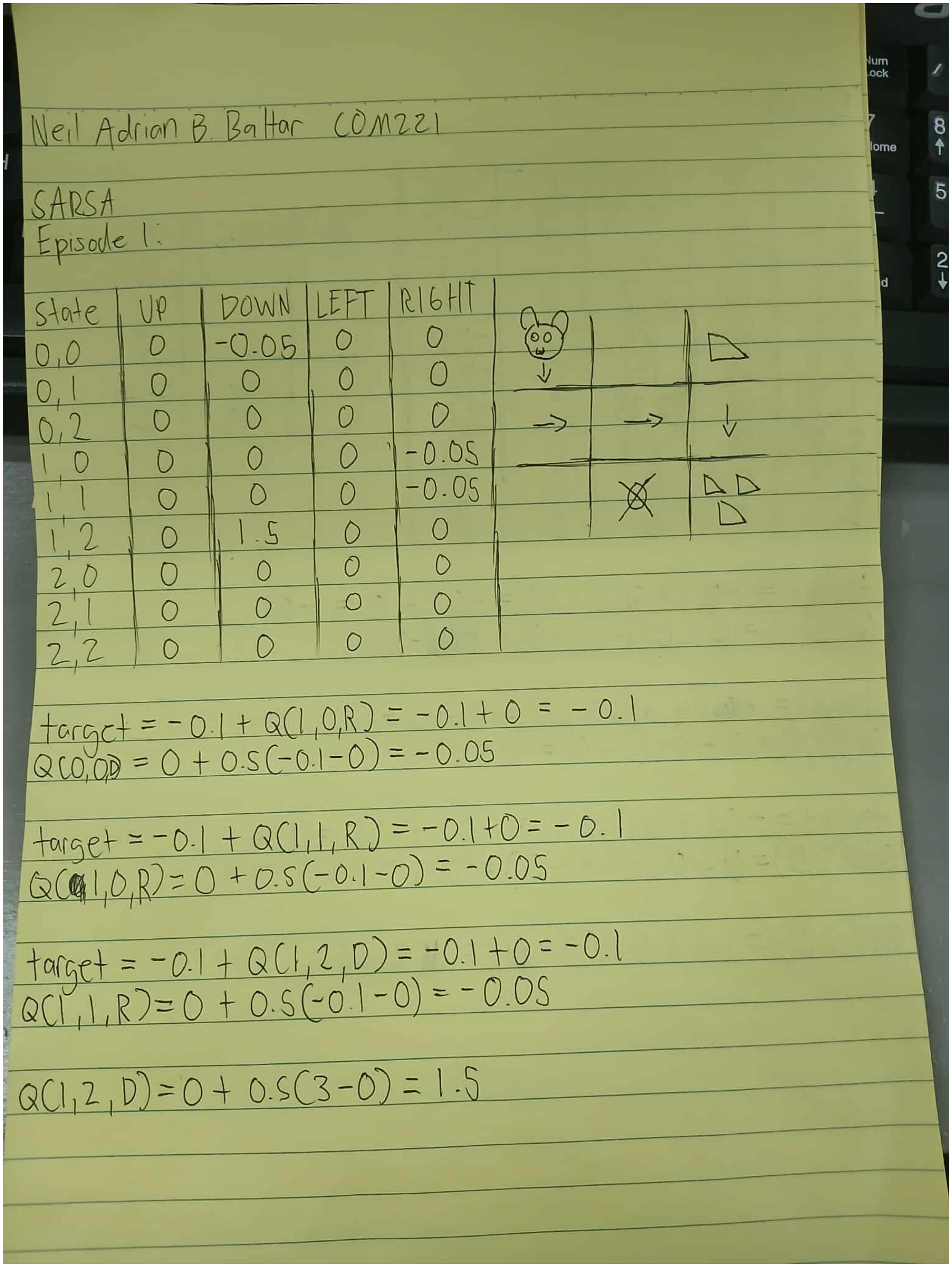
Neil Adrian B. Baltar

COM221



Neil Adrian B. Baltar COM221

SARSA

Episode 1:

State	UP	DOWN	LEFT	RIGHT
0,0	0	-0.05	0	0
0,1	0	0	0	0
0,2	0	0	0	0
1,0	0	0	0	-0.05
1,1	0	0	0	-0.05
1,2	0	1.5	0	0
2,0	0	0	0	0
2,1	0	0	0	0
2,2	0	0	0	0

$\text{target} = -0.1 + Q(1,0,R) = -0.1 + 0 = -0.1$

$Q(0,0,D) = 0 + 0.5(-0.1 - 0) = -0.05$

$\text{target} = -0.1 + Q(1,1,R) = -0.1 + 0 = -0.1$

$Q(1,0,R) = 0 + 0.5(-0.1 - 0) = -0.05$

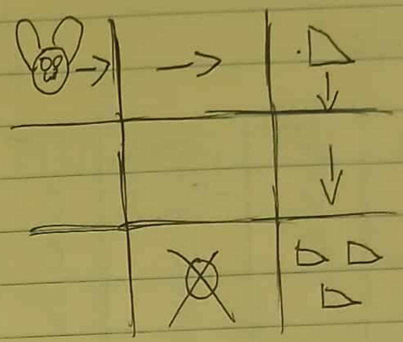
$\text{target} = -0.1 + Q(1,2,D) = -0.1 + 0 = -0.1$

$Q(1,1,R) = 0 + 0.5(-0.1 - 0) = -0.05$

$Q(1,2,D) = 0 + 0.5(3 - 0) = 1.5$

Episode 2:

State	UP	DOWN	LEFT	RIGHT
0,0	0	-0.05	0	-0.05
0,1	0	0	0	0.5
0,2	0	0.7	0	0
1,0	0	0	0	-0.05
1,1	0	0	0	-0.05
1,2	0	2.25	0	0
2,0	0	0	0	0
2,1	0	0	0	0
2,2	0	0	0	0



$target = -0.1 + Q(0,1,R) = -0.1 + 0 = -0.1$

$Q(0,0,R) = 0 + 0.5(-0.1-0) = -0.05$

$target = 1 + Q(0,2,D) = 1 + 0 = 1$

$Q(0,1,R) = 0 + 0.5(1-0) = 0.5$

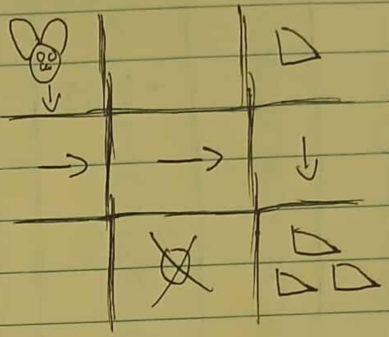
$target = -0.1 + Q(1,2,D) = -0.1 + 1.5 = 1.4$

$Q(0,2,D) = 0 + 0.5(1.4-0) = 0.7$

$Q(1,2,D) = 1.5 + 0.5(3-1.5) = 2.25$

Q-Learning
Episode 1:

State	UP	DOWN	LEFT	RIGHT
0,0	0	-0.05	0	0
0,1	0	0	0	0
0,2	0	0	0	0
1,0	0	0	0	-0.05
1,1	0	0	0	-0.05
1,2	0	1.5	0	0
2,0	0	0	0	0
2,1	0	0	0	0
2,2	0	0	0	0



$Q(0,0,D) = 0 + 0.5[-0.1 + 0 - 0] = -0.05$

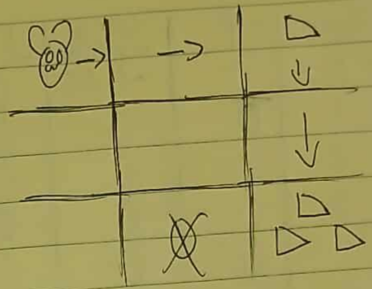
$Q(1,0,R) = 0 + 0.5[-0.1 + 0 - 0] = -0.05$

$Q(1,1,R) = 0 + 0.5[-0.1 + 0 - 0] = -0.05$

$Q(1,2,D) = 0 + 0.5[3 + 0 - 0] = 1.5$

Episode 2:

State	UP	DOWN	LEFT	RIGHT
0,0	0	-0.05	0	-0.05
0,1	0	0	0	0.5
0,2	0	0.7	0	0
1,0	0	0	0	-0.05
1,1	0	0	0	-0.05
1,2	0	2.25	0	0
2,0	0	0	0	0
2,1	0	0	0	0
2,2	0	0	0	0



$Q(0,0,R) = 0. + 0.5[-0.1 + 0 - 0] = -0.05$

$Q(0,1,R) = 0 + 0.5[1 + 0 - 0] = 0.5$

$Q(0,2,D) = 0 + 0.5[0.1 + \overset{1.5}{} - 0] = 0.7$

$Q(1,2,D) = 1.5 + 0.5[3 + 0 - \overset{1.5}{}] = 2.25$

How does the Q value of the starting state ((0,0)) differ under Sarsa and Q learning?
= They dont have any differences because the q-table starts with 0 values and dont provide actual change and only take the reward

Which one is learning the behavior (on policy)?
= Sarsa

Which one is learning the optimal greedy policy (off-policy)?
= Q-Learning