 to support computer science research.

```
(process 'give 'Gates '$15000000 'Harvard 'support-cs)
```

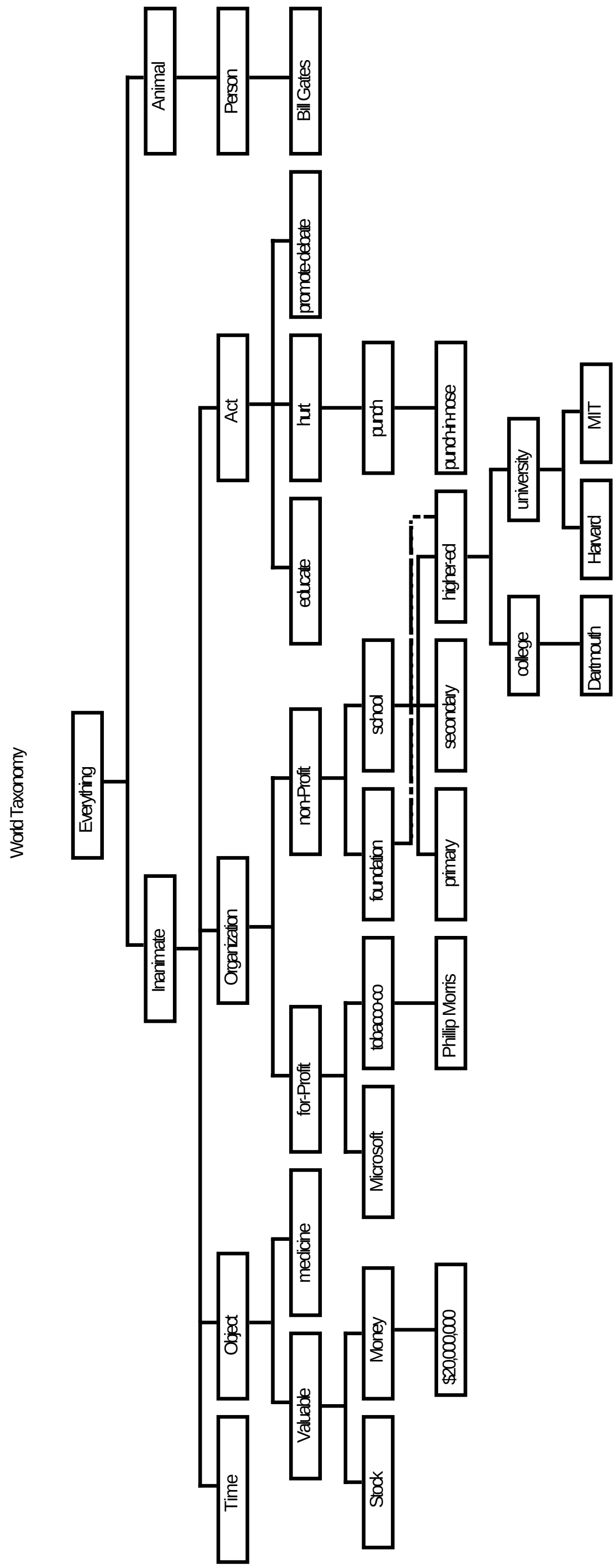


Figure 1: A partial taxonomy of concepts in the world, relevant to understanding sentences about gifts. Lines from a concept to its superconcept represent "a kind of" (*ako*), whereas lines from an instance to its superconcept represent "an instance of" (*isa*). The dotted line is to be considered present only for the last part of this problem.

Part A What is interesting? (4 Points)

To filter all these reports of giving, you need to define a generic procedure `process` that simply returns *true* or *false* depending on whether the thing being given is valuable and the entity to which it is given is a school.

Assume that George has defined each concept in Figure 1 as a CLOS class or (in the case of instances at the leaves) a CLOS instance. Define in the box below `defmethod` forms with appropriate restrictions on the classes of the arguments that will cause calls such as the above ones to `process` to identify whether the thing being given is valuable and the entity to which it is given is a school. A small number of such forms should suffice.

To remind you of the (obscure) syntax of `defmethod`, here is a definition, from problem set 10, of a method for `talk?` meant to apply to people from course VI and any subject:[1]

```
(defmethod talk? ((date VI) (subject T))
           ...)
```

[1] We assume that you can specialize on instances in the same way as on classes, and therefore can write `(defmethod process ((verb phone) (object Bill-Gates)) ...)` even though CLOS would require a more complex syntax because `phone` and `Bill-Gates` are both instances, not classes.

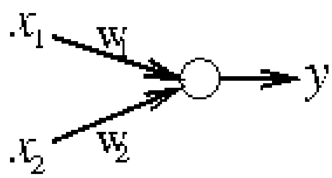
Part B (4 Points)

You observe that a large number of non-profits have gotten into trouble when they accept funds from tobacco companies, even to do worthwhile things. Add a method to the above set that excludes reports of giving by tobacco companies to non-profits from the set of reports for which `process` returns true.

Part C (4 Points)

After running this program for a while, you discover that sometimes institutions of higher education act as foundations. I.e., they are not only given money for educational purposes but may also give money to other schools. Assuming the presence of the dotted *ako* line in Figure 1, give the class precedence list for `MIT`.

Problem 4 Neural Nets (18 Points)



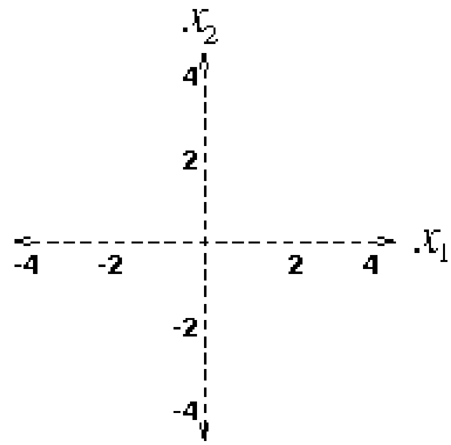


Figure 2: [Used in Part A] Left: A single-layer neural network. Right: The graphical representation of the input sample space.

Part A (5 Points)

Consider the single-layer neural network in Figure 2 (Left), with input $\vec{X} = [x_1, x_2]$ and output y.

1. Specify a threshold activation function and weight vector, $\vec{W} = [w_1, w_2]$ such that the network computes the boolean function:

   if $x_1/2 + x_2 > 0$ then $y = 1$,
   else $y = 0$.

   Answer:

   Threshold activation function:

   $\vec{W} =$

2. On the right side of Figure 2, draw the decision boundary in input space as specified by your neural network and label it **(1)**.

3. Draw a new network similar to the picture in Figure 2 (Left) that uses your same activation function from Question 1 but computes a *different* boolean function:

   if $x_1/2 + x_2 > 2$ then $y = 1$,
   else $y = 0$.

   Answer:

4. Specify the new weight vector $\vec{W}$.
   Answer:
   $\vec{W} =$

5. On the right side of Figure 2, draw the decision boundary computed by your new net and label it **(2)**.

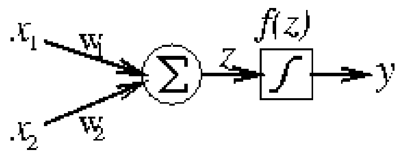


Figure 3: [Used in Part B] A single-layer neural network with a continuous activation function.

Part B (3 Points)

A single-layer neural network is shown in a more expanded form in Figure 3. It has a continuous "squashed-S" activation function f. You wish to train the net using back-propagation. As described in the book, back-propagation changes the values of the weights to improve the performance of the net.

Given one input vector $[x_1, x_2]$ for which the actual output is y and the desired output is d, write the update formula, Δw_i, for changing the weights on the single-layer network using back-propagation. Write your answer in terms of the variables x_1, x_2, y, d, and a learning rate r. Note that the z and $f(z)$ do not appear in the answer.

Answer:

Part C Zipcodes(5 Points)

You work for the post office and want to ease your job of sorting the mail by zipcode. Accordingly you decide to train a neural net to recognize the digits $[0 - 9]$. Fortunately a friend of yours has taken the computer vision course, 6.801, and constructs for you a system that finds the handwritten or printed zipcode on each envelope, separates each character, digitizes it, and outputs a 7x10 matrix in which each pixel (pictorial element in the matrix) is either black or white. Examples of two digits, 0 and 1, are shown in Figure 4. Your network should recognize all 10 digits, [0 - 9].

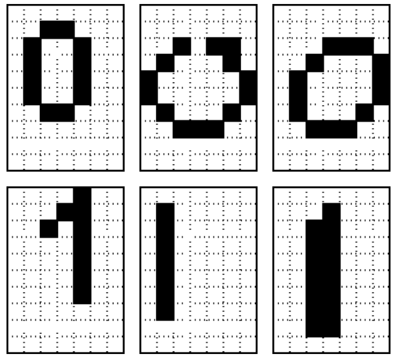


Figure 4: [Used in Part C] Six example outputs from a vision recognition system on hand-written zipcodes for the digits 0 and 1.

1. What is the input vector $\vec{X}$ to your neural net? How many input units does your net have?
   Answer:

2. What values can each x_i take and what do they represent pictorially?
   Answer:

3. What is the output vector $\vec{Y}$ of your neural net? How many output units are there and what values can they take?
   Answer:

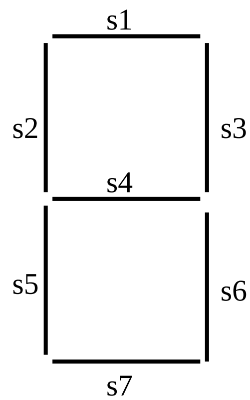
Part D Overfitting(5 Points)

What technique(s) could be used in general to reduce the problem of overfitting a multi-layer neural net? Circle *all* that apply.

1. Increase the number of nodes in each hidden layer.
2. Decrease the number of nodes in each hidden layer.
3. Increase the number of hidden layers.
4. Start with larger initial values for the weights.
5. Train the net until the actual output equals the desired output for all input training samples.
6. Train the net until the actual output converges on the desired output, *i.e.,* the actual output is within ϵ of the desired output for all input training samples, for some $\epsilon >> 0$.
7. Interrupt training at set intervals in order to test the net on a random selection of the samples used in training. Terminate training when the performance stops improving.
8. Interrupt training at set intervals in order to test the net on a random selection of the samples that were withheld from the training set. Terminate training when the performance stops improving.

Problem 5 Learning (23 Points)

Your one-person company which specializes in the classification of data sets is doing well. (Remember Quiz 2!) A new client hires you to build a program to classify images of seven-segment display digits. The seven-segment display is the type of display commonly used in digital watches, calculators, and instruments to display decimal data. A digit is displayed by illuminating a subset of the seven line segments shown in the figure below:



Let the variables $s_1 \ldots s_7$ represent the states of the line segments. Each s_i is a boolean variable: its value is 1 if the corresponding line segment is ON and 0 if the line segment is OFF. Each digit is completely described by the values of the seven variables as shown in the table below:

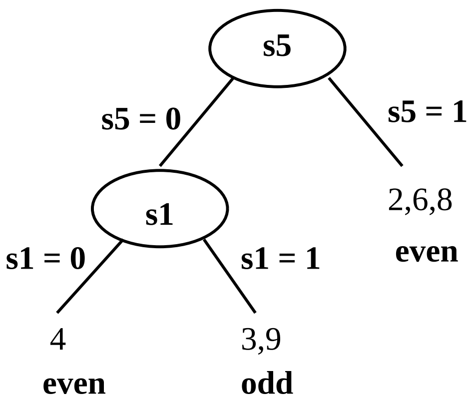
digit	s_1	s_2	s_3	s_4	s_5	s_6	s_7	even?
0	1	1	1	0	1	1	1	yes
1	0	0	1	0	0	1	0	no
2	1	0	1	1	1	0	1	yes
3	1	0	1	1	0	1	1	no
4	0	1	1	1	0	1	0	yes
5	1	1	0	1	0	1	1	no
6	1	1	0	1	1	1	1	yes
7	1	0	1	0	0	1	0	no
8	1	1	1	1	1	1	1	yes
9	1	1	1	1	0	1	1	no

This data is also available on the last page of the exam booklet.

As a test problem, your client asks you to construct a classifier that identifies the even digits. To test the predictive accuracy of a classification method, you split the 10 digits into two sets: the training set and the test set. We will assume the training set consists of six digits: four even digits (2,4,6,8) and two odd digits (3,9). The test set consists of the remaining four digits (0,1,5,7).

Part A (3 Points)

Given your previous success with the identification tree method, you run the `make-tree` procedure on the training set. (To your delight, your dog that ruined the floppy containing the `make-tree` procedure in Quiz 2 spares the backup diskettes this time.) The procedure returns the following identification tree for the training set:



p	p log p
$\frac{1}{6}$	-0.431
$\frac{1}{5}$	-0.464
$\frac{1}{4}$	-0.5
$\frac{1}{3}$	-0.528
$\frac{2}{5}$	-0.529
$\frac{1}{2}$	-0.5
$\frac{3}{5}$	-0.442
$\frac{2}{3}$	-0.39
$\frac{3}{4}$	-0.311
$\frac{4}{5}$	-0.258
$\frac{5}{6}$	-0.219

Figure 5: Identification tree for the training set: 2,3,4,6,8, 9. You may find the table on right useful in answering part A.

Calculate the average disorder for the top-level test node: s5.

Answer:

Part B (2 Points)

What does the identification tree predict for the test cases? Complete the following table:

Answer:

digit	prediction
0	
1	
5	
7	

Part C (5 Points)

You wonder how `Procedure W` would learn the concept of even digit. You decide to encode the situations operated on by `Procedure W` as follows:

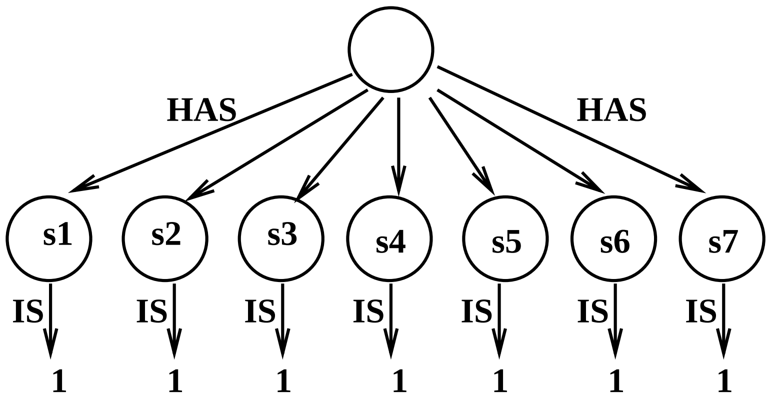


Figure 6: Description of the digit 0.

Consider the following sequence of training samples:

$$8, 9, 3, 2, 4, 6$$

Because there is no natural way to rank the importance of the seven variables, we will assume the `specialize` routine of `Procedure W` *ignores samples that have more than one difference from the evolving model*. As described in the book (pp 356-7), however, the `generalize` routine does handle multiple differences. Complete the following table to indicate the relevant induction heuristics used by `Procedure W` in the course of examining the training sequence. Choose the heuristics from the following list:

REQUIRE, FORBID, DROP-LINK, IGNORE

Note: *Ignore the exception-generating heuristics (the no-altering principle) described on page 358.*

Answer:

digit	induction heuristics	affected node
8	initial model	–
9		
3		
2		
4		
6		

Draw the model learned after this training sequence:

Part D (3 Points)

For the particular training sequence in part C, do `Procedure W` and the identification tree produced in part A agree on the classification of all 10 digits?

Circle the answer: YES NO

If you circle YES, explain if the concept is learned independent of the order of the training sequence.

If you circle NO, is there a reordering of the training sequence that might allow `Procedure W` to learn the same concept given by the identification tree? Why or why not?

Answer:

Part E (4 Points)

You want to explore the nearest neighborhood classification method on the problem of learning the even digits from the positive samples (2,4,6,8) and the negative samples (3,9). A natural metric to use is the Hamming distance between the vectors of line segment variables of two digits. Recall the Hamming distance between two bit vectors is the number of bits that differ in their corresponding positions.

Assuming that your the classification procedure reports even, odd, or tie, show the predictions of the nearest neighbor method for the test set:

Answer:

digit	prediction
0	
1	
5	
7	

Part F (6 Points)

Suppose you now have to work with a digit display that has two groups of segments: one group consists of the original, reliable, seven segments; and the other group consists of many "christmas tree" segments that are turned on or off at random each time the number presented changes. You do not know which segment belongs to which group.

Now the training set is based, as before, on positive samples (2,4,6,8) and the negative samples (3,9). This time, however, each digit is presented thousands of times (so for some presentations, a particular "christmas tree" segment will be on, and for others that same segment will be off).

Similarly, the test set is based, as before, on (0,1,5,7), but this time, any particular presentation of a test sample will include both the original segments and the randomly set "christmas tree" segments.

By circling the correct answers below, you are to show the predictive accuracy of each classification method given the new samples and new training set, relative to the performance for the same classification method working on the original samples and the original test set.

For `Identification Tree`:

```
A. predictive accuracy typically increases significantly
B. predictive accuracy typically remains roughly the same
C. predictive accuracy typically decreases significantly
```

For `Procedure W`:

```
A. predictive accuracy typically increases significantly
B. predictive accuracy typically remains roughly the same
C. predictive accuracy typically decreases significantly
```

For `Nearest Neighbor`:

```
A. predictive accuracy typically increases significantly
B. predictive accuracy typically remains roughly the same
C. predictive accuracy typically decreases significantly
```

Problem 6 Miscellaneous (10 points)

Each of the following true-false questions is worth 1 point. **If a statement is partly true and partly false, mark it false.**

1. T F Genetic algorithms basically do a kind of beam search in which the heuristic measure of "quality" involves a probabilistic selection biased toward the "fittest" individuals.
2. T F Populations of organisms in which the individuals can learn, can, in principle, evolve faster.
3. T F Some linguists believe that each of the forty or so phonemes in English correspond to particular values for a dozen or so distinctive features, such as the voicing feature.
4. T F The defect of the Yip–Sussman model of learning—the one that learns English pluralization rules for dog, cat, and horse—is that the learner must be presented thousands of samples and near misses from a knowledgeable teacher.
5. T F The purpose of multiple-argument dispatch—as offered, for example, by the commonlisp object system (CLOS)—is to enable class hierarchies to branch both upward and downward.
6. T F Two principles—the left-before-right principle and the subclass before superclass principle—ensure that, given any classification hierarchy, it is possible to determine which of many inherited procedures ought to take precedence.
7. T F Three sufficiently different views of a transparent polyhedron are sufficient to predict what the polyhedron will look like from any other view, but only if it is possible to determine which vertexes in the images correspond to one another.
8. T F Given a fixed search depth, using minimax with alpha-beta never requires less than half the computation required by straight minimax.
9. T F Bidirectional, breadth-first search between point A and point B requires approximately half the computation required by unidirectional search.
10. T F The frame concept includes the idea that certain small fixed vocabularies are sufficient to describe situations in significant domains.

This page intentionally left (almost) blank.

Rules for Problem 1

```
R1:     If      ?x gives milk
        then    ?x is a mammal                  .7

R2      If      ?x is warm-blooded
        then    ?x is a mammal                  .8

R3      If      ?x gives milk
                ?x does not lay eggs
                ?x bears live young
        then    ?x is a mammal                  .6

R4      If      ?x is a mammal
                ?x eats meat
        then    ?x is a carnivore               .9

R5      If      ?x is a mammal
                ?x has pointed teeth
                ?x has forward-looking eyes
        then    ?x is a carnivore               .5

R6      If      ?x is a mammal
                ?x is a stealthy hunter
        then    ?x is a carnivore               .5

R7      If      ?x has whiskers
                ?x has claws
        then    ?x is a feline                  .8

R8      If      ?x has light fur
                ?x has dark vertical bands
        then    ?x has stripes                  .7

R9      If      ?x is a carnivore
                ?x is a feline
                ?x is tawny colored
                ?x has stripes
        then    ?x is a tiger                   .9
```

Data for Problem 5

Let the variables $s_1 \ldots s_7$ represent the states of the line segments. Each s_i is a boolean variable: its value is 1 if the corresponding line segment is ON and 0 if the line segment is OFF. Each digit is completely described by the values of the seven variables as shown in the table below:

digit	s_1	s_2	s_3	s_4	s_5	s_6	s_7	even?
0	1	1	1	0	1	1	1	yes
1	0	0	1	0	0	1	0	no
2	1	0	1	1	1	0	1	yes
3	1	0	1	1	0	1	1	no
4	0	1	1	1	0	1	0	yes
5	1	1	0	1	0	1	1	no
6	1	1	0	1	1	1	1	yes
7	1	0	1	0	0	1	0	no
8	1	1	1	1	1	1	1	yes
9	1	1	1	1	0	1	1	no