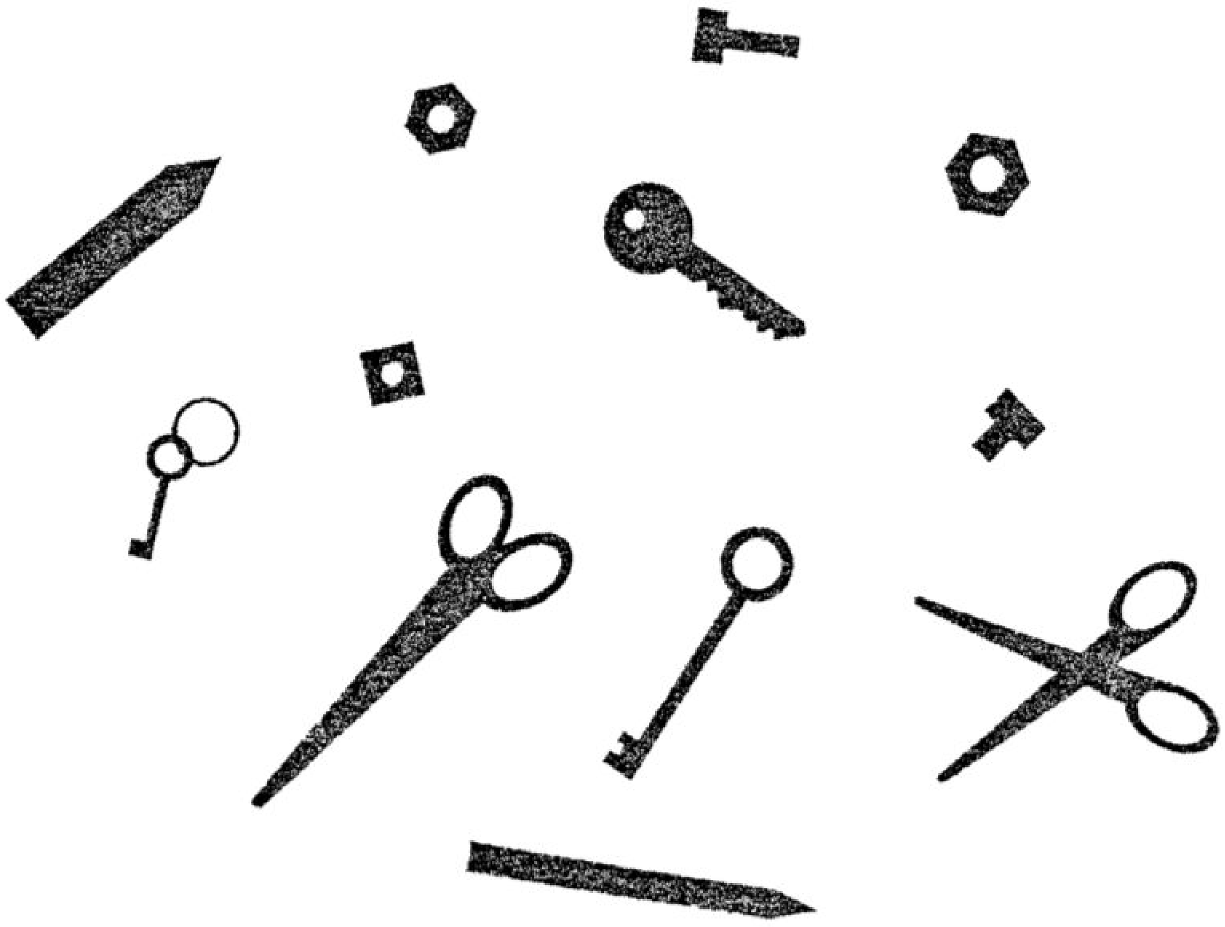
Inducing Rules from Examples

In this lab, we'll explore a kind of learning where examples and hypotheses are described in terms of a set of attributes. An object is described by specifying concrete values of the attributes in the object. Such a description is then a vector of attribute values.
The following figure shows some objects which belong to 5 classes : nut, screw, key, pen, scissors. In this case, the attributes are : the size, the shape and the number of holes in an



object.

Let the possible values of these attributes be :

size : small, large
shape : long, compact, other
holes : none, 1, 2, 3, many

Assume that a vision system has extracted the 3 attribute values for each object. These attribute definitions and the example objects can then be represented as a set of Prolog facts.

attribute(Attribute, [Val1, Val2, …]).
example(Class, [Attribute1 = Val1, Attribute2 = Val2, …])

Now assume that these facts are communicated to a learning program that is supposed to learn about the 5 classes. The result of learning will be a description of the classes in the form of rules that can be used for classifying new objects. For example, once the learning for classes nut and key is done, the output should be something like :

nut ⇐ [[size = small, holes = 1]]
key ⇐ [[shape = long, holes = 1], [shape = other, holes = 2]]

The meaning of these rules is :

An object is a nut if
    Its size is small and
    It has 1 hole

An object is a key if
    Its shape is long and
    It has 1 hole
    OR
    Its shape is ‘other’ and
    It has 2 holes

For learning such kind of rules, we’ll be using the **covering algorithm** which outputs **sound** and **complete** rules **:**

```
------------------------------------------------------------------------------------------------------------------
To induce a list of rules RULELIST for a set S of classified examples, do:

RULELIST :- empty
E :- S
while E contains positive examples do
    begin
        RULE :- InduceOneRule(E);
        Add RULE to RULELIST;
        Remove from E all the examples covered by RULE;
    end
------------------------------------------------------------------------------------------------------------------
```

The general form of such rules is :

Class ⇐ [Conj1, Conj2, ….]
Where Conj1, Conj2 are lists of attribute values of the form :
[Att1 = Val1, Att2 = Val2, …..]

A class description [Conj1, Conj2, ….] is interpreted as follows :

1.) An object matches the description if the object satisfies at least one of **Conj1, Conj2**, etc
2.) An object satisfies **Conj** if all the attribute values in Conj are same as in the object.

Thus attribute values in **Conj** are related conjunctively : none of them may be contradicted by the object whereas the lists **Conj1, Conj2,** etc., are related disjunctively : at least one of them has to be satisfied.

Our task today is to implement this rule based learning in Prolog.