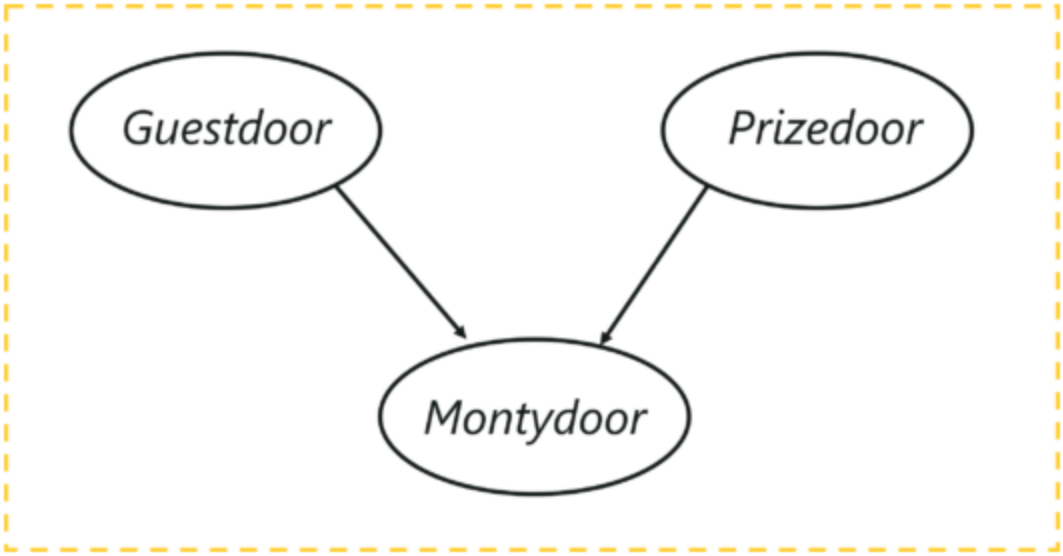
In [75]:
```
# **Bayesian Networks Python**
# In this demo, we'll be using Bayesian Networks to solve the famous Monty Hall Pro
# The Monty Hall problem named after the host of the TV series, 'Let's Make A Deal'
#for over a decade.
# So this is how it works. The game involves three doors, given that behind one of
# So you start by picking a random door, say #2. On the other hand, the host knows
# (behind which there is a goat). Here's the catch, you're now given a choice, the
# choice i.e. #2.
```



In [76]:
```
# What is the use of pgmpy?
# pgmpy is a python framework to work with these types of graph models. Several gra
#Pgmpy also allows users to create their own inference algorithm without getting in
```

In [77]:
```
pip install pgmpy
```

```
Requirement already satisfied: pgmpy in c:\users\student\appdata\local\programs\pyth
on\python311\lib\site-packages (0.1.23)
Requirement already satisfied: networkx in c:\users\student\appdata\local\programs\p
ython\python311\lib\site-packages (from pgmpy) (3.1)
Requirement already satisfied: numpy in c:\users\student\appdata\local\programs\pyth
on\python311\lib\site-packages (from pgmpy) (1.24.4)
Requirement already satisfied: scipy in c:\users\student\appdata\local\programs\pyth
on\python311\lib\site-packages (from pgmpy) (1.11.1)
Requirement already satisfied: scikit-learn in c:\users\student\appdata\local\progra
ms\python\python311\lib\site-packages (from pgmpy) (1.3.0)
Requirement already satisfied: pandas in c:\users\student\appdata\local\programs\pyt
hon\python311\lib\site-packages (from pgmpy) (2.0.3)
Requirement already satisfied: pyparsing in c:\users\student\appdata\local\programs
\python\python311\lib\site-packages (from pgmpy) (3.0.9)
Requirement already satisfied: torch in c:\users\student\appdata\local\programs\pyth
on\python311\lib\site-packages (from pgmpy) (2.0.1)
Requirement already satisfied: statsmodels in c:\users\student\appdata\local\program
s\python\python311\lib\site-packages (from pgmpy) (0.14.0)
Requirement already satisfied: tqdm in c:\users\student\appdata\local\programs\pytho
n\python311\lib\site-packages (from pgmpy) (4.65.0)
Requirement already satisfied: joblib in c:\users\student\appdata\local\programs\pyt
hon\python311\lib\site-packages (from pgmpy) (1.3.1)
Requirement already satisfied: opt-einsum in c:\users\student\appdata\local\programs
\python\python311\lib\site-packages (from pgmpy) (3.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\student\appdata\lo
cal\programs\python\python311\lib\site-packages (from pandas->pgmpy) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\student\appdata\local\progra
ms\python\python311\lib\site-packages (from pandas->pgmpy) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\student\appdata\local\prog
rams\python\python311\lib\site-packages (from pandas->pgmpy) (2023.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\student\appdata\loca
l\programs\python\python311\lib\site-packages (from scikit-learn->pgmpy) (3.2.0)
Requirement already satisfied: patsy>=0.5.2 in c:\users\student\appdata\local\progra
ms\python\python311\lib\site-packages (from statsmodels->pgmpy) (0.5.3)
Requirement already satisfied: packaging>=21.3 in c:\users\student\appdata\local\pro
grams\python\python311\lib\site-packages (from statsmodels->pgmpy) (23.1)
Requirement already satisfied: filelock in c:\users\student\appdata\local\programs\p
ython\python311\lib\site-packages (from torch->pgmpy) (3.12.2)
Requirement already satisfied: typing-extensions in c:\users\student\appdata\local\p
rograms\python\python311\lib\site-packages (from torch->pgmpy) (4.7.1)
Requirement already satisfied: sympy in c:\users\student\appdata\local\programs\pyth
on\python311\lib\site-packages (from torch->pgmpy) (1.12)
Requirement already satisfied: jinja2 in c:\users\student\appdata\local\programs\pyt
hon\python311\lib\site-packages (from torch->pgmpy) (3.1.2)
Requirement already satisfied: colorama in c:\users\student\appdata\local\programs\p
ython\python311\lib\site-packages (from tqdm->pgmpy) (0.4.6)
Requirement already satisfied: six in c:\users\student\appdata\local\programs\python
\python311\lib\site-packages (from patsy>=0.5.2->statsmodels->pgmpy) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\student\appdata\local\pro
grams\python\python311\lib\site-packages (from jinja2->torch->pgmpy) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in c:\users\student\appdata\local\progra
ms\python\python311\lib\site-packages (from sympy->torch->pgmpy) (1.3.0)
Note: you may need to restart the kernel to use updated packages.
```

In [63]:
```
from pgmpy.models import BayesianNetwork
from pgmpy.factors.discrete import TabularCPD
```

```
import networkx as nx
import pylab as plt
```

In [64]:
```
# Defining Bayesian Structure
model = BayesianNetwork([('Guest', 'Host'), ('Price', 'Host')])
```

In [65]:
```
import numpy as np
import warnings

# Suppress DeprecationWarnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

# Define the CPDs
cpd_guest = TabularCPD('Guest', 3, [[0.33], [0.33], [0.33]])
cpd_price = TabularCPD('Price', 3, [[0.33], [0.33], [0.33]])
cpd_host = TabularCPD('Host', 3, [
    [0, 0, 0, 0, 0.5, 1, 0, 1, 0.5],
    [0.5, 0, 1, 0, 0, 0, 1, 0, 0.5],
    [0.5, 1, 0, 1, 0.5, 0, 0, 0, 0]
], evidence=['Guest', 'Price'], evidence_card=[3, 3])

# Associating the CPDs with the network structure.
model.add_cpds(cpd_guest, cpd_price, cpd_host)
```

In [66]:
```
# Associating the CPDs with the network structure.
model.add_cpds(cpd_guest, cpd_price, cpd_host)
```

In [67]:
```
model.check_model()
```

Out[67]: True

In [82]:
```
# Infering the posterior probability
from pgmpy.inference import VariableElimination

infer = VariableElimination(model)
posterior_p = infer.query(['Host'], evidence={'Guest': 2, 'Price': 2})
print(posterior_p)
```

```
+---------+-------------+
| Host    |   phi(Host) |
+=========+=============+
| Host(0) |      0.5000 |
+---------+-------------+
| Host(1) |      0.5000 |
+---------+-------------+
| Host(2) |      0.0000 |
+---------+-------------+
```

In [69]:
```
# Infering the posterior probability
from pgmpy.inference import VariableElimination

infer = VariableElimination(model)
posterior_p = infer.query(['Host'], evidence={'Guest': 1, 'Price': 2})
print(posterior_p)
```

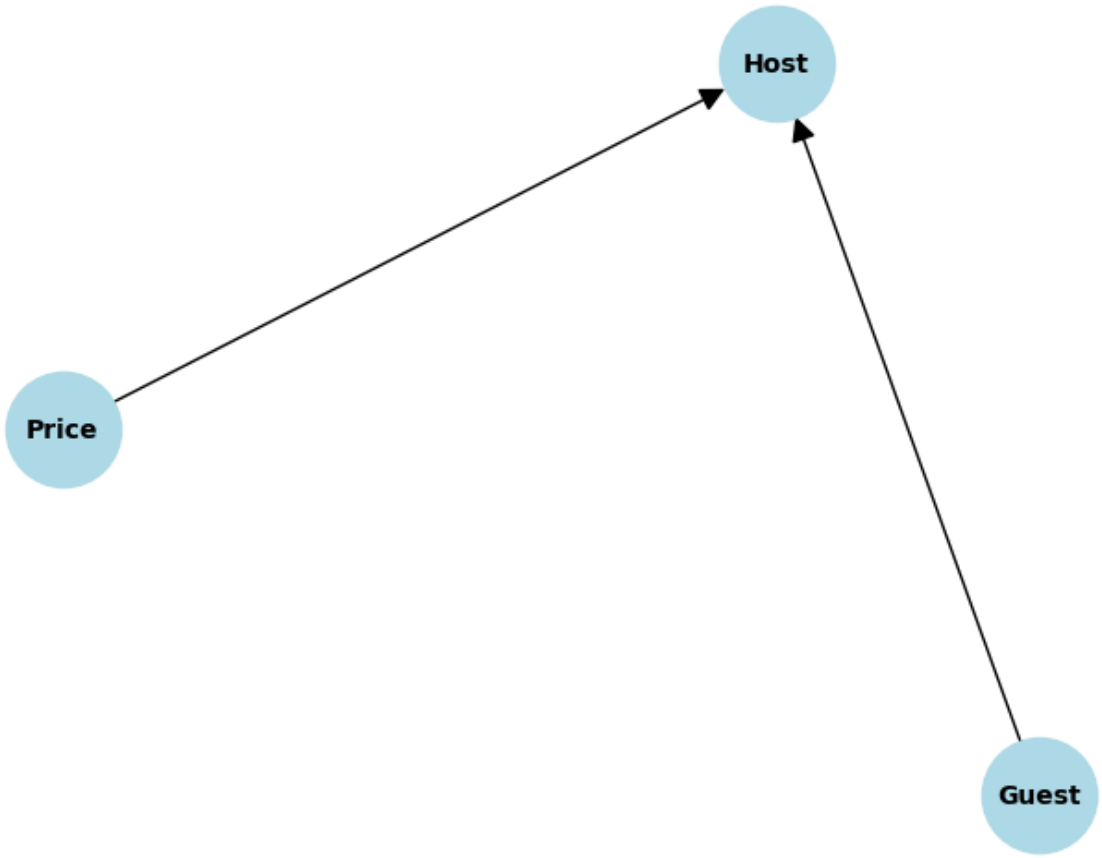
```
+---------+-------------+
| Host    |   phi(Host) |
+=========+=============+
| Host(0) |      1.0000 |
+---------+-------------+
| Host(1) |      0.0000 |
+---------+-------------+
| Host(2) |      0.0000 |
+---------+-------------+
```

In [74]:
```python
import networkx as nx
import matplotlib.pyplot as plt
from pgmpy.models import BayesianModel

# Assuming you have already defined and created the 'model' variable as a BayesianM
# Convert the Bayesian model to a networkx graph
graph = nx.DiGraph(model.edges())

# Draw the graph using NetworkX
nx.draw(graph, with_labels=True, node_color='lightblue', node_size=2000, font_size=

# Show the plot
plt.show()
```



In [ ]:
```python
```