Chapter 2 Solutions

2.1)

Each hypothesis specifies either a required attribute value or "don't care", yielding 4*3*3*3*3*3 = 972 possible combinations.

Adding Water-Current as an attribute, which can take on 3 values, multiplies the number of possible instances by 3 (to get 288) and the number of possible hypotheses by 4 (to get 2888).

In general, adding an attribute with k possible values, multiplies the number of possible instances by k and the number of possible hypotheses by k+1.

2.2 Similar to 2.2 Exercise:
Consider the following set of training examples (taken from Dean, Allen, and Aloimonos' book) to train a robot janitor to predict whether or not an office contains a recycling bin.

	STATUS	FLOOR	DEPT.	OFFICE SIZE	RECYCLING BIN?
1.	faculty	four	cs	Medium	yes
2.	faculty	four	ee	Medium	yes
3.	student	four	cs	Small	no
4.	faculty	five	cs	Medium	yes

Give a sequence of S and G boundary sets computed by the CANDIDATE-ELIMINATION algorithm if it is given the sequence of examples above in the order in which they appear on the table.

S0 = {[null,null,null,null]}
S1 = {[faculty,four,cs,medium]}
S2 = {[faculty,four,?,medium]}
S3 = S2
S4 = {[faculty,?,?,medium]}

G4 = G3
G3 = {[faculty,?,?,?], [?,?,?,medium]}
G2 = G1
G1 = G0

G0 = {[?,?,?,?]}

Exercise 2.2
Give the sequence of S and G boundary sets computed by the CANDIDATE-ELIMINATION algorithm if it is given the sequence of training examples from Table 2.1 in reverse order. Although the final version space will be the same regardless of the sequence of examples (why?), the sets S and G computed at intermediate stages will, of course, depend on this sequence. Can you come up with ideas for ordering the training examples to minimize the sum of the sizes of these intermediate S and G sets for the H used in the EnjoySport example?

Solution:

Initial State

S_0:

{< *Ø, Ø, Ø, Ø, Ø, Ø* >}

G_0:

{< *?, ?, ?, ?, ?, ?* >}

Training Example #4

S_1:

{< *Sunny, Warm, High, Strong, Cool, Change* >}

G_1:

{< *?, ?, ?, ?, ?, ?* >}

Training Example #3

S_2:

{< *Sunny, Warm, High, Strong, Cool, Change* >}

G_2:

{< *Sunny, ?, ?, ?, ?, ?* > < *?, Warm, ?, ?, ?, ?* > < *?, ?, ?, ?, Cool, ?* >}

Training Example #2

S_3:

{< *Sunny, Warm, High, Strong, ?, ?* >}

G_3:

{< *Sunny, ?, ?, ?, ?, ?* > < *?, Warm, ?, ?, ?, ?* >}

Training Example #1

S_4:

{< *Sunny, Warm, ?, Strong, ?, ?* >}

G_4:

{< *Sunny, ?, ?, ?, ?, ?* > < *?, Warm, ?, ?, ?, ?* >}

2.4 Exercise:
Consider the instance space consisting of integer points in the x, y plane and the set of hypotheses H consisting of rectangles. More precisely, hypotheses are of the form a <= x <= b, c <= y <= d, where a, b, c, and d can be any integers.

Consider the version space with respect to the set of positive (+) and negative (-) training examples shown below. What is the S boundary of the version space in this case? Write out the hypotheses and draw them in on the diagram.

Please see attached sheet for drawing.
S:
{< 4 <= x <= 6, 3 <= y <= 5 >}

This assumes that a rectangle is at minimum 1 x 1. This is also assuming that generalization or specification is the decreasing or increasing in value of a, b, c, or d.
What is the G boundary of this version space? Write out the hypotheses and draw them in.

Please see attached sheet for drawing.
G:
{< 3 <= x <= 8, 2 <= y <= 7 > }

My answer was missing { < 2 <= x <= 8, 2 <= y <= 5 > }
Suppose the learner may now suggest a new x, y instance and ask the trainer for its classification. Suggest a query guaranteed to reduce the size of the version space, regardless of how the trainer classifies it. Suggest one that will not.

The learner could request (7, 4) for classification. Actually, any point in (4 <= x <= 7, y = 6) or (x = 7, 3 <= y <= 6) will work. This is because the points along thesey two lines are between the version space bounds indentified by S and G. Since S and G should converge upon one hypothesis, one must generalize or specialize, respectively.
By selecting (5, 4), (4, 5), (5, 5), (6, 3) or (6, 4) reducing the space should be avoided. Since these points are already included by S, there should be no change in the space. This is a result of the bias imposed by the hypothesis representation.

Note: This only works with my original answer in the previous problem.
Now assume you are a teacher, attempting to teach a particular target concept (e.g., 3 <= x <= 5, 2 <= y <= 9). What is the smallest number of training examples you can provide so that the Candidate-Elimination algorithm will perfectly learn the target concept?

For the target concept to be learned exactly, G and S must converge, hence negative and positive examples must be given. One set of opposite points from the concept rectangle should be given as positive examples to force S to include or generalize to the rectangle concept (e.g., (3, 9) and (5, 2)). The other set of opposite corners should be expanded horizontally and vertically by 1 and be given as negative examples to restrict or specialize G to exclude everything but the concept rectangle (e.g., (3, 2)->(2, 1) and (5, 9)->(6, 10)). This all requires that 4 training examples be given.

2.5 Exercise:
A)

S1: {<<male brown tall US><female black short US>>}
G1: {<< ? ? ? ?>< ? ? ? ?>>}

S2: {<<male brown ? ?><female black short US>>}
G2: {<< ? ? ? ?>< ? ? ? ?>>}

S3: {<<male brown ? ?><female black short US>>}
G3: {<<male ? ? ?>< ? ? ? ?>>,

<< ? ? ? ?>< ? ? ? US>>}

S4: {<<male brown ? ?><female ? short ?>>}
G4: {<<male ? ? ?>< ? ? ? ?>>}

2.5B)

Given the single positive example, the consistent hypotheses are those which for each attribute either require the same value in the example, or don't care. This gives two possibilities for each attribute, for a total of 2^8 = 256 consistent hypotheses.

2.5C)

Once you have a single positive instance, for each attribute a single query may be generated to determine whether or not the specific value for the attribute in the original positive instance is of relevance to the target concept. To generate such a query, it is sufficient to copy the values from the original positive instance, but change the value of the attribute-in-question to something different. Such a series of queries will be of length equal to the number of attributes, and will assure that the learner will converge to a single, correct hypothesis, given that the hypothesis is expressible in the target language.

For example:

<<female black short Portuguese><female red tall Indian>>
^^^^^^
<<male brown short Portuguese><female red tall Indian>>
^^^^^
<<male black tall Portuguese><female red tall Indian>>
^^^^
<<male black short French ><female red tall Indian>>
^^^^^^
<<male black short Portuguese><male red tall Indian>>
^^^^
<<male black short Portuguese><female black tall Indian>>
^^^^^
<<male black short Portuguese><female red medium Indian>>
^^^^^^

<<male black short Portuguese><female red tall US >>
^^

Once a single positive example has been seen, each query will halve the remaining version space. The number of queries (8) necessary to converge on a single hypothesis is thus log-base-two of the size of the hypothesis space (256, as in part B).

2.5 D

Once you start working with a complete hypothesis space in the version space framework, you lose all inductive bias, and hence the ability to generalize to unseen examples. Therefore, to converge on a single hypothesis, you'd have to generate a query for each possible instance, other than the one already seen, in order to determine whether or not it would be covered by the target concept. Such a sequence would be of length (2*3*3*7)*(2*3*3*7) = 15876.