In [1]:

```
from pgmpy.models import BayesianModel
from pgmpy.estimators import BayesianEstimator
from pgmpy.estimators import ConstraintBasedEstimator
from pgmpy.estimators import K2Score, BicScore, MLE
import networkx as nx
import matplotlib.pyplot as plt
import graphviz
import pandas as pd
import torch
import pyro
import numpy as np
%matplotlib inline
%load_ext autoreload
%autoreload 2
```

```
/home/wwymak/anaconda3/envs/immo-ml/lib/python3.7/site-packages/stat
smodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is
deprecated. Use the functions in the public API at pandas.testing in
stead.
  import pandas.util.testing as tm
```

Building a DAG (based on the transport model)

1a) factorization of the joint distribution:

P(A, S, E, O, R, T) = P(A)P(S)P(E|A, S)P(R|E)P(O|E)P(T| R, O)

1b, c)

In [2]:

```
transport_model = BayesianModel([('Age', 'Education'),
                                 ('Sex', 'Education'),
                                 ('Education', 'Occupation'),
                                 ('Education', 'Resident'),
                                 ('Occupation', 'Transport'),
                                 ('Resident', 'Transport')
                                ])
```

1d)

In [3]:

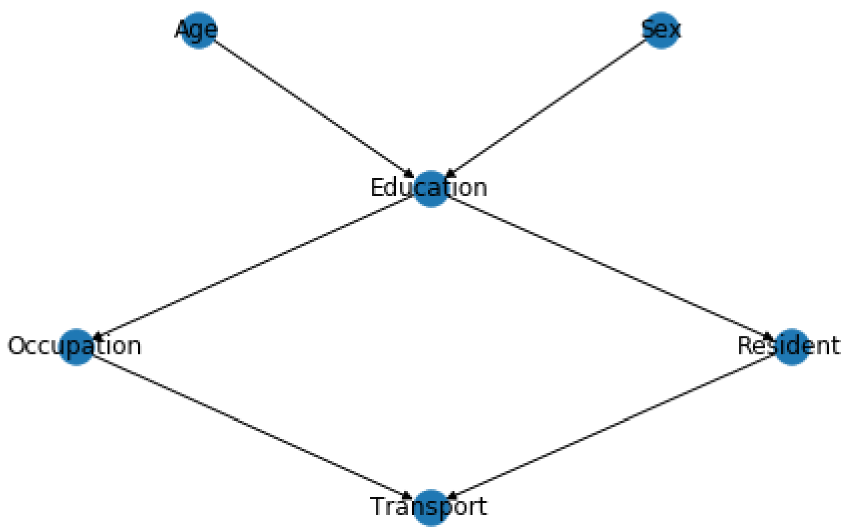
```
transport_model
```

Out[3]:

```
<pgmpy.models.BayesianModel.BayesianModel at 0x7f0c10305550>
```

In [4]:

```
pos1 = nx.nx_pydot.graphviz_layout(transport_model, prog='dot')
nx.draw(transport_model, pos1, with_labels=True)
plt.show()
```



2a)

In [5]:

```
transport_model.nodes
```

Out[5]:

```
NodeView(('Age', 'Education', 'Sex', 'Occupation', 'Resident', 'Tran
sport'))
```

In [6]:

```
transport_model.edges
```

Out[6]:

```
OutEdgeView([('Age', 'Education'), ('Education', 'Occupation'), ('Ed
ucation', 'Resident'), ('Sex', 'Education'), ('Occupation', 'Transpo
rt'), ('Resident', 'Transport')])
```

2b)

In [7]:

```
for node in transport_model.nodes:
    print(f"node: {node}, parents: {transport_model.get_parents(node)}, childre
n: {transport_model.get_children(node)}")
```

```
node: Age, parents: [], children: ['Education']
node: Education, parents: ['Age', 'Sex'], children: ['Occupation',
'Resident']
node: Sex, parents: [], children: ['Education']
node: Occupation, parents: ['Education'], children: ['Transport']
node: Resident, parents: ['Education'], children: ['Transport']
node: Transport, parents: ['Occupation', 'Resident'], children: []
```

2c)

In [8]:

```
#Markov blanket of 'A'
transport_model.get_markov_blanket('Age')
```

Out[8]:

```
['Sex', 'Education']
```

In [9]:

```
#Markov blanket of 'E'
transport_model.get_markov_blanket('Education')
```

Out[9]:

```
['Sex', 'Age', 'Resident', 'Occupation']
```

In [10]:

```
#Markov blanket of 'T'
try:
    transport_model.get_markov_blanket('Transport')
except KeyError as e:
    print(f"Transport has no markov blanket")
```

```
Transport has no markov blanket
```

2d) For each node, it's markov blanket are the set of nodes which if you know the values of (so they are fixed and no longer probabilities), then that node is independent of all the other nodes in the network

2e) The Markov blanket for a node is the set of nodes consisting of the node's parents, it's children, and it's children's other parents.

E.g. for A, it's got no parents, but it has a child('E') and the child has another parent ('S'), so A's markov blanket is S, E

Similarly, for E, parents are A, S, children O, R (these have no other children besides E), so markov blanket for E is A, S, O, R

3a)

In [11]:

```
data = pd.read_csv('survey2.txt', sep=" ")
data = data.rename(columns={"A": "Age", "R": "Resident", "E": "Education",  "O":
"Occupation",
                            "S": "Sex", "T" :"Transport"})
```

In [12]:

```
def create_and_fit_model(data, estimator, equivalent_sample_size):
    transport_model = BayesianModel([('Age', 'Education'),
                                  ('Sex', 'Education'),
                                  ('Education', 'Occupation'),
                                  ('Education', 'Resident'),
                                  ('Occupation', 'Transport'),
                                  ('Resident', 'Transport')
                                 ])
    transport_model.fit(data, estimator=estimator, equivalent_sample_size=equiva
lent_sample_size)
    return transport_model
```

3b)

In [13]:

```
equivalent_sample_sizes = [1,10, 20, 100, 1000]
for sz in equivalent_sample_sizes:
    model =  create_and_fit_model(data, BayesianEstimator, sz)
    nodes = ['Age', 'Education', 'Sex', 'Occupation', 'Resident', 'Transport']
    print('equivalent_sample_sizes/iss:', sz)
    for node in nodes:

        print(model.get_cpds(node))
```

```
equivalent_sample_sizes/iss: 1
+------------+----------+
| Age(adult) | 0.357539 |
+------------+----------+
| Age(old)   | 0.157842 |
+------------+----------+
| Age(young) | 0.484619 |
+------------+----------+
+-----------------+---------------------+---------------------+-----
---------------+--------------------+--------------------+--------
-------------+
| Age             | Age(adult)          | Age(adult)          | Age 
(old)          | Age(old)            | Age(young)          | Age(yo
ung)          |
+-----------------+---------------------+---------------------+-----
---------------+--------------------+--------------------+--------
-------------+
| Sex             | Sex(F)              | Sex(M)              | Sex 
(F)            | Sex(M)              | Sex(F)              | Sex(M)
|
+-----------------+---------------------+---------------------+-----
---------------+--------------------+--------------------+--------
-------------+
| Education(high) | 0.6389365351629502  | 0.7191616766467066  | 0.84
46808510638297 | 0.8913043478260869  | 0.15581051073279054 | 0.80998
24868651488  |
+-----------------+---------------------+---------------------+-----
---------------+--------------------+--------------------+--------
-------------+
| Education(uni)  | 0.36106346483704976 | 0.28083832335329345 | 0.15
53191489361702 | 0.10869565217391303 | 0.8441894892672095  | 0.19001
751313485116 |
+-----------------+---------------------+---------------------+-----
---------------+--------------------+--------------------+--------
-------------+
+--------+----------+
| Sex(F) | 0.546899 |
+--------+----------+
| Sex(M) | 0.453101 |
+--------+----------+
+------------------+----------------------+--------------------+
| Education        | Education(high)      | Education(uni)      |
+------------------+----------------------+--------------------+
| Occupation(emp)  | 0.9801641586867305   | 0.9653130287648054  |
+------------------+----------------------+--------------------+
| Occupation(self) | 0.019835841313269494 | 0.03468697123519458 |
+------------------+----------------------+--------------------+
+-----------------+----------------------+--------------------+
| Education       | Education(high)      | Education(uni)      |
+-----------------+----------------------+--------------------+
| Resident(big)   | 0.7175102599179206  | 0.9382402707275804  |
+-----------------+----------------------+--------------------+
| Resident(small) | 0.28248974008207933 | 0.06175972927241963 |
+-----------------+----------------------+--------------------+
+------------------+---------------------+--------------------+----
-----------------+--------------------+
| Occupation       | Occupation(emp)     | Occupation(emp)     | Occ
upation(self)    | Occupation(self)    |
+------------------+---------------------+--------------------+----
-----------------+--------------------+
| Resident         | Resident(big)       | Resident(small)     | Res
```

```
ident(big)       | Resident(small)    |
+-----------------+---------------------+--------------------+----
-----------------+--------------------+
| Transport(car)   | 0.7108471892319873  | 0.5465529495380241  | 0.6
855345911949685  | 0.7254901960784315 |
+-----------------+---------------------+--------------------+----
-----------------+--------------------+
| Transport(other) | 0.1388756927949327  | 0.07746979388770434 | 0.1
5723270440251572 | 0.2549019607843137 |
+-----------------+---------------------+--------------------+----
-----------------+--------------------+
| Transport(train) | 0.15027711797307997 | 0.3759772565742715  | 0.1
5723270440251572 | 0.0196078431372549 |
+-----------------+---------------------+--------------------+----
-----------------+--------------------+
equivalent_sample_sizes/iss: 10
+------------+----------+
| Age(adult) | 0.357214 |
+------------+----------+
| Age(old)   | 0.160199 |
+------------+----------+
| Age(young) | 0.482587 |
+------------+----------+
+-----------------+---------------------+--------------------+-------
--------------+----------+---------------------+--------------------
-+
| Age             | Age(adult)          | Age(adult)         | Age(ol
d)            | Age(old) | Age(young)          | Age(young)
|
+-----------------+---------------------+--------------------+-------
--------------+----------+---------------------+--------------------
-+
| Sex             | Sex(F)              | Sex(M)             | Sex(F)
| Sex(M)   | Sex(F)              | Sex(M)              |
+-----------------+---------------------+--------------------+-------
--------------+----------+---------------------+--------------------
-+
| Education(high) | 0.6368243243243243  | 0.716824644549763 | 0.8319
672131147541  | 0.8825   | 0.15808823529411764 | 0.8051724137931034
|
+-----------------+---------------------+--------------------+-------
--------------+----------+---------------------+--------------------
-+
| Education(uni)  | 0.36317567567567566 | 0.283175355450237 | 0.1680
3278688524587 | 0.1175   | 0.8419117647058824  | 0.19482758620689655
|
+-----------------+---------------------+--------------------+-------
--------------+----------+---------------------+--------------------
-+
+--------+----------+
| Sex(F) | 0.546269 |
+--------+----------+
| Sex(M) | 0.453731 |
+--------+----------+
+------------------+----------------------+----------------------+
| Education        | Education(high)      | Education(uni)       |
+------------------+----------------------+----------------------+
| Occupation(emp)  | 0.9743243243243244   | 0.9583333333333334   |
+------------------+----------------------+----------------------+
| Occupation(self) | 0.025675675675675677 | 0.041666666666666664 |
+------------------+----------------------+----------------------+
```

Education	Education(high)	Education(uni)
Resident(big)	0.7148648648648649	0.9316666666666666
Resident(small)	0.2851351351351351	0.06833333333333333

Occupation	Occupation(emp)	Occupation(emp)	Occupation(self)	Occupation(self)
Resident	Resident(big)	Resident(small)	Resident(big)	Resident(small)
Transport(car)	0.7092399873856827	0.5425383542538353	0.6344086021505376	0.5897435897435898
Transport(other)	0.13970356354462313	0.08228730822873083	0.18279569892473116	0.28205128205128205
Transport(train)	0.1510564490696941	0.3751743375174338	0.18279569892473116	0.12820512820512822

equivalent_sample_sizes/iss: 20

Age(adult)	0.356863
Age(old)	0.162745
Age(young)	0.480392

Age	Age(adult)	Age(adult)	Age(old)	Age(old)	Age(young)	Age(young)
Sex	Sex(F)	Sex(M)	Sex(F)	Sex(M)	Sex(F)	Sex(M)
Education(high)	0.6345514950166113	0.7142857142857143	0.8188976377952756	0.8731707317073171	0.16058394160583941	0.7999999999999999
Education(uni)	0.3654485049833887	0.2857142857142857	0.1811023622047245	0.12682926829268293	0.8394160583941606	0.19999999999999998

```
--------------+---------------------+---------------------+---------
------------+
+--------+----------+
| Sex(F) | 0.545588 |
+--------+----------+
| Sex(M) | 0.454412 |
+--------+----------+
+------------------+-----------------+---------------------+
| Education        | Education(high) | Education(uni)      |
+------------------+-----------------+---------------------+
| Occupation(emp)  | 0.968           | 0.9508196721311475  |
+------------------+-----------------+---------------------+
| Occupation(self) | 0.032           | 0.04918032786885246 |
+------------------+-----------------+---------------------+
+-----------------+-----------------+---------------------+
| Education       | Education(high) | Education(uni)      |
+-----------------+-----------------+---------------------+
| Resident(big)   | 0.712           | 0.9245901639344263  |
+-----------------+-----------------+---------------------+
| Resident(small) | 0.288           | 0.07540983606557378 |
+-----------------+-----------------+---------------------+
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Occupation       | Occupation(emp)     | Occupation(emp)     | Occ
upation(self)    | Occupation(self)    |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Resident         | Resident(big)       | Resident(small)     | Res
ident(big)       | Resident(small)     |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(car)   | 0.7074701820464533  | 0.5382513661202186  | 0.5
925925925925926  | 0.5185185185185186  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(other) | 0.1406151914626491  | 0.08743169398907104 | 0.2
0370370370370372 | 0.29629629629629634 |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(train) | 0.15191462649089768 | 0.37431693989071035 | 0.2
0370370370370372 | 0.1851851851851852  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
equivalent_sample_sizes/iss: 100
+------------+----------+
| Age(adult) | 0.354386 |
+------------+----------+
| Age(old)   | 0.180702 |
+------------+----------+
| Age(young) | 0.464912 |
+------------+----------+
+-----------------+---------------------+---------------------+-----
----------------+---------------------+---------------------+--------
------------+
| Age             | Age(adult)          | Age(adult)          | Age 
(old)           | Age(old)            | Age(young)          | Age(yo
ung)        |
+-----------------+---------------------+---------------------+-----
----------------+---------------------+---------------------+--------
------------+
| Sex             | Sex(F)              | Sex(M)              | Sex 
```

```
(F)               | Sex(M)               | Sex(F)               | Sex(M)
|
+-----------------+----------------------+----------------------+-----
----------------+----------------------+---------------------+--------
------------+
| Education(high) | 0.6187683284457478  | 0.69593147751606    | 0.74
25149700598802  | 0.8122448979591838  | 0.1793103448275862 | 0.76417
9104477612  |
+-----------------+----------------------+----------------------+-----
----------------+----------------------+---------------------+--------
------------+
| Education(uni)  | 0.38123167155425225 | 0.30406852248394006 | 0.25
748502994011974 | 0.18775510204081636 | 0.8206896551724138 | 0.23582
08955223881 |
+-----------------+----------------------+----------------------+-----
----------------+----------------------+---------------------+--------
------------+
+--------+----------+
| Sex(F) | 0.540789 |
+--------+----------+
| Sex(M) | 0.459211 |
+--------+----------+
+------------------+----------------------+---------------------+
| Education        | Education(high)     | Education(uni)      |
+------------------+----------------------+---------------------+
| Occupation(emp)  | 0.9228915662650602  | 0.8985507246376812  |
+------------------+----------------------+---------------------+
| Occupation(self) | 0.07710843373493977 | 0.10144927536231885 |
+------------------+----------------------+---------------------+
+-----------------+----------------------+---------------------+
| Education       | Education(high)     | Education(uni)     |
+-----------------+----------------------+---------------------+
| Resident(big)   | 0.691566265060241   | 0.8753623188405797 |
+-----------------+----------------------+---------------------+
| Resident(small) | 0.30843373493975906 | 0.1246376811594203 |
+-----------------+----------------------+---------------------+
+------------------+----------------------+---------------------+----
-----------------+---------------------+
| Occupation       | Occupation(emp)     | Occupation(emp)     | Occ
upation(self)    | Occupation(self)    |
+------------------+----------------------+---------------------+----
-----------------+---------------------+
| Resident         | Resident(big)       | Resident(small)     | Res
ident(big)       | Resident(small)     |
+------------------+----------------------+---------------------+----
-----------------+---------------------+
| Transport(car)   | 0.6938898971566848  | 0.5093896713615023  | 0.4
5614035087719296 | 0.39080459770114945 |
+------------------+----------------------+---------------------+----
-----------------+---------------------+
| Transport(other) | 0.14761040532365396 | 0.12206572769953053 | 0.2
7192982456140347 | 0.3218390804597701  |
+------------------+----------------------+---------------------+----
-----------------+---------------------+
| Transport(train) | 0.15849969751966123 | 0.3685446009389671  | 0.2
7192982456140347 | 0.2873563218390805  |
+------------------+----------------------+---------------------+----
-----------------+---------------------+
equivalent_sample_sizes/iss: 1000
+------------+----------+
| Age(adult) | 0.342972 |
```

Age(old)	0.263454
Age(young)	0.393574

Age	Age(adult)	Age(adult)	Age(old)	Age(old)	Age(young)	Age(young)
Sex	Sex(F)	Sex(M)	Sex(F)	Sex(M)	Sex(F)	Sex(M)
Education(high)	0.5512010113780025	0.5997818974918212	0.5656401944894651	0.6100719424460432	0.3021276595744681	0.6127388535031847
Education(uni)	0.4487989886219975	0.40021810250817885	0.4343598055105348	0.3899280575539569	0.6978723404255319	0.3872611464968153

Sex(F)	0.518675
Sex(M)	0.481325

Education	Education(high)	Education(uni)
Occupation(emp)	0.7028901734104046	0.6729559748427673
Occupation(self)	0.2971098265895954	0.3270440251572327

Education	Education(high)	Education(uni)
Resident(big)	0.5919075144508671	0.6628930817610063
Resident(small)	0.40809248554913297	0.3371069182389937

Occupation	Occupation(emp)	Occupation(emp)	Occupation(self)	Occupation(self)
Resident	Resident(big)	Resident(small)	Resident(big)	Resident(small)

```
| Transport(car)   | 0.5893470790378007  | 0.4014532243415077 | 0.35
107731305449935 | 0.33989501312335957 |
+-----------------+--------------------+-------------------+-----
---------------+--------------------+
| Transport(other) | 0.20146048109965634 | 0.2515894641235241 | 0.32
44613434727503  | 0.3320209973753281  |
+-----------------+--------------------+-------------------+-----
---------------+--------------------+
| Transport(train) | 0.20919243986254293 | 0.3469573115349682 | 0.32
44613434727503  | 0.3280839895013123  |
+-----------------+--------------------+-------------------+-----
---------------+--------------------+
```

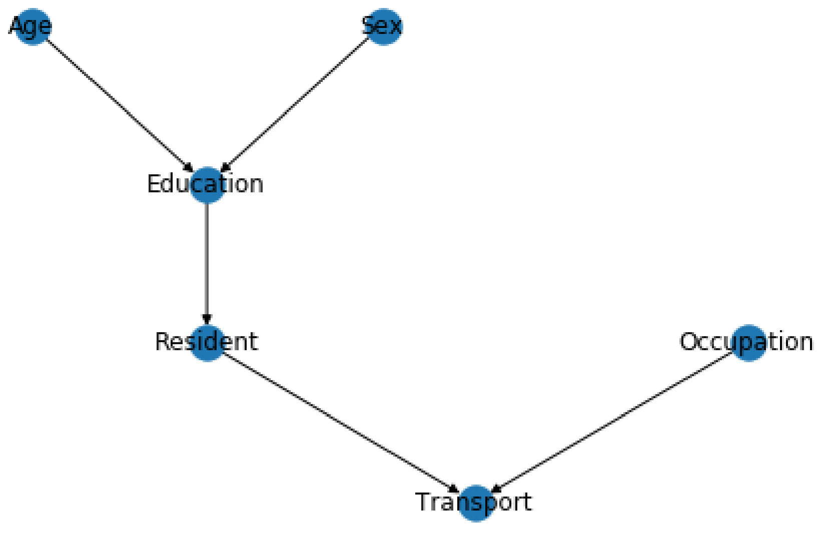
As the equivalent sample size increase, the probabilities of nodes which are children of other nodes move more towards 0.5 and away from the extremes of 0 and 1. There is no change for probabilities of nodes that are not children(Age, Sex).

For Age and Sex, since they are not conditioned on any other nodes, their probabilities can be derived from the data directly and hence not affected by changing the equivalent sample size parameter. For the other nodes which are conditioned on other nodes, the equivalent sample size parameter adds extra 'imaginary' observed instances e.g. for `P(E | A='young', S='F')`, you would count how many E='uni' instances when A='young', S='F', and E='high' instances when A='young', S='F'. The equivalent sample size adds extra counts to both of E='uni' instances when A='young', S='F', and E='high' instances when A='young', S='F', which means you are less likely to get `P(E='uni' | A='young', S='F') = 0` or `P(E='uni' | A='young', S='F') = 1` due to the extra samples. We can see this effect from the above tables where as equivalent sample size increases, the highest probabilities for R/T/E/O decreases and the lowest probabilities increases

4a)

In [14]:

```
transport_model2 = transport_model.copy()
transport_model2.remove_edge('Education', 'Occupation')
pos2 = nx.nx_pydot.graphviz_layout(transport_model2, prog='dot')
nx.draw(transport_model2, pos2, with_labels=True)
```



4b)

In [15]:

```
equivalent_sample_sizes = 1
transport_model2.fit(data, estimator=BayesianEstimator)
transport_model.fit(data, estimator=BayesianEstimator)
nodes = ['Age', 'Education', 'Sex', 'Occupation', 'Resident', 'Transport']

for node in nodes:
    print(node)
    print('original')
    print(transport_model.get_cpds(node))
    print('updated ')
    print(transport_model2.get_cpds(node))
```

```
Age
original
+------------+----------+
| Age(adult) | 0.357393 |
+------------+----------+
| Age(old)   | 0.158897 |
+------------+----------+
| Age(young) | 0.483709 |
+------------+----------+
updated
+------------+----------+
| Age(adult) | 0.357393 |
+------------+----------+
| Age(old)   | 0.158897 |
+------------+----------+
| Age(young) | 0.483709 |
+------------+----------+
Education
original
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Age            | Age(adult)          | Age(adult)         | Age(o
ld)            | Age(old)            | Age(young)          | Age(you
ng)           |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Sex            | Sex(F)              | Sex(M)             | Sex
(F)              | Sex(M)              | Sex(F)              | Sex
(M)              |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Education(high) | 0.6379897785349233  | 0.7181168057210965 | 0.838
9121338912134  | 0.8873417721518988  | 0.15682656826568267 | 0.80782
60869565217  |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Education(uni)  | 0.36201022146507666 | 0.2818831942789034 | 0.161
08786610878664 | 0.11265822784810128 | 0.8431734317343174  | 0.19217
391304347825 |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
updated
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Age            | Age(adult)          | Age(adult)         | Age(o
ld)            | Age(old)            | Age(young)          | Age(you
ng)           |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
-------------+
| Sex            | Sex(F)              | Sex(M)             | Sex
(F)              | Sex(M)              | Sex(F)              | Sex
(M)              |
+----------------+---------------------+--------------------+------
---------------+---------------------+---------------------+--------
```

```
-------------+
| Education(high) | 0.6379897785349233  | 0.7181168057210965 | 0.838
9121338912134  | 0.8873417721518988  | 0.15682656826568267 | 0.80782
60869565217  |
+-----------------+---------------------+--------------------+------
---------------+--------------------+--------------------+--------
-------------+
| Education(uni)  | 0.36201022146507666 | 0.2818831942789034 | 0.161
08786610878664 | 0.11265822784810128 | 0.8431734317343174  | 0.19217
391304347825 |
+-----------------+---------------------+--------------------+------
---------------+--------------------+--------------------+--------
-------------+
Sex
original
+--------+----------+
| Sex(F) | 0.546617 |
+--------+----------+
| Sex(M) | 0.453383 |
+--------+----------+
updated
+--------+----------+
| Sex(F) | 0.546617 |
+--------+----------+
| Sex(M) | 0.453383 |
+--------+----------+
Occupation
original
+------------------+----------------------+----------------------+
| Education        | Education(high)      | Education(uni)       |
+------------------+----------------------+----------------------+
| Occupation(emp)  | 0.9775510204081632   | 0.9621848739495799   |
+------------------+----------------------+----------------------+
| Occupation(self) | 0.022448979591836733 | 0.037815126050420166 |
+------------------+----------------------+----------------------+
updated
+------------------+-----------+
| Occupation(emp)  | 0.970677  |
+------------------+-----------+
| Occupation(self) | 0.0293233 |
+------------------+-----------+
Resident
original
+-----------------+--------------------+---------------------+
| Education       | Education(high)    | Education(uni)      |
+-----------------+--------------------+---------------------+
| Resident(big)   | 0.7163265306122449 | 0.9352941176470588  |
+-----------------+--------------------+---------------------+
| Resident(small) | 0.2836734693877551 | 0.06470588235294118 |
+-----------------+--------------------+---------------------+
updated
+-----------------+--------------------+---------------------+
| Education       | Education(high)    | Education(uni)      |
+-----------------+--------------------+---------------------+
| Resident(big)   | 0.7163265306122449 | 0.9352941176470588  |
+-----------------+--------------------+---------------------+
| Resident(small) | 0.2836734693877551 | 0.06470588235294118 |
+-----------------+--------------------+---------------------+
Transport
original
+------------------+---------------------+---------------------+----
```

```
-----------------+---------------------+
| Occupation       | Occupation(emp)     | Occupation(emp)     | Occ
upation(self)    | Occupation(self)    |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Resident         | Resident(big)       | Resident(small)     | Res
ident(big)       | Resident(small)     |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(car)   | 0.7101311838153944  | 0.5447498238195913  | 0.6
608187134502924  | 0.6507936507936508  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(other) | 0.13924450766556032 | 0.07963354474982381 | 0.1
6959064327485382 | 0.2698412698412699  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(train) | 0.15062430851904537 | 0.3756166314305849  | 0.1
6959064327485382 | 0.07936507936507937 |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
updated
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Occupation       | Occupation(emp)     | Occupation(emp)     | Occ
upation(self)    | Occupation(self)    |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Resident         | Resident(big)       | Resident(small)     | Res
ident(big)       | Resident(small)     |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(car)   | 0.7101311838153944  | 0.5447498238195913  | 0.6
608187134502924  | 0.6507936507936508  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(other) | 0.13924450766556032 | 0.07963354474982381 | 0.1
6959064327485382 | 0.2698412698412699  |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
| Transport(train) | 0.15062430851904537 | 0.3756166314305849  | 0.1
6959064327485382 | 0.07936507936507937 |
+------------------+---------------------+---------------------+----
-----------------+---------------------+
```

The local distributions are all unchanged except for that of the 'Occupation' node

Original factorisation is: P(A, S, E, O, R, T) = P(A)P(S)P(E|A, S)P(R|E)P(O|E)P(T| R, O)

New factorisation is: P(A, S, E, O, R, T) = P(A)P(S)P(E|A, S)P(R|E)P(O)P(T| R, O)

So the 'O' node had it's distribution changed from one that is dependent on E to one that isn't. As expected, the distributions on the other nodes hasn't changed as we can see that the factorisation hasn't changed.

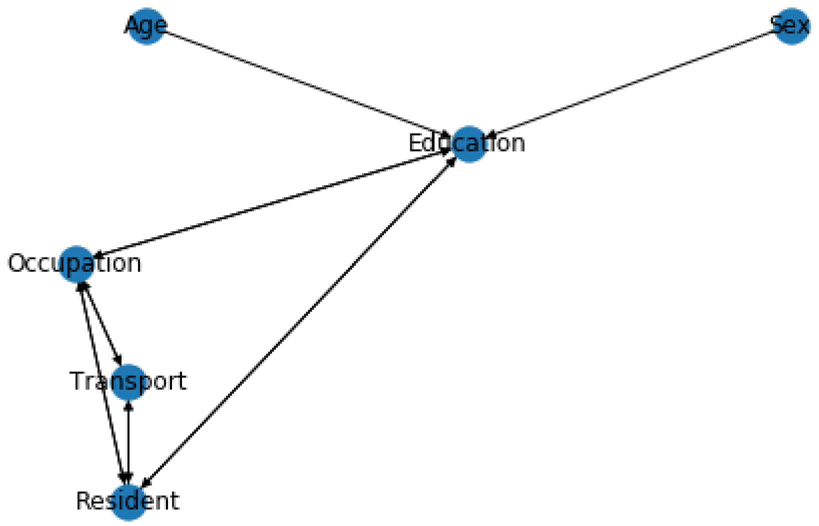
5a)

In [16]:

```
D1 = transport_model.copy()
p1 = ConstraintBasedEstimator.model_to_pdag(D1)
```

In [17]:

```
pos3 = nx.nx_pydot.graphviz_layout(p1, prog='dot')
nx.draw(p1,pos3, with_labels=True)
plt.show()
```



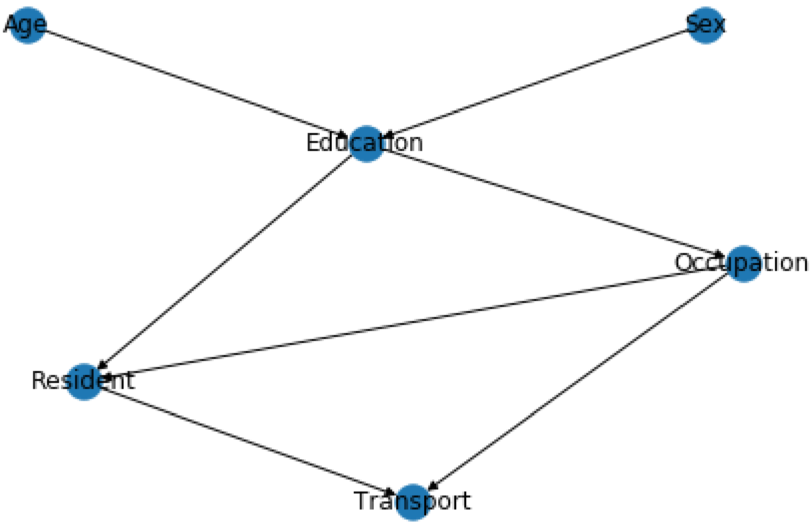
P1 and D1 have the same node/edge combinations. This is because P1 represents the set of equivalent dags of D1, and hence P1 and D1 would have the same skeleton (undirected graph). Any vstructs in D1 should also be preserved in P1 (otherwise it is no longer equivalent as the conditional dependence will be the same)

D1 is more or less all vstructs so there is no difference between D1 and P1 (or at least not in principal, in pgmpy modelling, there is an extra edge between R and O which is rather unexpected and shouldn't be correct...)

5b)

In [18]:

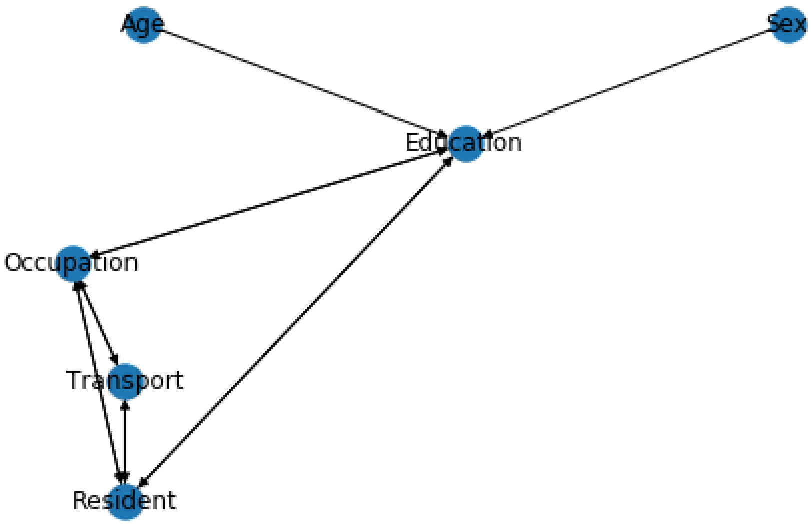
```
D2 = D1.copy()
D2.add_edge('Occupation', 'Resident')
nx.draw(D2, nx.nx_pydot.graphviz_layout(D2, prog='dot'), with_labels=True)
```



5c)

In [19]:

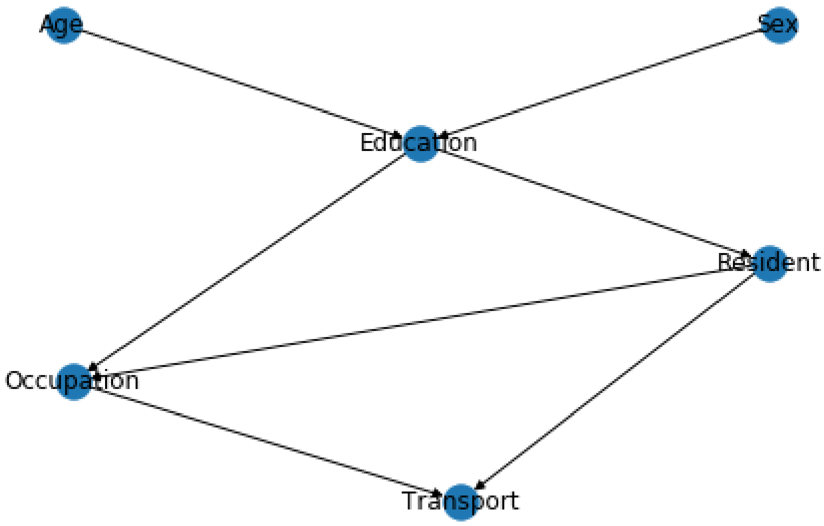
```
P2 = ConstraintBasedEstimator.model_to_pdag(D2)
nx.draw(P2, nx.nx_pydot.graphviz_layout(P2, prog='dot'), with_labels=True)
```



5d)

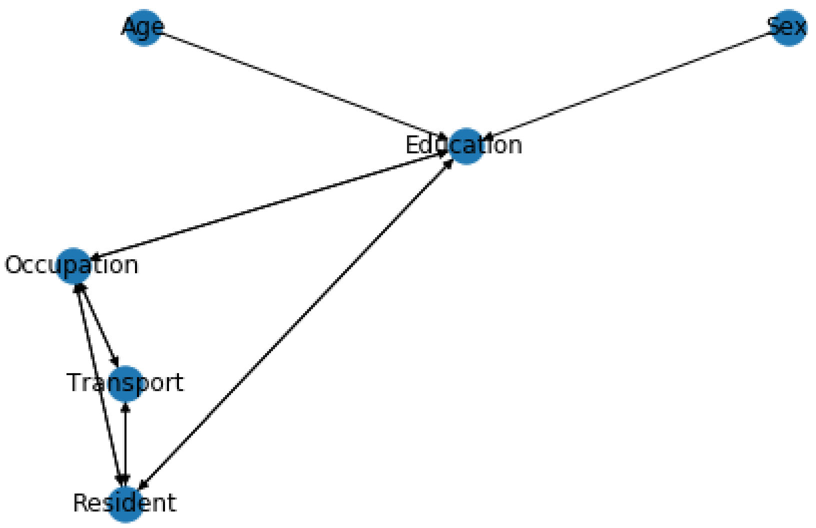
In [32]:

```
D3= D1.copy()
D3.add_edge('Resident','Occupation')
nx.draw(D3, nx.nx_pydot.graphviz_layout(D3, prog='dot'), with_labels=True)
```



In [21]:

```
P3 = ConstraintBasedEstimator.model_to_pdag(D3)
nx.draw(P3, nx.nx_pydot.graphviz_layout(P3, prog='dot'), with_labels=True)
```



5e) The loglikelihood score doesn't seem to be availabe in pgmpy. Here I used another method-- BicScore instead (Bayesian information criterion, which is a modified version of the likelihood score but with a penalty added to stop overfitting) . D2 and D3 share the same score-- this is expected because the score is dependent on the underlying PDags and D2 and D3 have the same dags

In [22]:

```
BicScore(data).score(D1), BicScore(data).score(D2),BicScore(data).score(D3)
```

Out[22]:

```
(-2420.007975311849, -2425.3464677738216, -2425.3464677738216)
```

In [ ]:

```

```

6a)

In [ ]:

```

```

In [23]:

```
def pyro_model():
    A_alias = ['young','adult','old']
    S_alias = ['M','F']
    E_alias = ['high','uni']
    O_alias = ['emp','self']
    R_alias = ['small','big']
    T_alias = ['car','train','other']

    age_probs = torch.tensor([0.48, 0.36, 0.16])
    sex_probs = torch.tensor([0.45, 0.55 ])

    occupation_probs = torch.tensor([
            [0.98, 0.97],
            [0.02, 0.03]
    ])
    education_probs = torch.tensor([
            [[0.64, 0.36], [0.84, 0.16], [0.16, 0.84]],
            [[0.72, 0.28], [0.89, 0.11], [0.81, 0.19]]])

    resident_probs = torch.tensor([
            [0.72, 0.94],
            [0.28, 0.06]])
    transport_probs = torch.tensor([[[0.54,0.38,0.07], [0.71, 0.15, 0.14]],
                     [[0.73,0.02,0.25], [0.69,0.18,0.18]]])

    age = pyro.sample("Age", pyro.distributions.Categorical(probs=age_probs)) #y
oung, adult, old
    sex = pyro.sample("Sex",  pyro.distributions.Categorical(probs=sex_probs)) #
male, female


    education = pyro.sample("Education", pyro.distributions.Categorical(probs=ed
ucation_probs[sex][age]))


    occupation  = pyro.sample("Occupation", pyro.distributions.Categorical(occup
ation_probs[education]))
     # big ,small
    resident  = pyro.sample("Resident", pyro.distributions.Categorical(resident_
probs[education]))


    transport = pyro.sample("Transport", pyro.distributions.Categorical(transpor
t_probs[occupation][resident]))

    return{'age': age,
           'sex': sex,
           'education': education,
           'occupation': occupation,
           'resident': resident,
           'transport': transport
          }
```

In [24]:

```
pyro_model()
```

Out[24]:

```
{'age': tensor(0),
 'sex': tensor(0),
 'education': tensor(1),
 'occupation': tensor(0),
 'resident': tensor(0),
 'transport': tensor(0)}
```

6b) You observe a person with a university degree. What is your prediction of this person's means of travel?

so we know eductions="uni"

In [25]:

```
conditioned_model_uni = pyro.condition(pyro_model, data={"Education": torch.tens
or(1), })
```

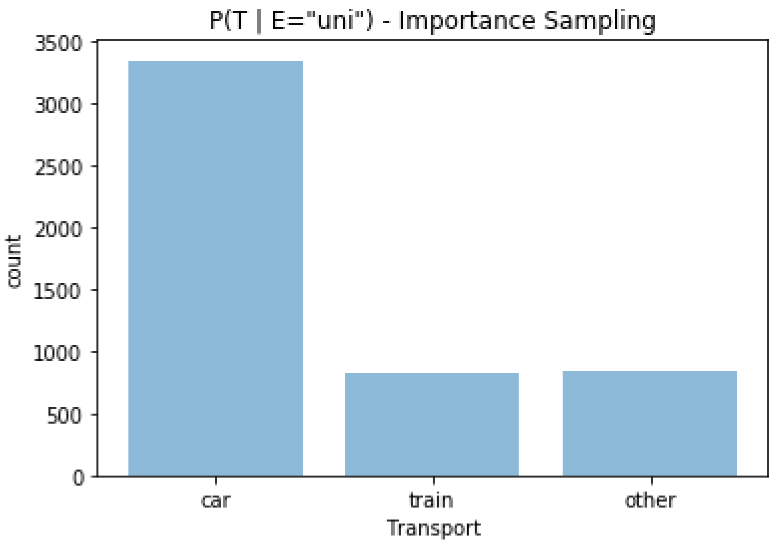
In [26]:

```
t_posterior = pyro.infer.Importance(conditioned_model_uni, num_samples=5000).run
()
t_marginal = pyro.infer.EmpiricalMarginal(t_posterior,"Transport")
t_samples = [t_marginal().item() for _ in range(5000)]
t_unique, t_counts = np.unique(t_samples, return_counts=True)

plt.bar(t_unique, t_counts, align='center', alpha=0.5)
plt.xticks(t_unique, ['car','train','other'])
plt.ylabel('count')
plt.xlabel('Transport')
plt.title('P(T | E="uni") - Importance Sampling')
```

Out[26]:

```
Text(0.5, 1.0, 'P(T | E="uni") - Importance Sampling')
```



6c) You observe a self-employed person who lives in a big city. What is your prediction of this person's age? Provide either a MAP estimate or a histogram of the marginal on the variable "A".

In [27]:

```
conditioned_model_emp_resi = pyro.condition(pyro_model,
    data={"Resident": torch.tensor(1), "Occupation": torch.tensor(0) })
```

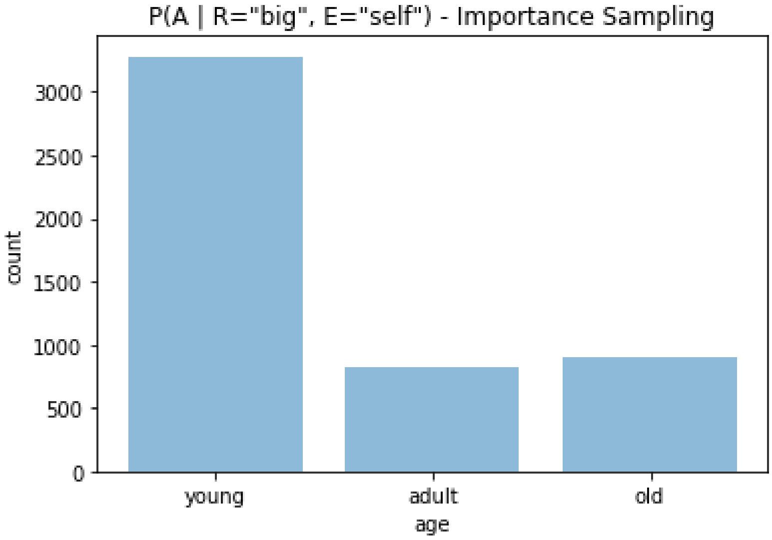
In [28]:

```
a_posterior = pyro.infer.Importance(conditioned_model_emp_resi, num_samples=5000
).run()
a_marginal = pyro.infer.EmpiricalMarginal(a_posterior,"Age")
a_samples = [t_marginal().item() for _ in range(5000)]
a_unique, a_counts = np.unique(a_samples, return_counts=True)

plt.bar(a_unique, a_counts, align='center', alpha=0.5)
plt.xticks(a_unique, ['young','adult','old'])
plt.ylabel('count')
plt.xlabel('age')
plt.title('P(A | R="big", E="self") - Importance Sampling')
```

Out[28]:

```
Text(0.5, 1.0, 'P(A | R="big", E="self") - Importance Sampling')
```



In [29]:

```
c = ConstraintBasedEstimator(data)
pdag = c.skeleton_to_pdag(*c.estimate_skeleton())
pdag.edges() # edges: A->C, B->C, A--D (not directed)
```

Out[29]:

```
OutEdgeView([('Age', 'Education'), ('Age', 'Sex'), ('Resident', 'Tra
nsport'), ('Education', 'Sex'), ('Sex', 'Age'), ('Sex', 'Educatio
n')])
```

In [ ]: