**Lab Exercise 9**

**UNCERTAIN KNOWLEDGE REPRESENTATION**

**Title:** UNCERTAINTY KNOWLEDGE REPRESENTATION IN MONTY HALL PROBLEM

Problem Description:

Let us think about the game show (Let’s Make a Deal) from the shows point of view, the show and the shows host (Monty) knows where there is a goat and where there isn’t a goat and therefore knows which door the prize is behind. So, we are given three doors (Door #1, Door #2, and Door#3). Behind the first door is a goat, behind the second door is a car (our prize) and behind the third door is another goat.

Solution:

**Suppose the contestant chooses Door #1**, then the host reveals Door #3 (the goat), since the host knows Door #2 contains the prize car. In this case it is good for the contestant to switch. If the contestant/player switches then the contestant would win the prize. If the contestant/player did NOT switch then the contestant would lose the prize.

Suppose the contestant chooses Door #2, then the host reveals either Door #1 or Door #3. In this case it is bad to switch for the contestant. If the contestant/player switches then the contestant would lose the prize. If the contestant/player did NOT switch then the contestant would win the prize.

Suppose the contestant chooses Door #3, then the host reveals Door #1. In this case it is good for the contestant to switch. If the contestant/player switches then the contestant would win the prize. If the contestant/player did NOT switch then the contestant would lose the prize.

In the above scenario, we can easily see that if the contestant had switched , then the contestant would’ve won 2/3 or about 66.66% of the time and lost only 1/3 or about 33.33% of the time. If the contestant chose not switch, then the contestant would’ve won 1/3 or about 33.33% of the time and lost 2/3 or about 66.66% of the time

**Scenario #1 for always switching:**  
1. You choose door #1  
2.The host opens door #3 , because the prize is behind door #2, and asks if you want to switch.  
3.You decide to switch to door #2  
4.**YOU WIN!**

**Scenario #2 for always switching**  
1.You choose door #2  
2.The host opens door #3 or door #1 , because the prize is behind door # 2, and asks if you want to switch.  
3.You decide to switch to door #1 or door #3  
4.**YOU LOSE.**

**Scenario #3 for always switching**  
1. You choose door #3  
2. The host opens door #1 , because the prize is behind door #2, and asks if you want to switch.  
3. You decide to switch to door #2  
4. **YOU WIN!**

**Python Code:**

#Import the libraries

import random

import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')

#The host will reveal a door that doesn't contain the prize

#Remember the host knows the door that the prize is behind

def get\_non\_prize\_door(host, num\_doors, player\_choice):

i = 1

while (i == host or i== player\_choice ):

i = (i+1)%(num\_doors)

return i

#Have the player switch to the other unopened door

def switch\_function(shown\_door, num\_doors, player\_choice):

i = 1

while (i == shown\_door or i== player\_choice ):

i = (i+1)%(num\_doors)

return i

#Play the game

def monty\_hall\_game(switch, num\_tests):

win\_switch\_cnt = 0

win\_no\_switch\_cnt = 0

lose\_switch\_cnt = 0

lose\_no\_switch\_cnt = 0

doors = [0,1,2] #Get the doors

num\_doors = len(doors) #Get the number of doors

for i in range(0,num\_tests):

door\_with\_prize = random.randint(0, num\_doors-1) #Randomly choose the door with the wanted prize

host = door\_with\_prize #The host knows which door has the prize

#The player chooses initially a random door that s/he believes has the prize

player\_choice = random.randint(0, num\_doors-1)

original\_player\_choice = player\_choice

shown\_door = get\_non\_prize\_door(host, num\_doors, player\_choice)

if switch == True:

player\_choice = switch\_function(shown\_door,num\_doors, player\_choice)

if player\_choice == host and switch == False:

#Then the player wins from not switching

print('Player Wins (No switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice ,', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

win\_no\_switch\_cnt = win\_no\_switch\_cnt + 1

elif player\_choice == host and switch == True:

#Then the player wins from switching

print('Player Wins (switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

win\_switch\_cnt = win\_switch\_cnt +1

elif player\_choice != host and switch == False:

#The player lost from not switching

print('Player Lost (No switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

lose\_no\_switch\_cnt = lose\_no\_switch\_cnt + 1

elif player\_choice != host and switch == True:

#The player lost from switching

print('Player Lost (switch) - The player chose door: ', player\_choice,' Original choice: ',original\_player\_choice , ', Door with prize:', door\_with\_prize, ', Shown Door: ',shown\_door )

lose\_switch\_cnt = lose\_switch\_cnt + 1

else:

print('SOMETHING IS WRONG')

return win\_no\_switch\_cnt,win\_switch\_cnt,lose\_no\_switch\_cnt,lose\_switch\_cnt, num\_tests

#Play the game

x = monty\_hall\_game(True, 10)

print('Win switch %: ', x[1]/ x[4])

print('Lose switch %: ', x[3]/ x[4])

print('Win No switch %: ', x[0]/ x[4])

print('Lose No switch %: ', x[2]/ x[4])

# Output:

# 

# Result: Thus the uncertain knowledge representation is applied and verified.

**NAME :- HARIT**

**Reg No :- RA1811030010037**

**Sec :- L1**