CS533
Intelligent Agents and Decision Making
Homework 3, Winter 2011

1. Construct a simple Markov Decision Process such that the optimal policy for maximizing finite-horizon total reward must be non-stationary. That is, the MDP should not have a stationary policy that maximizes the finite horizon total reward.

2. In class it was mentioned that STRIPS is just a representation for a restricted class of MDPs.

   (a) Describe how, in general, a STRIPS planning problem can be converted into an equivalent MDP. In other words, given a STRIPS planning problem, describe the components of an MDP (i.e. state space, actions, transition function, rewards) such that solutions to the MDP are a solutions to the STRIPS problem.

   (b) Roughly how many states does the MDP have for a STRIPS problem with n propositions?

   (c) What is the complexity of finite horizon value iteration (for a horizon of h) in terms of the number of propositions and actions?

3. In many problems, not all actions are applicable in all states and many actions only lead to a small number of next states, compared to the total number of states. In this question, we consider how the complexity of finite-horizon value iteration and policy evaluation can be improved for such problems.

   To capture the notion of applicable actions, suppose that we have a function LEGAL(s) that takes a state s and returns the set of legal actions in s. Also suppose that we have a function NEXT(s, a), which takes a state s and action a as input and returns the set of states that have non-zero probability of occurring after taking a in state s. That is,

   $$\text{NEXT}(s, a) = \{s' \mid T(s, a, s') > 0\}.$$

   Assume that we are considering an MDP with n states and m actions such that for any state s and action a we have LEGAL(s) $\leq k$ and NEXT(s, a) $\leq r$. Assume that the time and space complexity of evaluating the functions NEXT and LEGAL are linear in the sizes of their output (i.e. the number of elements in their sets).

   (a) Describe how to modify the finite-horizon policy evaluation algorithm described in class, using one or both of the new functions, so that the time complexity is improved when $r < n$ and $k < m$. What is the time complexity? The time complexity should be expressed in terms of r and k when possible and may also involve n and m.

   (b) Repeat part (a) but for the finite-horizon value iteration algorithm described in class.