Homework 5 – Parts A and B

Part A:

1. (Making Simple Decisions, Ch. 16) Work problem 16.2 on page 610 in the text (“Tickets to a lottery…”).
2. (Making Simple Decisions, Ch. 16) Work problem 16.11 on page 611 in the text (“A used-car buyer…”).
3. (Making Complex Decisions, Ch. 17) Work problem 17.1 (1st sentence) on page 646 in the text (“For the 4x3…”).
4. (Making Complex Decisions, Ch. 17) Work problem 17.3 on page 646 in the text (“Can any finite…”).
5. (Making Complex Decisions, Ch. 17) Work problem 17.4 on page 646 in the text (“Consider an undiscounted MDP…”).
6. (Making Complex Decisions, Ch. 17) Work problem 17.8 on page 647-648 in the text (“In this exercise we will consider…”).
7. (Making Complex Decisions, Ch. 17) Work problem 17.10 on page 648 in the text (“In the children’s game…”).

Part B:

This part of homework 5 uses the supplied LISP code for Markov Decision Processes. Refer to Homework 2 for a reminder of how to start up the code. Here, instead of `(aima-load 'logic), you’ll use the following:

`(aima-load 'uncertainty)

An overview of the “uncertainty” code is provided here: http://aima.cs.berkeley.edu/lisp/doc/overview-UNCERTAINTY.html.

The source code for testing the uncertainty software is here: http://aima.cs.berkeley.edu/lisp/uncertainty/test-uncertainty.lisp, and can be executed as follows:

`(test 'uncertainty)

The test-uncertainty.lisp code currently implements the Markov Decision Process for the 4x3 stochastic environment in Figure 17.1. This code first determines the optimal policy for this environment, and then runs the agent from its starting location until it reaches a terminal state. As you will see from running the test code, the testing outputs the location of the agent at each time step, its reward, its action, etc. Upon running the test code, the optimal policy found is that shown in Figure 17.2(a), and the agent moves according to this policy from state (1,1) until it reaches the terminal state (4,1).

Figure 17.2(b) shows 4 different optimal policies for different settings of the reward for each state, $R(s)$. As stated on page 616, six other optimal policies exist for other ranges of $R(s)$.

**Your assignment for Part B:** Using the supplied LISP code, make adaptations as needed to find the other 6 optimal policies for ranges of $R(s)$. Turn in:

- 6 graphs like those shown in Figure 17.2(b) that illustrate the 6 new optimal policies, clearly noting the ranges of $R(s)$ that apply to each policy (just as it is done in Figure 17.2(b)). (Note: Together with the 4 policies shown in Figure 17.2(b), these policies should cover all possible settings of $R(s)$.)

- The output of 6 runs of the supplied code that demonstrate these 6 distinct policies. (You should select a starting point of the agent that illustrates the differences of each policy from its “neighboring” policies. For some policies, the behavior will be the same for an agent starting in, say (1,1), than for an agent starting in another state, such as (4, 1). So, select a starting location that is most unique to that policy compared to its neighboring policies).