Problem 1

Thomas Gauthier
Nicolas Gauthier
Raul da Silva Souza

November 2017, 13

Question 1

We implement the value iteration algorithm for the problem. The code, implemented in Matlab, is available at `https://github.com/ThomasGauth/ENSTA-ROB311/blob/master/Problem1_ROB311.m`.

To solve this problem we need to know the rewards, the actions and the transition matrices. The first two are given in the problem, but the transition matrices were calculated by hand and then added to the code. From this data, we calculate the utilities for all states using the algorithm from the lecture. We use two vectors : U and U1 to calculate the new utilities out of the previous ones at each iteration. We initialize them with zero values. We stop the algorithm and consider that it converged when the new and old utilities are close enough. To decide how close it is, we use a factor we call *epsilon*. The stopping criterion consists in a comparison between this factor and the RMSE between the two utilities vector.

In the end, the best policy is computed according to the formula given in the lecture (slide 19).

Question 2

The algorithm takes 3709 iterations to converge with gamma = 0.999 and epsilon = 0.01. The calculated policy is ['east', 'east', 'still', 'east', 'east', 'south'], which corresponds respectively to the states ['state 1', 'state 2', 'state 3', 'state 4', 'state 5', 'state 6'].

Question 3

With gamma = 0.1 and epsilon = 0.01, the algorithm takes 3 iterations to converge. The best policy is ['east', 'east', 'still', 'east', 'east', 'south'].

We can note that in the two cases, the policies are the same : go in the direction of the terminal state. Even if the utilities are not the same, the hierarchy between values is the same and thus, the best policy too. The main difference between the policies with gamma = 0.999 and gamma = 0.1 is the number of iterations needed to converge. In the second case, convergence is much faster.