Ricky Ma 82943424
Bruce Cui 13412151
Late days claimed: 0

Question 1: Reinforcement Learning - Q-Learning

a. $Q[s, a] = r + \gamma max_{a'}(Q[s', a'])$

   i. Q[s17, right] = 2 + 0.9*max_a'(Q[s18, a']) = 2 + 0.9*0 = 2

   ii. Q[s18, up] = 8 + 0.9*max_a'(Q[s14, a']) = 8 + 0.9*0 = 8

   iii. Q[s14, right] = -6 + 0.9*max_a'(Q[s15, a']) = -6 + 0.9*0 = -6

b. $Q^i[s, a] = Q^{i-1}[s, a] + \alpha_k((r + \gamma max_{a'}Q^{i-1}[s', a']) - Q^{i-1}[s, a])$

   i. Q[s23, up] = 0 + 0.9*max_a'(Q[s18, a']) = 0 + 0.9*8 = 7.2

   ii. Q[s18, up] = Q^{i-1}[s18,up] + (1/k)*((r + 0.9*max_a'Q^{i-1}[s14, a']) - Q^{i-1}[s18, a'])

   = Q^{i-1}[s18,up] + (1/2)*((0 + 0.9*0) - Q^{i-1}[s18, a'])

   = 8 + 0.5*(-8) = 4

   iii. Q[s14, right] = Q^{i-1}[s14, right] + 0.5*((10 + 0.9*max_a'Q^{i-1}[s14, a']) - Q^{i-1}[s14, a'])

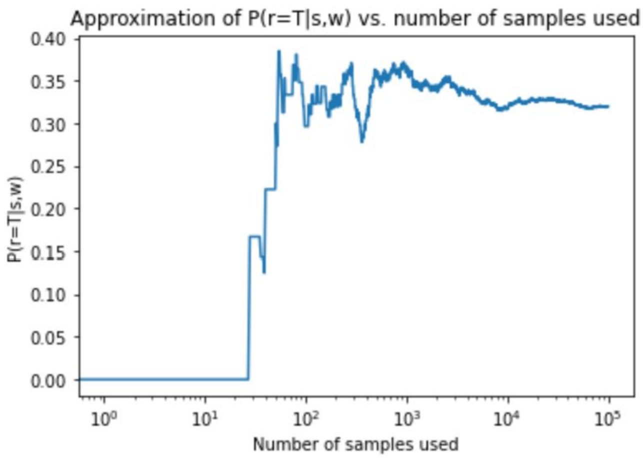
   = -6 + 0.5*((10 + 0.9*0) - (-6))

   = -6 + 0.5*(16) = 2

c. Only the second update in part b would change, because SARSA would choose the action with value -6. Q-learning chooses the action greedily, meaning the action with value 0 is chosen, because 0 > -6. SARSA, instead, only evaluates the actions suggested by the current policy, and therefore chooses the action with value -6.

Question 2: Approximate Reasoning in Belief Networks

a.




```
Approx. of P(r=T|s,w) w/ 100000 samples: 0.3198136868505912
```

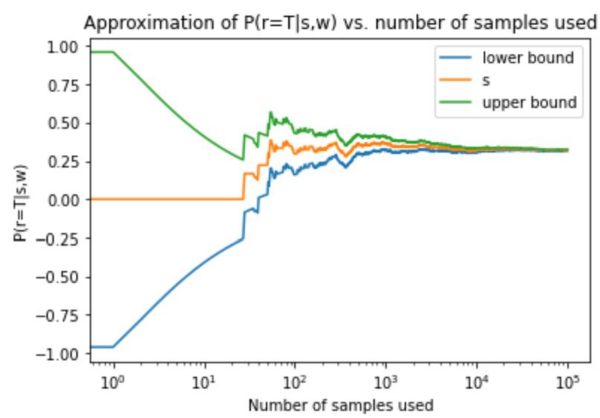
b. Hoeffding's inequality: $P(|s-p|>\varepsilon) \leq 2e^{-2n\varepsilon^2}$

$$2e^{-2n\varepsilon^2} < 0.05 \Rightarrow lne^{-2n\varepsilon^2} < ln(0.025) \Rightarrow$$

$$n\varepsilon^2 >- 0.5ln(0.025) \Rightarrow \varepsilon^2 > \frac{ln(40)}{2n}$$

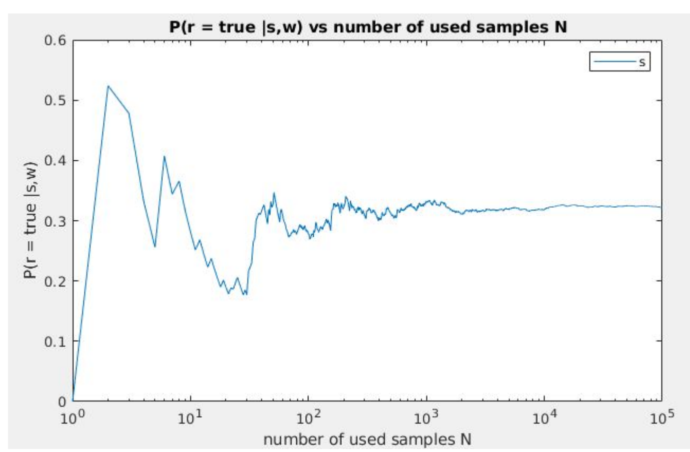
Tightest bound: $\varepsilon > \sqrt{\frac{ln(40)}{2n}}$

```
Accepted samples at N=100000: 27910
epsilon for n=27910: 0.008129284366775905
```




```
Approx. of P(r=T|s,w) w/ 100000 samples: 0.3198136868505912
```

c. It is evident that approximating P(r|s,w) using the likelihood weighting converges faster (we can see from the plot that the algorithm converges between 100-1000 samples). The convergence is also much more stable. With rejection sampling, the algorithm doesn't converge until more than 10000 samples are used.



Question 3: Temporal Reasoning in Belief Networks

a. $P(X_i = t|e),\ P(X_0 = true) = P(X_0 = false) = 0.5,\ e = (t, f, t)$

i. $P(X_1|e) = \alpha P(X_1|e_{0:1})P(e_{2:3}|X_1) = \alpha * [0.765, 0.235] * [0.196, 0.262] \propto [0.709, 0.291]$

- $P(X_1|e_{0:1}) = \alpha P(e_1|X_1)\sum_{X_0} P(X_1|X_0)P(X_0|e_0)$
  - $= \alpha * [0.8, 0.3] * (0.5[0.7, 0.3] + 0.5[0.4, 0.6])$
  - $= \alpha * [0.44, 0.135] \propto [0.765, 0.235]$
- $P(e_{2:3}|X_1) = (0.2 * 0.65[0.7, 0.4]) + (0.7 * 0.5[0.3, 0.6]) = [0.196, 0.262]$
- $P(X_1 = t|e) = 0.709$

ii. $P(X_2|e) = \alpha P(X_2|e_{0:2})P(e_{3:3}|X_2) = \alpha * [0.389, 0.611] * [0.65, 0.5] \propto [0.454, 0.546]$

- $P(X_2|e_{0:2}) = \alpha P(e_2|X_2)\sum_{X_1} P(X_2|X_1)P(X_1|e_{0:1})$
  - $= \alpha * [0.3, 0.8] * (0.765[0.7, 0.3] + 0.235[0.4, 0.6])$
  - $= \alpha * [0.18885, 0.2964] \propto [0.389, 0.611]$
- $P(e_{3:3}|X_2) = (0.8 * 1[0.7, 0.4]) + (0.3 * 1[0.3, 0.6]) = [0.65, 0.5]$
- $P(X_2 = t|e) = 0.454$

iii. $P(X_3|e) = \alpha P(X_3|e_{0:3})P(e_{4:3}|X_3) = \alpha * [0.74, 0.26] * [1, 1] \propto [0.74, 0.26]$

- $P(X_3|e_{0:3}) = \alpha P(e_3|X_3)\sum_{X_2} P(X_3|X_2)P(X_2|e_{0:2})$
  - $= \alpha * [0.8, 0.3] * (0.389[0.7, 0.3] + 0.611[0.4, 0.6])$
  - $= \alpha * [0.4136, 0.1449] \propto [0.741, 0.259]$
- $P(e_{4:3}|X_3) = [1, 1]$
- $P(X_3 = t|e) = 0.741$

b. Most likely sequence with Viterbi algorithm: $[s_1, s_2, s_3] = [true, true, true]$

i. $m_{1:1} = P(X_1|e) = [0.709, 0.291]$

ii. $m_{1:2} = P(e_2|X_2) * [max[0.8P(X_2|X_1), 0.2P(X_2|\neg X_1)], max[0.8P(\neg X_2|X_1), 0.2P(\neg X_2|\neg X_1)]]$

- $= [0.2, 0.7] * [max[0.709 * 0.7, 0.291 * 0.4], max[0.709 * 0.3, 0.291 * 0.6]]$
- $= [0.2, 0.7] * [max[0.4963, 0.1164], max[0.2127, 0.1746]]$
- $= [0.2, 0.7] * [0.4963, 0.2127] = [0.09926, 0.14889]$

The previous state that maximized the probability of being in the true state is: $X_1 = t$

The previous state that maximized the probability of being in the false state is: $X_1 = t$

iii. $m_{1:3} = P(e_3|X_3) * [max[0.099P(X_3|X_2), 0.149P(X_3|\neg X_2)], max[0.099P(\neg X_3|X_2), 0.149P(\neg X_3|\neg X_2)]]$

- $= [0.8, 0.3] * [max[0.099 * 0.7, 0.149 * 0.4], max[0.099 * 0.3, 0.149 * 0.6]]$
- $= [0.8, 0.3] * [max[0.0695, 0.0596], max[0.0298, 0.0893]]$
- $= [0.8, 0.3] * [0.0695, 0.0893] = [0.0556, 0.0268]$

The previous state that maximized the probability of being in the true state is: $X_2 = t$

The previous state that maximized the probability of being in the false state is: $X_2 = f$