candidate algorithm

In [3]:
```
import numpy as np
import pandas as pd
```

In [6]:
```
a=pd.read_csv('lab4.csv')
a
```

Out[6]:

	Color	Toughness	Fungus	Appearance	Poisonous
0	green	hard	no	wrinkled	yes
1	green	hard	yes	smooth	no
2	brown	soft	no	wrinkled	no
3	orange	hard	no	wrinkled	yes
4	green	soft	yes	smooth	yes

In [8]:
```
concepts = np.array(a.iloc[:,0:-1])
print("\nInstances are:\n",concepts)
```

```
Instances are:
 [['green' 'hard' 'no' 'wrinkled']
 ['green' 'hard' 'yes' 'smooth']
 ['brown' 'soft' 'no' 'wrinkled']
 ['orange' 'hard' 'no' 'wrinkled']
 ['green' 'soft' 'yes' 'smooth']]
```

In [10]:
```
target = np.array(a.iloc[:,-1])
print("\nTarget Values are: ",target)
```

```
Target Values are:  ['yes' 'no' 'no' 'yes' 'yes']
```

In [11]:
```
def learn(concepts, target):
    specific_h = concepts[0].copy()
    print("\nInitialization of specific_h and genearal_h")
    print("\nSpecific Boundary: ", specific_h)
    general_h = [["?" for i in range(len(specific_h))] for i in range(len(specifi
    print("\nGeneric Boundary: ",general_h)

    for i, h in enumerate(concepts):
        print("\nInstance", i+1 , "is ", h)
        if target[i] == "yes":
            print("Instance is Positive ")
            for x in range(len(specific_h)):
                if h[x]!= specific_h[x]:
                    specific_h[x] ='?'
                    general_h[x][x] ='?'

        if target[i] == "no":
            print("Instance is Negative ")
            for x in range(len(specific_h)):
                if h[x]!= specific_h[x]:
                    general_h[x][x] = specific_h[x]
                else:
                    general_h[x][x] = '?'

        print("Specific Boundary after ", i+1, "Instance is ", specific_h)
        print("Generic Boundary after ", i+1, "Instance is ", general_h)
        print("\n")

    indices = [i for i, val in enumerate(general_h) if val == ['?', '?', '?', '?
    for i in indices:
        general_h.remove(['?', '?', '?', '?', '?', '?'])
    return specific_h, general_h

s_final, g_final = learn(concepts, target)
```

```
Initialization of specific_h and genearal_h

Specific Boundary:  ['green' 'hard' 'no' 'wrinkled']

Generic Boundary:  [['?', '?', '?', '?'], ['?', '?', '?', '?'], ['?', '?', '?',
'?'], ['?', '?', '?', '?']]

Instance 1 is  ['green' 'hard' 'no' 'wrinkled']
Instance is Positive
Specific Boundary after  1 Instance is  ['green' 'hard' 'no' 'wrinkled']
Generic Boundary after  1 Instance is  [['?', '?', '?', '?'], ['?', '?', '?',
'?'], ['?', '?', '?', '?'], ['?', '?', '?', '?']]



Instance 2 is  ['green' 'hard' 'yes' 'smooth']
Instance is Negative
Specific Boundary after  2 Instance is  ['green' 'hard' 'no' 'wrinkled']
Generic Boundary after  2 Instance is  [['?', '?', '?', '?'], ['?', '?', '?',
```

```
'?'], ['?', '?', 'no', '?'], ['?', '?', '?', 'wrinkled']]


Instance 3 is  ['brown' 'soft' 'no' 'wrinkled']
Instance is Negative
Specific Boundary after  3 Instance is  ['green' 'hard' 'no' 'wrinkled']
Generic Boundary after  3 Instance is  [['green', '?', '?', '?'], ['?', 'hard',
'?', '?'], ['?', '?', '?', '?'], ['?', '?', '?', '?']]


Instance 4 is  ['orange' 'hard' 'no' 'wrinkled']
Instance is Positive
Specific Boundary after  4 Instance is  ['?' 'hard' 'no' 'wrinkled']
Generic Boundary after  4 Instance is  [['?', '?', '?', '?'], ['?', 'hard',
'?', '?'], ['?', '?', '?', '?'], ['?', '?', '?', '?']]


Instance 5 is  ['green' 'soft' 'yes' 'smooth']
Instance is Positive
Specific Boundary after  5 Instance is  ['?' '?' '?' '?']
Generic Boundary after  5 Instance is  [['?', '?', '?', '?'], ['?', '?', '?',
'?'], ['?', '?', '?', '?'], ['?', '?', '?', '?']]
```

In [12]:
```
print("Final Specific_h: ", s_final, sep="\n")
print("Final General_h: ", g_final, sep="\n")
```

```
Final Specific_h:
['?' '?' '?' '?']
Final General_h:
[['?', '?', '?', '?'], ['?', '?', '?', '?'], ['?', '?', '?', '?'], ['?', '?',
'?', '?']]
```

In [ ]:
```

```