import numpy

from pomegranate import \*

guest = DiscreteDistribution({'A': 1./3, 'B': 1./3, 'C': 1./3})

prize = DiscreteDistribution({'A': 1./3, 'B': 1./3, 'C': 1./3})

monty = ConditionalProbabilityTable(

[[ 'A', 'A', 'A', 0.0 ],

[ 'A', 'A', 'B', 0.5 ],

[ 'A', 'A', 'C', 0.5 ],

[ 'A', 'B', 'A', 0.0 ],

[ 'A', 'B', 'B', 0.0 ],

[ 'A', 'B', 'C', 1.0 ],

[ 'A', 'C', 'A', 0.0 ],

[ 'A', 'C', 'B', 1.0 ],

[ 'A', 'C', 'C', 0.0 ],

[ 'B', 'A', 'A', 0.0 ],

[ 'B', 'A', 'B', 0.0 ],

[ 'B', 'A', 'C', 1.0 ],

[ 'B', 'B', 'A', 0.5 ],

[ 'B', 'B', 'B', 0.0 ],

[ 'B', 'B', 'C', 0.5 ],

[ 'B', 'C', 'A', 1.0 ],

[ 'B', 'C', 'B', 0.0 ],

[ 'B', 'C', 'C', 0.0 ],

[ 'C', 'A', 'A', 0.0 ],

[ 'C', 'A', 'B', 1.0 ],

[ 'C', 'A', 'C', 0.0 ],

[ 'C', 'B', 'A', 1.0 ],

[ 'C', 'B', 'B', 0.0 ],

[ 'C', 'B', 'C', 0.0 ],

[ 'C', 'C', 'A', 0.5 ],

[ 'C', 'C', 'B', 0.5 ],

[ 'C', 'C', 'C', 0.0 ]], [guest, prize])

s1 = State(guest, name="guest")

s2 = State(prize, name="prize")

s3 = State(monty, name="monty")

model = BayesianNetwork("Monty Hall Problem")

model.add\_states(s1, s2, s3)

model.add\_edge(s1, s3)

model.add\_edge(s2, s3)

model.bake()

print(model.probability([['A', 'B', 'C'],['A','A','C'],['A','A','A']]))

print(model.predict([['A',None,'C'],['A','A',None],['A','B',None]])