Concept Learning

Prof(Dr) Premanand Ghadekar

Out Line

- Introduction to Concept & Concept Learning
- Hypothesis
- Find S-Algorithm
- Viriance Bias etc.

Concept & Concept Learning

- **Classification** also called **Concept Learning**, consists of learning a description of a class of objects.
- This description is typically used to predict whether new objects fit the class.

Example Classification Tasks

- **Classify parts into -** defective or OK.
- **Mammogram analysis -** given a Mammogram, estimate the probability that it is normal, pre-cancerous or cancerous.
- **Document understanding** - given a rectangular region from a scanned region, classify it is as text or graphics.

Machine Learning for Classification (Concept Learning)

- Assume you have a goal concept that you are trying to learn, called the **target concept.**
- Your guesses or approximations of the target concept are called **hypotheses.**
- An object (fact) which is used to help learn the goal concept is called an **instance or an example**

Machine Learning for Classification (Concept Learning)

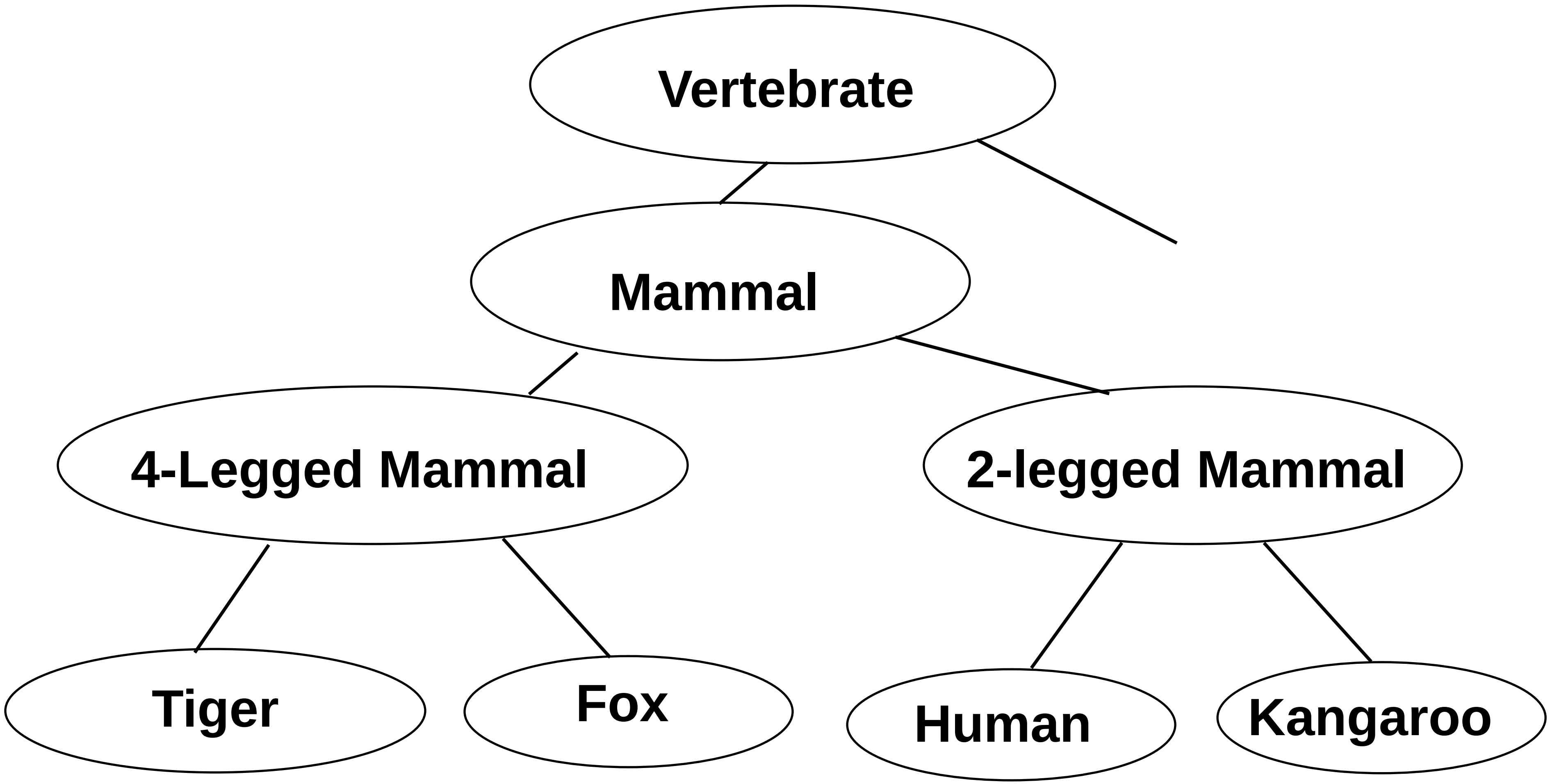
- An instance/example x is described by a vector of features also called **attributes,** i.e. $c=\langle x_1 ...., xn \rangle$
- **Features can be:-**

- **Nominal,** if there is no structure given to the values. Example: colour.
- **Ordered,** if there is an order to the values.

  Examples: Temperature, weight
- **Structured,** if the values can be put in a tree structure, called a generalization hierarchy.

  Example: animal taxonomy.

A Structured Attribute



Concept Learning Problem

- Given: A labeling function f that maps feature vectors into a discrete set of a k classes. That is,
- $f : x \longrightarrow \{0,1,2 \ldots\ldots\ldots k-1\}$
- Often, there are only 2 classes, called "positive" (+) and "negative" (-).
- **Target Concept** is called **Hypothesis.**
- Represent each training example as a pair (x, f(x)). These are the example that will be used for learning the concept.
- Problem:-From a set of (x, f(x)) pairs, learn the target concept f.

Concept Learning Problem

- Given: <x, f(x)> pairs, infer f.
- Given a finite sample, it is often impossible to guess the true function f.
- Approach; Find some pattern (Called a Hypothesis) in the training examples and assume that the pattern will hold for future examples too.

x	f(x)
1	1
2	4
3	9
4	16
5	?

Another Example

- Suppose you want to **learn the Concept "apple".**
- You get the following training instance:

Colour	Shape	Diameter	Apple?
Red	Round	4" dia	+
Yellow	Round	4.3" dia	+
Green	Square	5" dia	-
Green	Round	3" dia	+

Another Example

- Then you could learn a rule or set of rules to distinguish positive from negative examples.
- These are called **classification rules**. Some example might be:

Round ➡ apple

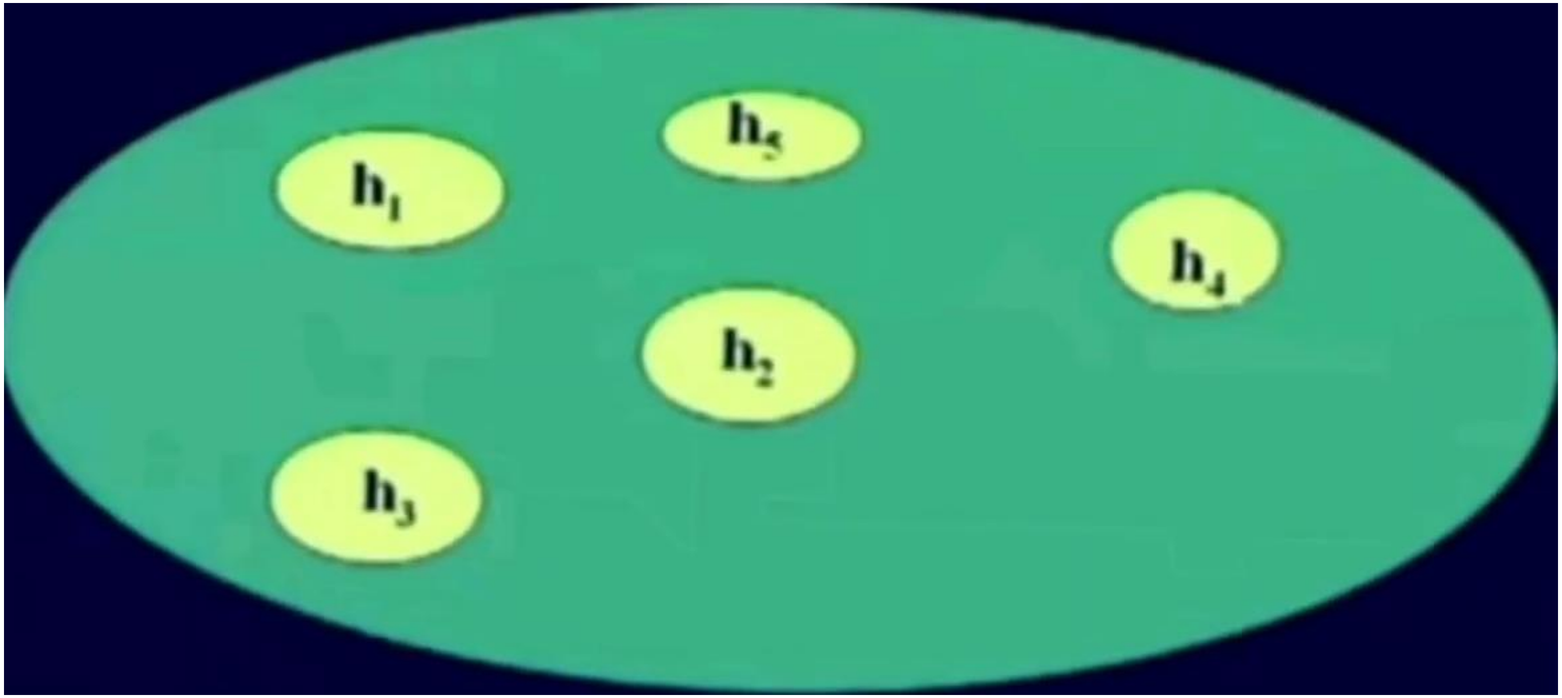
diameter < 5" ➡ apple

round & diameter < 5" ➡ apple

Hypothesis or Model Selection

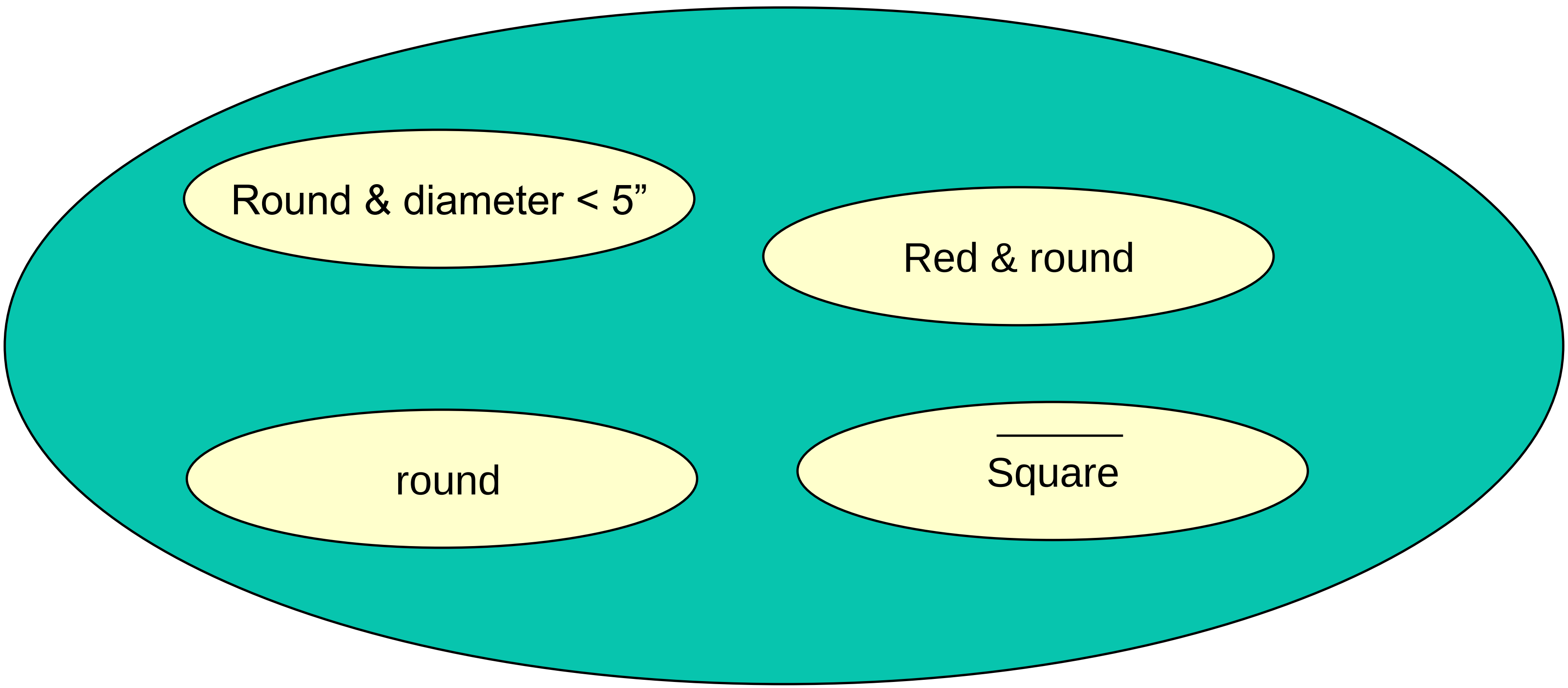
- What **class of hypotheses should we consider?**
- Assume f is a set of rules. Then the space of hypotheses consists of rule sets.
- The space of hypotheses may consist of simple polynomials. Regression could be used to learn f.
- The space of hypotheses may consist of decision trees.
- The space of hypotheses may consist of neural nets, and we need to learn the weights.

Hypotheses space H



Hypotheses space H

Suppose you want to learn the Concept “apple”.
Some possible Hypotheses



Some Definitions

Training Set- The set of all training examples given to the learner.

Consistent hypothesis – A hypothesis that is consistent with all of the training examples.

Testing Set- The set of all examples given to the learner after it has learned its hypothesis. This set is used to test the accuracy (correctness) of the learned hypothesis over unseen (new) examples.

Some Definitions

Consistent hypothesis- A hypothesis that is consistent with all of the training examples.

If the examples are labeled + and – then a consistent hypothesis is one that implies all of the + examples and none of the – examples.

FIND S Algorithm

- **Find-S Algorithm Machine Learning**
- FIND S Algorithm is used to find the **Maximally Specific Hypothesis.**
- Using the Find-S algorithm gives a single maximally specific hypothesis for the given set of training examples

FIND S Algorithm

❖ Find-S Algorithm Machine Learning

1. Initilize h to the most specific hypothesis in H

2. For each positive training instance x

    For each attribute contraint ai in h

        If the contraint ai is satisfied by x

            then do nothing

        Else

            replace ai in h by the next more general constraint that is satisfied by x

3. Output the hypothesis h

FIND S Algorithm

❖ Maximally Specific Hypothesis Solved Example

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

1. How many concepts are possible for this instance space?

Solution: 2 * 3 * 2 * 2 * 3 = 72

2. How many hypotheses can be expressed by the hypothesis language?

Solution: 4 * 5 * 4 * 4 * 5 = 1600

3. Calculate Semantically Distinct Hypothesis

Solution: (3 * 4 * 3 * 3 * 4) + 1 = 433

FIND S Algorithm

❖ Maximally Specific Hypothesis Solved Example

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

1. **Apply the FIND-S algorithm by hand on the given training set. Consider the examples in the specified order and write down your hypothesis each time after observing an example.**

Soln-

Step 1:

h0 = (ø, ø, ø, ø, ø)

Step 2:

X1 = (some, small, no, expensive, many) – No

Negative Example Hence Ignore

FIND S Algorithm

Soln- h1 = (ø, ø, ø, ø, ø)

X2 = (many, big, no, expensive, one) – Yes

h2 = (many, big, no, expensive, one)

X3 = (some, big, always, expensive, few) – No

Negative example hence Ignore

h3 = (many, big, no, expensive, one)

X4 = (many, medium, no, expensive, many) – Yes

h4 = (many, ?, no, expensive, ?)

X5 = (many, small, no, affordable, many) – Yes

h5 = (many, ?, no, ?, ?)

Step 3:

Final Maximally Specific Hypothesis is:

h5 = (many, ?, no, ?, ?)

FIND S Algorithm

FIND-S: Step-2

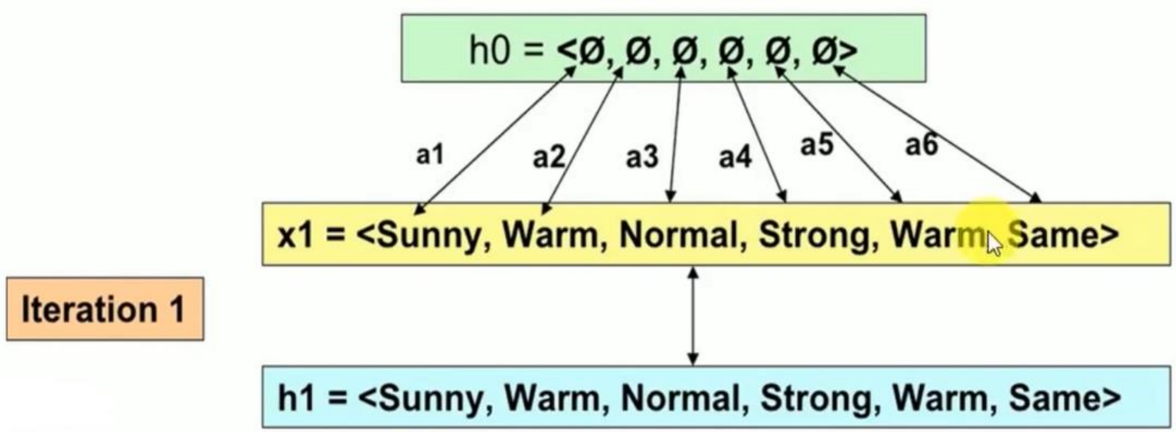
Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

2. For each positive training instance x
   - For each attribute constraint a_i in h

     If the constraint a_i is satisfied by x

     Then do nothing

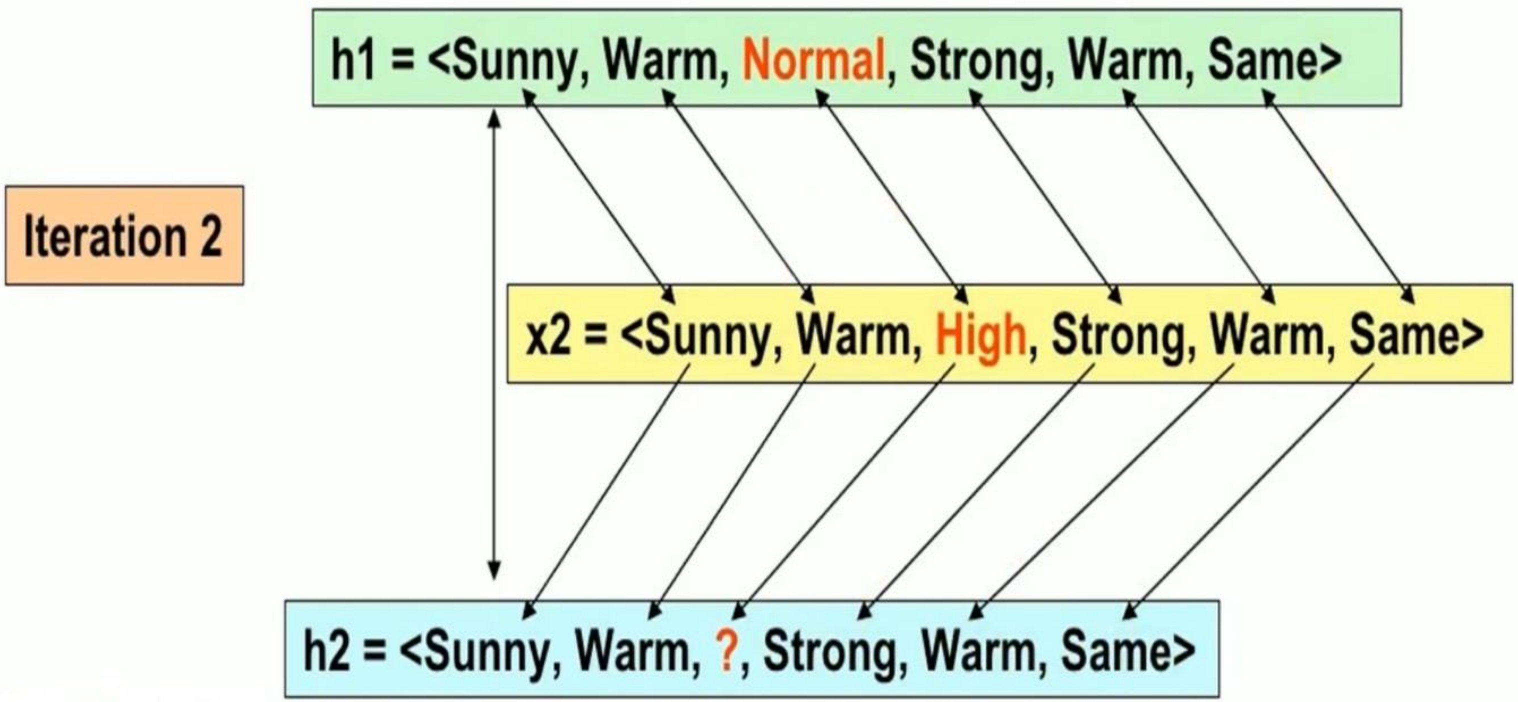
     Else replace a_i in h by the next more general constraint that is satisfied by x



FIND S Algorithm

FIND-S: Step-2

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes



FIND S Algorithm

FIND-S: Step-2

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Iteration 3

Ignore

h3 = <Sunny, Warm, ?, Strong, Warm, Same>

FIND S Algorithm

FIND-S: Step-2

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

h3 = < Sunny, Warm, ?, Strong, Warm, Same >

Iteration 4

x4 = < Sunny, Warm, High, Strong, Cool, Change >

Step 3

Output

h4 = <Sunny, Warm, ?, Strong, ?, ?>

Thanks