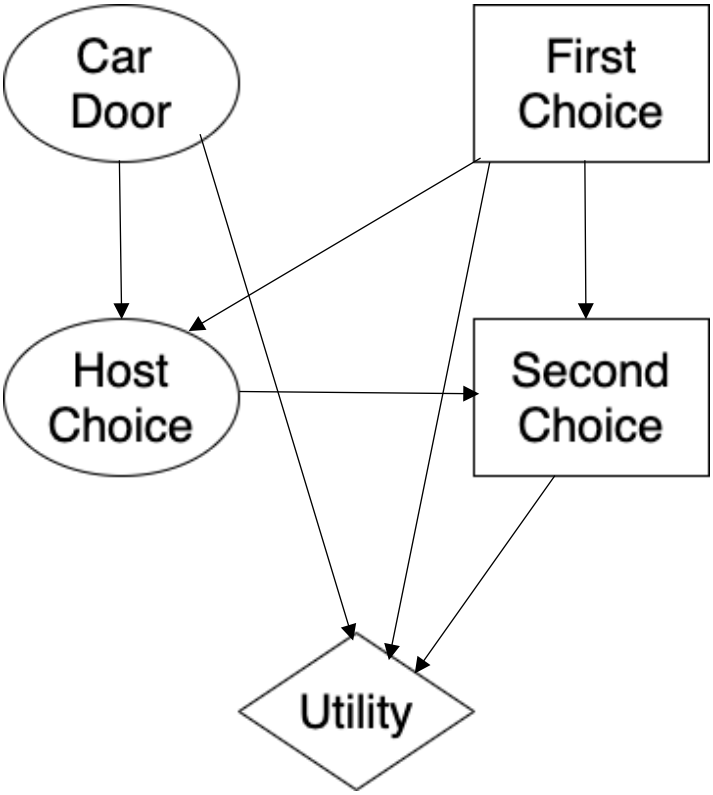
20726636 An Nguyen

1.a)



Car door	Prob
1	p1
2	p2
3	1-p1-p2

FirstChoice	CarDoor	HostChoice	P(HostChoice | FirstChoice ^ CarDoor)
1	1	Smaller	0.5
1	1	Bigger	0.5
1	2	Smaller	0
1	2	Bigger	1
1	3	Smaller	1
1	3	Bigger	0
2	1	Smaller	0
2	1	Bigger	1
2	2	Smaller	0.5
2	2	Bigger	0.5
2	3	Smaller	1
2	3	Bigger	0
3	1	Smaller	0
3	1	Bigger	1
3	2	Smaller	1
3	2	Bigger	0

3	3	Smaller	0.5
3	3	Bigger	0.5

CarDoor	FirstChoice	SecondChoice	Utility
1	1	Stay	1
1	1	Switch	0
1	2	Stay	0
1	2	Switch	1
1	3	Stay	0
1	3	Switch	1
2	1	Stay	0
2	1	Switch	1
2	2	Stay	1
2	2	Switch	0
2	3	Stay	0
2	3	Switch	1
3	1	Stay	0
3	1	Switch	1
3	2	Stay	0
3	2	Switch	1
3	3	Stay	1
3	3	Switch	0

b) Let the tables above represent the factor tables f1, f2, f3 respectively such that p1 = 1/3, p2 = 1/3.

1. Multipy f1 and f2 to get f4

CarDoor	FirstChoice	SecondChoice	HostChoice	Value
1	1	Stay	Smaller	0.167
1	2	Stay	Smaller	0
1	3	Stay	Smaller	0
1	1	Switch	Smaller	0
1	2	Switch	Smaller	0
1	3	Switch	Smaller	0
1	1	Stay	Bigger	0.167
1	2	Stay	Bigger	0
1	3	Stay	Bigger	0
1	1	Switch	Bigger	0
1	2	Switch	Bigger	0.333
1	3	Switch	Bigger	0.333
2	1	Stay	Smaller	0
2	2	Stay	Smaller	0.167
2	3	Stay	Smaller	0
2	1	Switch	Smaller	0
2	2	Switch	Smaller	0

2	3	Switch	Smaller	0.333
2	1	Stay	Bigger	0
2	2	Stay	Bigger	0.167
2	3	Stay	Bigger	0
2	1	Switch	Bigger	0.333
2	2	Switch	Bigger	0
2	3	Switch	Bigger	0
3	1	Stay	Smaller	0
3	2	Stay	Smaller	0
3	3	Stay	Smaller	0.167
3	1	Switch	Smaller	0.333
3	2	Switch	Smaller	0.333
3	3	Switch	Smaller	0
3	1	Stay	Bigger	0
3	2	Stay	Bigger	0
3	3	Stay	Bigger	0.167
3	1	Switch	Bigger	0
3	2	Switch	Bigger	0
3	3	Switch	Bigger	0

2. Sum out CarDoor since it is not apparent of any of the decision nodes to get f5

FirstChoice	**SecondChoice**	**HostChoice**	**Value**
1	Stay	Smaller	0.167
2	Stay	Smaller	0.167
3	Stay	Smaller	0.167
1	Switch	Smaller	0.333
2	Switch	Smaller	0.333
3	Switch	Smaller	0.333
1	Stay	Bigger	0.167
2	Stay	Bigger	0.167
3	Stay	Bigger	0.167
1	Switch	Bigger	0.333
2	Switch	Bigger	0.333
3	Switch	Bigger	0.333

3. Sum out SecondChoice by restricting it in f5 to produce f6. Optimal policy for SecondChoice is Switch

FirstChoice	**HostChoice**	**Value**
1	Smaller	0.333
1	Bigger	0.333
2	Smaller	0.333
2	Bigger	0.333
3	Smaller	0.333
3	Bigger	0.333

4. Sum out HostChoice from f6 to produce f7

FirstChoice	Value
1	0.667
2	0.667
3	0.667

5. Restrict f7 to get optimal policy for FirstChoice

Value
0.667

6. Sum out the result to get expected utility of 0.667. Optimal policy for FirstChoice is to pick any door since they all have the same value. Optimal policy for SecondChoice is to switch. Expected utility of the optimal policy is 0.667.

c)

Car door	Prob
1	0.6
2	0.25
3	0.15

Let the table above represent the factor table f1. f2 and f3 are the same as in part 2.

1. Multipy f1 and f2 to get f4

CarDoor	FirstChoice	SecondChoice	HostChoice	Value
1	1	Stay	Smaller	0.3
1	2	Stay	Smaller	0
1	3	Stay	Smaller	0
1	1	Switch	Smaller	0
1	2	Switch	Smaller	0
1	3	Switch	Smaller	0
1	1	Stay	Bigger	0.3
1	2	Stay	Bigger	0
1	3	Stay	Bigger	0

1	1	Switch	Bigger	0
1	2	Switch	Bigger	0.6
1	3	Switch	Bigger	0.6
2	1	Stay	Smaller	0
2	2	Stay	Smaller	0.125
2	3	Stay	Smaller	0
2	1	Switch	Smaller	0
2	2	Switch	Smaller	0
2	3	Switch	Smaller	0.25
2	1	Stay	Bigger	0
2	2	Stay	Bigger	0.125
2	3	Stay	Bigger	0
2	1	Switch	Bigger	0.25
2	2	Switch	Bigger	0
2	3	Switch	Bigger	0
3	1	Stay	Smaller	0
3	2	Stay	Smaller	0
3	3	Stay	Smaller	0.075
3	1	Switch	Smaller	0.15
3	2	Switch	Smaller	0.15
3	3	Switch	Smaller	0
3	1	Stay	Bigger	0
3	2	Stay	Bigger	0
3	3	Stay	Bigger	0.075
3	1	Switch	Bigger	0
3	2	Switch	Bigger	0
3	3	Switch	Bigger	0

2. Sum out CarDoor since it is not apparent of any of the decision nodes to get f5

FirstChoice	**SecondChoice**	**HostChoice**	**Value**
1	Stay	Smaller	0.3
2	Stay	Smaller	0.125
3	Stay	Smaller	0.075
1	Switch	Smaller	0.15
2	Switch	Smaller	0.15
3	Switch	Smaller	0.25
1	Stay	Bigger	0.3
2	Stay	Bigger	0.125
3	Stay	Bigger	0.075
1	Switch	Bigger	0.25
2	Switch	Bigger	0.6
3	Switch	Bigger	0.6

3. Sum out SecondChoice by restricting it in f5 to produce f6. Optimal policy for SecondChoice is Switch

FirstChoice	HostChoice	Value
1	Smaller	0.3
1	Bigger	0.3
2	Smaller	0.15
2	Bigger	0.6
3	Smaller	0.25
3	Bigger	0.6

4. Sum out HostChoice from f6 to produce f7

FirstChoice	Value
1	0.6
2	0.75
3	0.85

5. Restrict f7 to get optimal policy for FirstChoice

Value
0.85

6. Sum out the result to get expected utility of 0.85. Optimal policy for FirstChoice is to pick door 3. Optimal policy for SecondChoice to stay when FirstChoice is 1 and to switch otherwise. Expected utility of the optimal policy is 0.85.