Homework **Spring 2011**
COMP150 AML - Learning, Planning and Acting in Complex Environments
Department of Computer Science, Tufts University **Roni Khardon**

Homework Assignment 1

This assignment asks you to implement versions of value iteration (VI) and policy iteration (PI) for a particular MDP and evaluate their performance.
The assignment is due by Thursday Feb 17 (in class).

Consider a generalized version of the example MDP from class. We have a $n \times n$ grid with $n = 15$, with a single reward of 10 at the top right state, and where the episode ends when this state is reached. Intended moves: *left* and *down* are deterministic. Intended moves: *up* and *right* do not always perform as intended and instead have the following transitions. Action: *up* goes up with probability 0.5, left with probability 0.25, and right with probability 0.25. Action: *right* goes right with probability 0.9, left with probability 0.05, and down with probability 0.05. In the above, if the implied move hits the boundary the agent stays in the same position. We assume a discount factor of 0.9 in this problem.

Your task is to implement the VI and PI algorithms where in PI we perform iterative policy evaluation. Fix a stopping criterion of error < 0.01 in VI and for iterative policy evaluation. Your implementation should be such that you can "empirically evaluate" the greedy policy with respect to the current value function after every iteration. To "empirically evaluate" a policy, we run it for 100 episodes where each episode is at most 100 steps long. Each run starts at the bottom left state and the quality of the policy is evaluated as the average of the total discounted reward received in these runs.

You should implement the algorithms and measure the quality of policies after every iteration. Then plot the quality of the policies as a function of the "effort" of the algorithm. You should produce two plots comparing VI and PI. In the first, quality is plotted as a function of the number of iterations. In the second the quality is plotted as a function of the number of "single action backups". By the latter we mean the number of times you calculate $r(s,a) + \sum_{s'} Pr[s'|s,a]V(s')$ during the algorithm. Assuming we have S states and A actions in each state, a single VI iteration costs SA "single action backups" since we have A such backups per state. A single PI iteration costs $SA + SI$ such backups where I is the number of internal iterations in policy evaluation.

Please submit (1) printouts of your code and (2) results of the experiments which should include the graphs explained above and any observations on them.

It is up to you to decide on language for implementation but please make sure to write in good style and document your code clearly so we can read your code.

Extra Work: Those wishing to explore further may want to explore how the results change as a function of the error parameter chosen (0.01 above). Also you may want to implement a version of modified policy iteration and evaluate the performance when the number of internal iterations in policy evaluation is kept small.