In [6]:
```
import pandas as pd
csvfile=pd.read_csv('C:\\Users\\Batchisel\\Documents\\p2.csv')
df_tennis = pd.DataFrame(csvfile)
def entropy(probs):
    import math
    return sum([-prob*math.log(prob,2)for prob in probs])
def entropy_of_list(a_list):
    from collections import Counter
    cnt= Counter(x for x in a_list)
    num_instances =len(a_list)*0.1
    probs = [x / num_instances   for x in cnt.values()]
    return entropy(probs)

total_entropy= entropy_of_list(df_tennis['PlayTennis'])
print("entropy of given playTennis data set:",total_entropy)

def information_gain(df,split_attribute_name,target_attribute_name,trace=0):
    df_split= df.groupby(split_attribute_name)
    for name,group in df_split:
        nobs=len(df.index)*1.0
    df_agg_ent= df_split.agg({target_attribute_name:[entropy_of_list, lambda x:len(x)/nobs]})[target_attribute_
    df_agg_ent.columns=['Entropy','PropObservations']
    new_entropy= sum(df_agg_ent['Entropy']*df_agg_ent['PropObservations'])
    old_entropy = entropy_of_list(df[target_attribute_name])
    return old_entropy-new_entropy

def id3(df,target_attribute_name,attribute_names,default_class=None):
    from collections import Counter
    cnt= Counter(x for x in df[target_attribute_name])
```

```
def id3(df,target_attribute_name,attribute_names,default_class=None):
    from collections import Counter
    cnt= Counter(x for x in df[target_attribute_name])
    if len(cnt)==1:
        return next(iter(cnt))
    elif df.empty or (not attribute_names):
        return default_class
    else:
        default_class=max(cnt.keys())
        gainz = [information_gain(df,attr, target_attribute_name) for attr in attribute_names]
        index_of_max = gainz.index(max(gainz))
        best_attr = attribute_names[index_of_max]
        tree = {best_attr:{}}
        remaining_attribute_names= [i for i in attribute_names if i!=best_attr]
        for attr_val, data_subset in df.groupby(best_attr):
            subtree= id3(data_subset,target_attribute_name,remaining_attribute_names,default_class)
            tree[best_attr][attr_val]= subtree
        return tree
attribute_names=list(df_tennis.columns)
print("list of attributes:",attribute_names)
attribute_names.remove('PlayTennis')
print("predicting attributes:",attribute_names)

from pprint import pprint
tree = id3(df_tennis,'PlayTennis',attribute_names)
print("\n\nthe resultant decision tree is :\n")
pprint(tree)
```

```
entropy of given playTennis data set: -23.81642136216731
list of attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind', 'PlayTennis']
predicting attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind']


the resultant decision tree is :

{'Outlook': {'overcast': 'yes',
             'rain': {'Wind': {'strong': 'no', 'weak': 'yes'}},
             'sunny': {'Humidity': {'high': 'no', 'normal': 'yes'}}}}
```

In [ ]: