In [38]:

```
import getopt
import sys

import numpy as np
import pandas as pd
import query

database = dict()


def read_table(table):
    file = open(table, "r")
    lines = file.readlines()
    lines = [line.replace("\n", "") for line in lines]
    lines = [line.strip() for line in lines]
    lines = [line.split(',') for line in lines]
    table_name = lines[0][0]
    data = np.array(lines[1:])
    print('table data', data)
    columns = []
    for i in xrange(data.shape[1] - 1):
        columns.append('Var' + str(i+1))
    columns.append('Prob')
    df = pd.DataFrame(data = data, columns = columns)
    database[table_name] = df
    print('dataframe',df)


def parse_query(query):
    file = open(query, "r")
    lines = file.readlines()
    sentence = lines[0]
    sentence.strip().split("||")

    print sentence.strip().split("||")
    queryline=sentence.strip().split("||")
```

In [42]:

```
tables = ["t1.txt","t2.txt","t3.txt"]
for table in tables:
    read_table(table)
```

```
('table data', array([['0', '0.7'],
       ['1', '0.8'],
       ['2', '0.6']], dtype='|S3'))
('dataframe',   Var1 Prob
0     0  0.7
1     1  0.8
2     2  0.6)
('table data', array([['0', '0.7'],
       ['1', '0.3'],
       ['2', '0.5']], dtype='|S3'))
('dataframe',   Var1 Prob
0     0  0.7
1     1  0.3
2     2  0.5)
('table data', array([['0', '0', '0.8'],
       ['0', '1', '0.4'],
       ['0', '2', '0.5'],
       ['1', '2', '0.6'],
       ['2', '2', '0.9']], dtype='|S3'))
('dataframe',   Var1 Var2 Prob
0     0     0  0.8
1     0     1  0.4
2     0     2  0.5
3     1     2  0.6
4     2     2  0.9)
```

In [37]:

```
    queries = "query1.txt"
    parse_query(queries)
```

```
['R(x1,y1),Q(x1)']
```

In [40]:

```
    file = open("t3.txt" ,"r")
    lines = file.readlines()
    lines = [line.replace("\n", "") for line in lines]
    lines = [line.strip() for line in lines]
    lines = [line.split(',') for line in lines]
    table_name = lines[0][0]
    data = np.array(lines[1:])
    print('table data', data)
```

```
('table data', array([['0', '0', '0.8'],
       ['0', '1', '0.4'],
       ['0', '2', '0.5'],
       ['1', '2', '0.6'],
       ['2', '2', '0.9']], dtype='|S3'))
```

In [43]:
```
print(database)
```

```
{'Q':    Var1 Prob
0     0   0.7
1     1   0.3
2     2   0.5, 'P':    Var1 Prob
0     0   0.7
1     1   0.8
2     2   0.6, 'R':    Var1 Var2 Prob
0     0     0   0.8
1     0     1   0.4
2     0     2   0.5
3     1     2   0.6
4     2     2   0.9}
```

In [44]:
```
print(database['Q'])
```

```
  Var1 Prob
0     0   0.7
1     1   0.3
2     2   0.5
```

In [45]:
```
print(database['P'])
```

```
  Var1 Prob
0     0   0.7
1     1   0.8
2     2   0.6
```

In [56]:
```
database['R']["Prob"]= pd.to_numeric(database['R']["Prob"])
```

In [58]:
```
database['R']["NegProb"]= (1-database['R']["Prob"])
```

In [112]:
```
database['R']["PosProb"]= (1-database['R']["NegProb"])
```

In [113]:
```
print(database['R'])
```

```
  Var1 Var2  Prob  NegProb  PosProb
0     0     0   0.8       0.2       0.8
1     0     1   0.4       0.6       0.4
2     0     2   0.5       0.5       0.5
3     1     2   0.6       0.4       0.6
4     2     2   0.9       0.1       0.9
```

In [114]:
```
print(1-database['R']["NegProb"].product())
Rprob=1-database['R']["NegProb"].product()
```

```
0.9976
```

In [115]:
```
database['Q']["Prob"]= pd.to_numeric(database['Q']["Prob"])
database['Q']["NegProb"]= (1-database['Q']["Prob"])
```

```
In [116]: print(1-database['Q']["NegProb"].product())
          Qprob=1-database['R']["NegProb"].product()

          0.895
```

```
In [117]: Qprob*Rprob

Out[117]: 0.99520576
```

```
In [118]: print('The key is to find the neg probability in each (x,y) pairing, gro
          uping on x')
          print(database['R'].groupby('Var1').prod())
          df = pd.DataFrame(database['R'].groupby('Var1').prod())
          database['Rprod']=database['R'].groupby('Var1').prod()

          The key is to find the neg probability in each (x,y) pairing, grouping
          on x
                Prob  NegProb  PosProb
          Var1
          0     0.16     0.06     0.16
          1     0.60     0.40     0.60
          2     0.90     0.10     0.90
```

```
In [119]: print(database['Q'])

            Var1  Prob  NegProb
          0    0   0.7      0.3
          1    1   0.3      0.7
          2    2   0.5      0.5
```

```
In [120]: print(database['Rprod'])

                Prob  NegProb  PosProb
          Var1
          0     0.16     0.06     0.16
          1     0.60     0.40     0.60
          2     0.90     0.10     0.90
```

```
In [127]: result = pd.merge(database['Rprod'],database['Q'],how='inner', on = 'Var
          1')
```

```
In [128]: print(result)

            Var1  Prob_x  NegProb_x  PosProb  Prob_y  NegProb_y
          0    0    0.16       0.06     0.16     0.7        0.3
          1    1    0.60       0.40     0.60     0.3        0.7
          2    2    0.90       0.10     0.90     0.5        0.5
```

```
In [141]: answer = 1-((1-result["Prob_y"]*(1-result["NegProb_x"]))).prod()
```

```
In [142]: answer

Out[142]: 0.845758
```

```
In [143]: 'yay able to get the 8457 from parsing the files'
```

```
Out[143]: 'yay able to get the 8457 from parsing the files'
```

```
In [ ]:
```