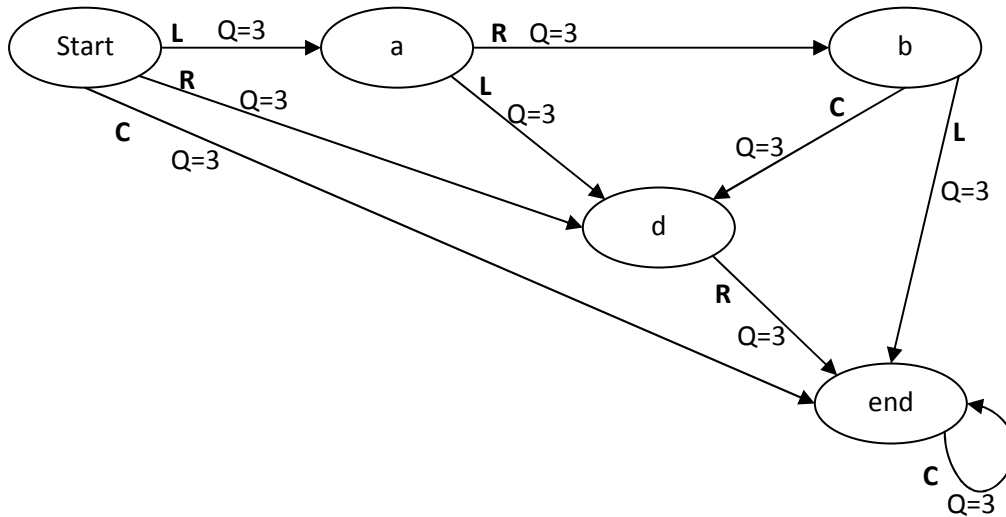


Solution to CS760 HW 4 (Spring 2010)

1. Initial values: all $Q=3$.



i) For the first episode: start->a->b->d->end, we have the following Q values:

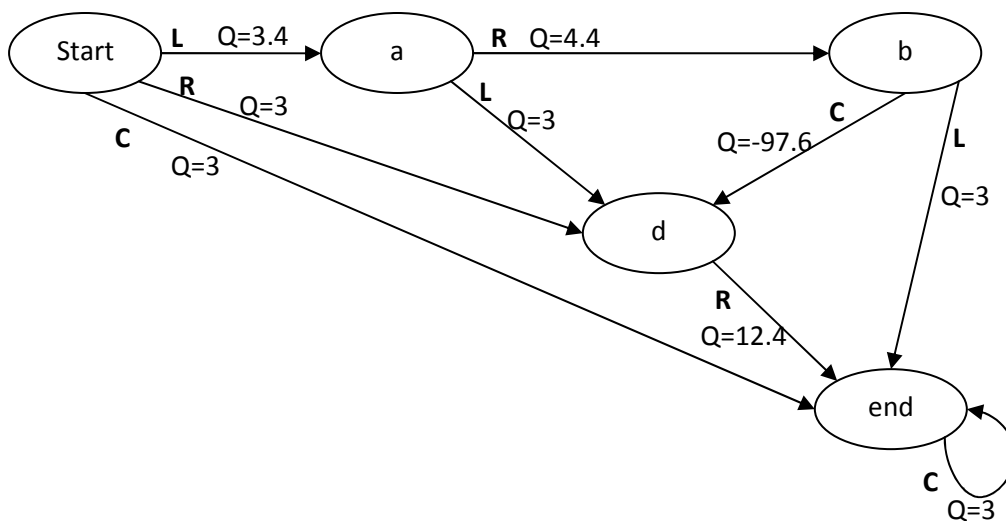
Step 1: start->a. We have $Q(start, L) = R(start, L) + \gamma \max_{action} Q(a, action) = 1 + 0.8 * 3 = 3.4$

Step 2: a->b. We have $Q(a, R) = R(a, R) + \gamma \max_{action} Q(b, action) = 2 + 0.8 * 3 = 4.4$

Step 3: b->d. We have $Q(b, C) = R(b, C) + \gamma \max_{action} Q(d, action) = -100 + 0.8 * 3 = -97.6$

Step 4: d->end. We have $Q(d, R) = R(d, R) + \gamma \max_{action} Q(end, action) = 10 + 0.8 * 3 = 12.4$

The resulting Q table is shown below:



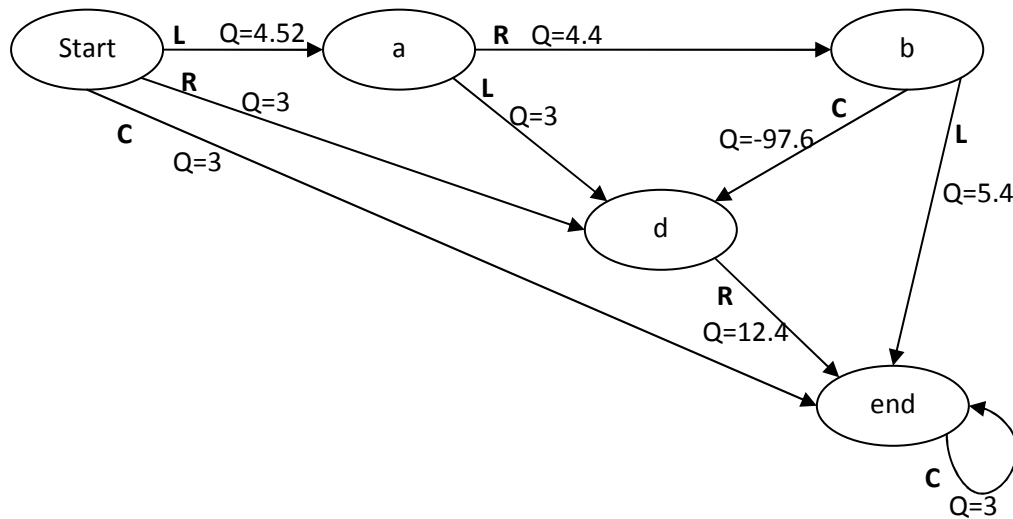
ii) For the second episode: start->a->b->end, we have the following Q values:

Step 1: start->a. We have $Q(start, L) = R(start, L) + \gamma \max_{action} Q(a, action) = 1 + 0.8 * 4.4 = 4.52$

Step 2: a->b. We have $Q(a, R) = R(a, R) + \gamma \max_{action} Q(b, action) = 2 + 0.8 * 3 = 4.4$

Step 3: b->end. We have $Q(b, L) = R(b, L) + \gamma \max_{action} Q(end, action) = 3 + 0.8 * 3 = 5.4$

The resulting Q table is shown below:



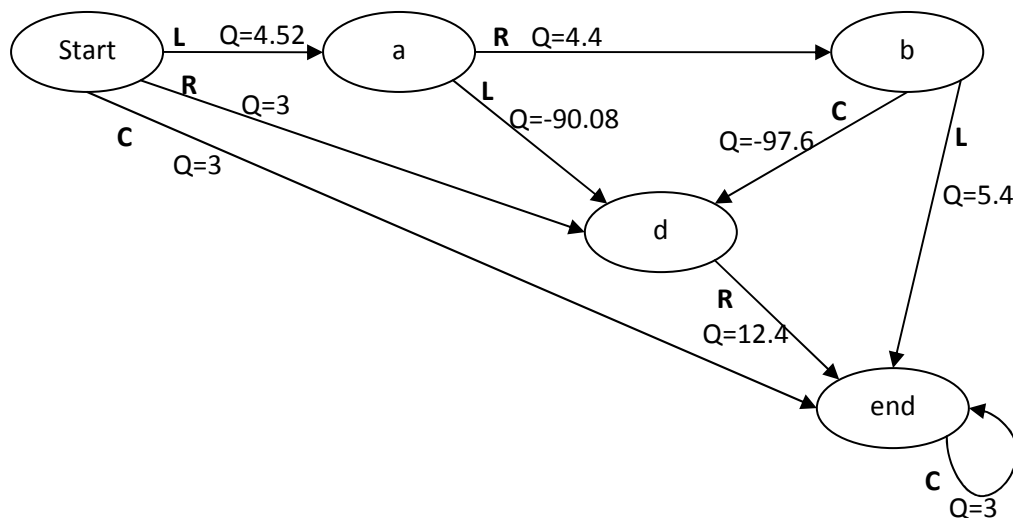
iii) For the third episode: start->a->d->end, we have the following Q values:

Step 1: start->a. We have $Q(start, L) = R(start, L) + \gamma \max_{action} Q(a, action) = 1 + 0.8 * 4.4 = 4.52$

Step 2: a->d. We have $Q(a, L) = R(a, L) + \gamma \max_{action} Q(d, action) = -100 + 0.8 * 12.4 = -90.08$

Step 3: d->end. We have $Q(d, R) = R(d, R) + \gamma \max_{action} Q(end, action) = 10 + 0.8 * 3 = 12.4$

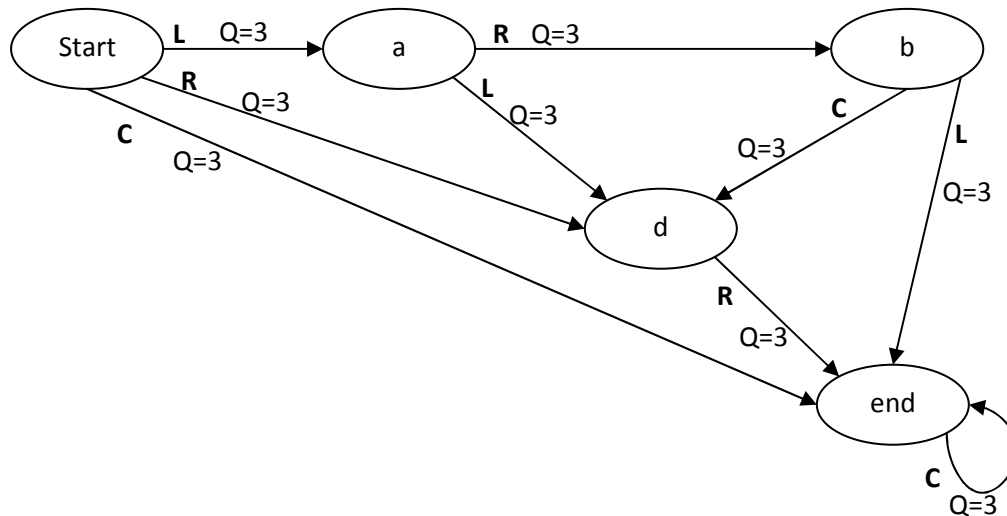
The resulting Q table is shown below:



2. We need to use the learning rate (i.e., alpha) here in order for the Q values to converge, because for SARSA we use the actual next action instead of the maximum of the Q value for next state. We don't want the latest estimate to overwrite the previous estimate, since the latest estimate might be due to an exploration move.

This time recalculate problem 1 using SARSA.

Initial values: all Q=3.



i) For the first episode: start->a->b->d->end, we have the following Q values:

Step 1: start->a. We have

$$Q(start, L) = Q(start, L) + \frac{1}{1 + visits(start, L)} [R(start, L) + \gamma Q(a, R) - Q(start, L)]$$

$$= 3 + \frac{1}{1} [1 + 0.8 * 3 - 3] = 3.4$$

It is also fine to start counting visits from 1 instead of 0; in that case, the first alpha value is 1/2, and the new estimate is averaged with the initial Q values in the Q table.

Step 2: a->b. We have

$$Q(a, R) = Q(a, R) + \frac{1}{1 + visits(a, R)} [R(a, R) + \gamma Q(b, C) - Q(a, R)]$$

$$= 3 + \frac{1}{1} [2 + 0.8 * 3 - 3] = 4.4$$

Step 3: b->d. We have

$$Q(b, C) = Q(b, C) + \frac{1}{1 + visits(b, C)} [R(b, C) + \gamma Q(d, R) - Q(b, C)]$$

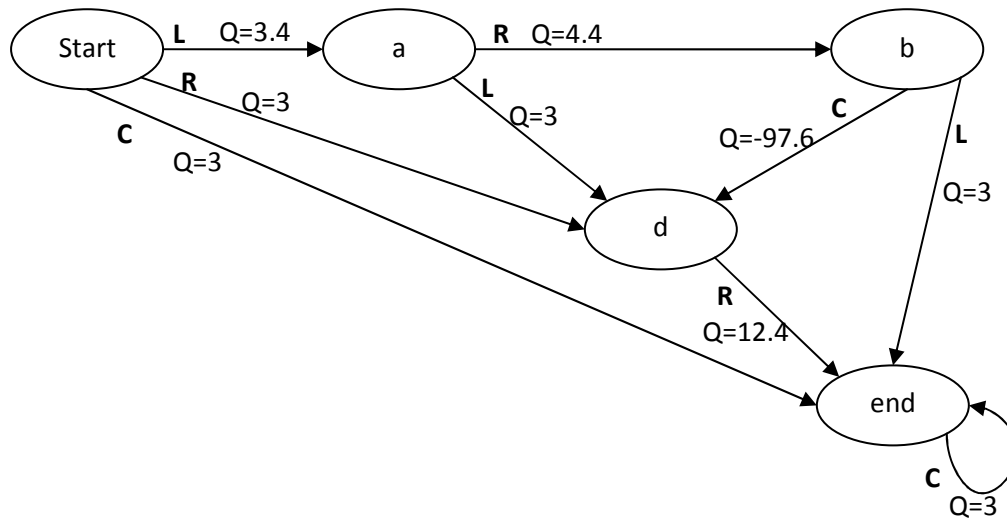
$$= 3 + \frac{1}{1} [-100 + 0.8 * 3 - 3] = -97.6$$

Step 4: d->end. We have

$$Q(d, R) = Q(d, R) + \frac{1}{1 + \text{visits}(d, R)} [R(d, R) + \gamma Q(\text{end}, C) - Q(d, R)]$$

$$= 3 + \frac{1}{1} [10 + 0.8 * 3 - 3] = 12.4$$

The resulting Q table is shown below:



ii) For the first episode: start->a->b->end, we have the following Q values:

Step 1: start->a. We have

$$Q(\text{start}, L) = Q(\text{start}, L) + \frac{1}{1 + \text{visits}(\text{start}, L)} [R(\text{start}, L) + \gamma Q(a, R) - Q(\text{start}, L)]$$

$$= 3.4 + \frac{1}{2} [1 + 0.8 * 4.4 - 3.4] = 3.96$$

Step 2: a->b. We have

$$Q(a, R) = Q(a, R) + \frac{1}{1 + \text{visits}(a, R)} [R(a, R) + \gamma Q(b, L) - Q(a, R)]$$

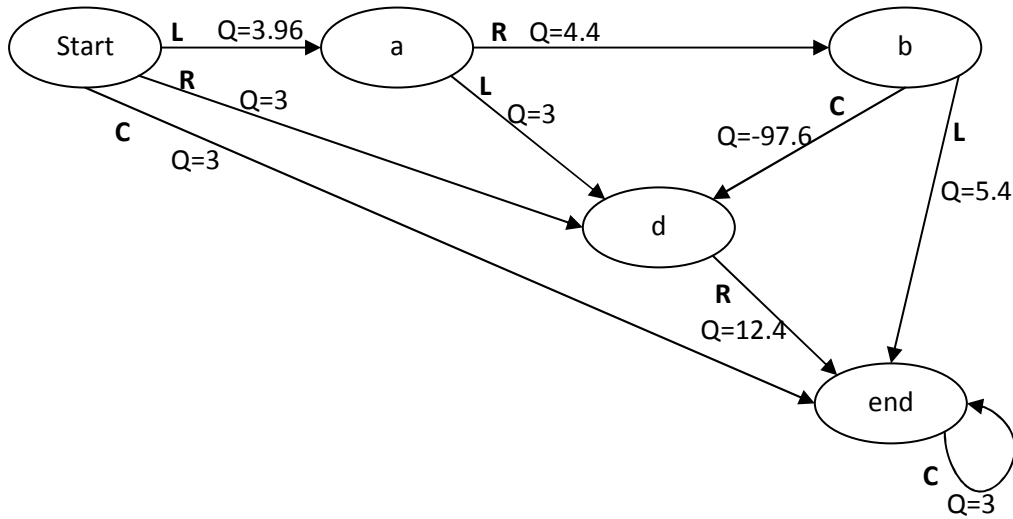
$$= 4.4 + \frac{1}{2} [2 + 0.8 * 3 - 4.4] = 4.4$$

Step 3: b->end. We have

$$Q(b, L) = Q(b, L) + \frac{1}{1 + \text{visits}(b, L)} [R(b, L) + \gamma Q(\text{end}, C) - Q(b, L)]$$

$$= 3 + \frac{1}{1} [3 + 0.8 * 3 - 3] = 5.4$$

The resulting Q table is shown below:



iii) For the first episode: start->a->d->end, we have the following Q values:

Step 1: start->a. We have

$$Q(start, L) = Q(start, L) + \frac{1}{1 + visits(start, L)} [R(start, L) + \gamma Q(a, L) - Q(start, L)]$$

$$= 3.96 + \frac{1}{3} [1 + 0.8 * 3 - 3.96] = 3.773$$

Step 2: a->d. We have

$$Q(a, L) = Q(a, L) + \frac{1}{1 + visits(a, L)} [R(a, L) + \gamma Q(d, R) - Q(a, L)]$$

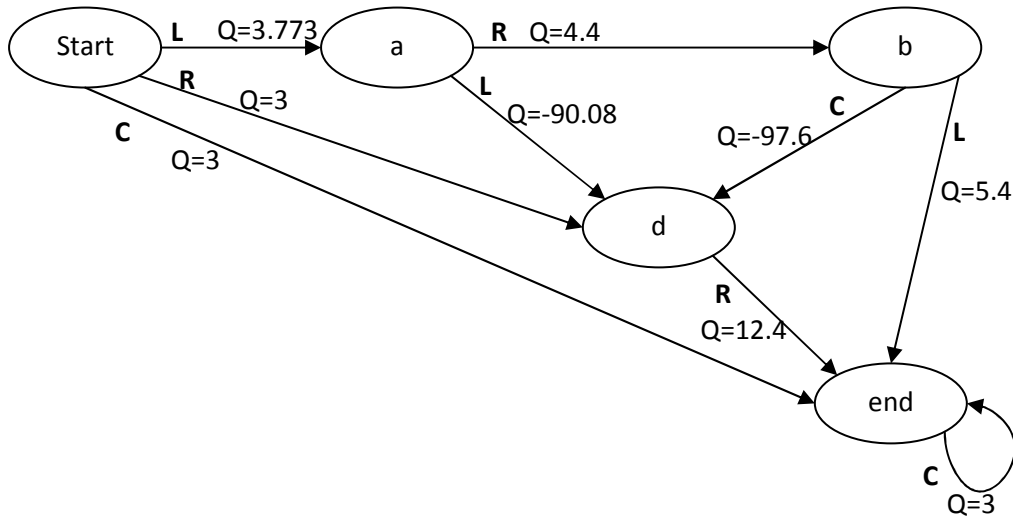
$$= 3 + \frac{1}{1} [-100 + 0.8 * 12.4 - 3] = -90.08$$

Step 3: d->end. We have

$$Q(d, R) = Q(d, R) + \frac{1}{1 + visits(d, R)} [R(d, R) + \gamma Q(end, C) - Q(d, R)]$$

$$= 12.4 + \frac{1}{2} [10 + 0.8 * 3 - 12.4] = 12.4$$

The resulting Q table is shown below:



3. If RL is performed for a large number of episodes, then we could consider each Q value to be converged to its true value. Since the “end” node is an absorbing goal state, we can easily calculate the convergence value by calculating it backwardly. Note: an informal argument is sufficient. It is certainly fine to compute all the Q’s, but doing so was not required.

First, we could repeatedly calculate:

$$Q(\text{end}, C) = R(\text{end}, C) + \gamma \max_{\text{action}} Q(\text{end}, \text{action}) = \gamma Q(\text{end}, C), \text{ so } Q(\text{end}, C) = 0$$

Next, we could calculate $Q(\text{start}, C)$, $Q(b, L)$, $Q(d, R)$:

$$Q(\text{start}, C) = R(\text{start}, C) + \gamma Q(\text{end}, C) = 0$$

$$Q(b, L) = R(b, L) + \gamma Q(\text{end}, C) = 3$$

$$Q(d, R) = R(d, R) + \gamma Q(\text{end}, C) = 10$$

Now we have $\max_{\text{action}} Q(d, \text{action}) = Q(d, R) = 10$, so $Q(b, C)$, $Q(a, L)$, $Q(\text{start}, R)$ could be

calculated in the same manner:

$$Q(b, C) = R(b, C) + \gamma \max_{\text{action}} Q(d, \text{action}) = -100 + 0.8 * 10 = -92$$

$$Q(a, L) = R(a, L) + \gamma \max_{\text{action}} Q(d, \text{action}) = -100 + 0.8 * 10 = -92$$

$$Q(\text{start}, R) = R(\text{start}, R) + \gamma \max_{\text{action}} Q(d, \text{action}) = -100 + 0.8 * 10 = -92$$

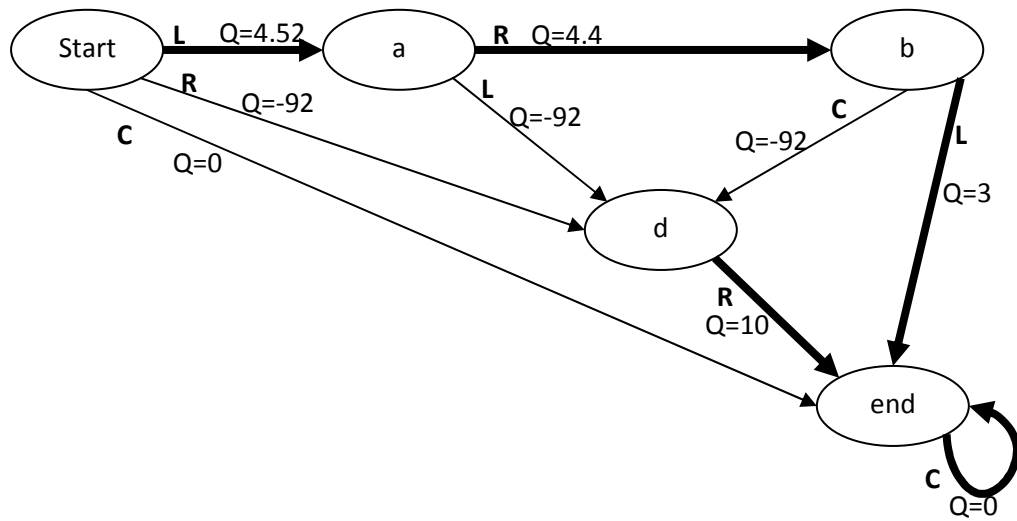
Now we have $\max_{\text{action}} Q(b, \text{action}) = Q(b, L) = 3$, so we could calculate $Q(a, R)$:

$$Q(a, R) = R(a, R) + \gamma \max_{\text{action}} Q(b, \text{action}) = 2 + 0.8 * 3 = 4.4$$

Now we have $\max_{\text{action}} Q(a, \text{action}) = Q(a, R) = 4.4$, so we could calculate $Q(\text{start}, L)$:

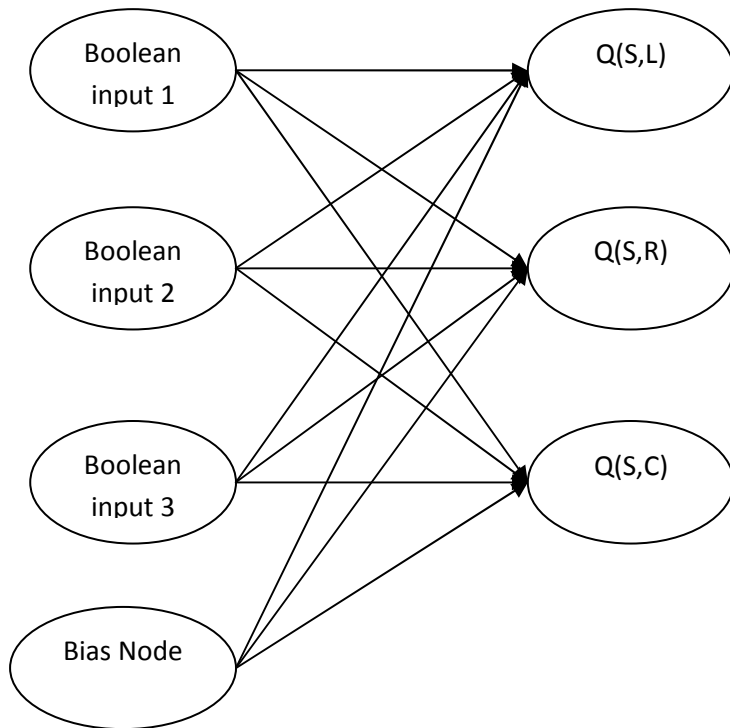
$$Q(\text{start}, L) = R(\text{start}, L) + \gamma \max_{\text{action}} Q(a, \text{action}) = 1 + 0.8 * 4.4 = 4.52$$

For each node, the policy will be the maximum Q value from that node, and is illustrated in thick arrows in the following figure:



The generated policy (thick lines indicated above) is exactly the maximum discounted cumulative reward achieved from each state.

4. The perceptron for representing the Q function could be represented as the figure below:



The Boolean input nodes correspond to the three bits used to represent input states. For example, for “start” state, the input to three nodes will be 0, 0, 1, respectively. The activation node always has an input of 1 (ok to use -1 instead) for bias. The output nodes determine the Q value for each action with the current state. The Q value for each of the action is the weighted sum of the product of the weight and its corresponding input. For example, assume the input nodes are denoted as I_1, I_2, I_3, I_4 , and the weight from node i to node j is denoted as w_{ij} . Then the output $Q_j = \sum_i I_i w_{ij}$.

For the illustration of training process, I used the following table to represent the weights:

	Input 1	Input 2	Input 3	Bias
Q(S,L)	3	3	3	3
Q(S,R)	3	3	3	3
Q(S,C)	3	3	3	3

For the first episode: start->a->b->d->end, we have the following Q values:

Step 1: start->a. We have

$$Q(start, L) = \sum_i I_i w_{i,L} = 0*3 + 0*3 + 1*3 + 1*3 = 6$$

$$Q(a, R) = \sum_i I_i w_{i,R} = 0*3 + 1*3 + 0*3 + 1*3 = 6$$

So the target output should be:

$$T(start, L) = Q(start, L) + \alpha[R(start, L) + \gamma Q(a, R) - Q(start, L)] = 6 + 1*[1 + 0.8*6 - 6] = 5.8$$

And we can update the weights for $w_{i,L}$ as:

$$w_{i,L} = w_{i,L} + \eta(T(start, L) - Q(start, L)) * I_i$$

$$\text{So, } w_{1,L} = 3, w_{2,L} = 3, w_{3,L} = 3 + 0.5*(5.8 - 6)*1 = 2.9, w_{4,L} = 3 + 0.5*(5.8 - 6)*1 = 2.9$$

Step 2: a->b. We have

$$Q(a, R) = \sum_i I_i w_{i,R} = 0*3 + 1*3 + 0*3 + 1*3 = 6$$

$$Q(b, C) = \sum_i I_i w_{i,C} = 1*3 + 0*3 + 0*3 + 1*3 = 6$$

So the target output should be:

$$T(a, R) = Q(a, R) + \alpha[R(a, R) + \gamma Q(b, C) - Q(a, R)] = 6 + 1*[2 + 0.8*6 - 6] = 6.8$$

And we can update the weights for $w_{i,R}$ as:

$$w_{i,R} = w_{i,R} + \eta(T(a, R) - Q(a, R)) * I_i$$

$$\text{So, } w_{1,R} = 3, w_{2,R} = 3 + 0.5*(6.8 - 6)*1 = 3.4, w_{3,R} = 3, w_{4,R} = 3 + 0.5*(6.8 - 6)*1 = 3.4$$

Step 3: b->d. We have

$$Q(b, C) = \sum_i I_i w_{i,C} = 1*3 + 0*3 + 0*3 + 1*3 = 6$$

$$Q(d, R) = \sum_i I_i w_{i,C} = 1*3 + 1*3.4 + 0*3 + 1*3.4 = 9.8$$

So the target output should be:

$$T(b, C) = Q(b, C) + \alpha[R(b, C) + \gamma Q(d, R) - Q(b, C)] = 6 + 1*[-100 + 0.8*9.8 - 6] = -92.16$$

And we can update the weights for $w_{i,C}$ as:

$$w_{i,C} = w_{i,C} + \eta(T(b, C) - Q(b, C)) * I_i$$

$$\text{So, } w_{1,C} = 3 + 0.5 * (-92.16 - 6) * 1 = -46.08, w_{2,C} = 3, w_{3,C} = 3,$$

$$w_{4,C} = 3 + 0.5 * (-92.16 - 6) * 1 = -46.08$$

Step 4: d->end. We have

$$Q(d, R) = \sum_i I_i w_{i,C} = 1*3 + 1*3.4 + 0*3 + 1*3.4 = 9.8$$

$$Q(\text{end}, C) = \sum_i I_i w_{i,C} = 0$$

So the target output should be:

$$T(d, R) = Q(d, R) + \alpha[R(d, R) + \gamma Q(\text{end}, C) - Q(d, R)] = 9.8 + 1*[10 + 0.8*0 - 9.8] = 10$$

And we can update the weights for $w_{i,R}$ as:

$$w_{i,R} = w_{i,R} + \eta(T(d, R) - Q(d, R)) * I_i$$

$$\text{So, } w_{1,R} = 3 + 0.5 * (10 - 9.8) * 1 = 3.1, w_{2,C} = 3.4 + 0.5 * (10 - 9.8) * 1 = 3.5, w_{3,C} = 3,$$

$$w_{4,C} = 3.4 + 0.5 * (10 - 9.8) * 1 = 3.5$$

So after the first episode, the weight matrix is could be expressed as below:

	Input 1	Input 2	Input 3	Bias
Q(S,L)	3	3	2.9	2.9
Q(S,R)	3.1	3.5	3	3.5
Q(S,C)	-46.08	3	3	-46.08

Following this first episode, the initial states for the three possible actions are:

$$Q(\text{start}, L) = \sum_i I_i w_{i,L} = 0*3 + 0*3 + 1*2.9 + 1*2.9 = 5.8$$

$$Q(\text{start}, R) = \sum_i I_i w_{i,R} = 0*3.1 + 0*3.5 + 1*3 + 1*3.5 = 6.5$$

$$Q(\text{start}, C) = \sum_i I_i w_{i,R} = 0*(-46.08) + 0*3 + 1*3 + 1*(-46.08) = -43.08$$