Homework 1

Due 9/11/2012 2 PM

1. **Question 1.7** This question aims to test your understanding of the principles of AI.

   - Although bar code scanning is in a sense computer vision, these are not AI systems. The problem of reading a bar code is an extremely limited and artificial form of visual interpretation, and it has been carefully designed to be as simple as possible, given the hardware.
   - In many respects. The problem of determining the relevance of a web page to a query is a problem in natural language understanding, and the techniques are related to those we will discuss in Chapters 22 and 23. Search engines like Ask.com, which group the retrieved pages into categories, use clustering techniques analogous to those we discuss in Chapter 20. Likewise, other functionalities provided by a search engines use intelligent techniques; for instance, the spelling corrector uses a form of data mining based on observing users' corrections of their own spelling errors. On the other hand, the problem of indexing billions of web pages in a way that allows retrieval in seconds is a problem in database design, not in artificial intelligence.
   - To a limited extent. Such menus tends to use vocabularies which are very limited - e.g. the digits, "Yes", and "No" - and within the designers' control, which greatly simplifies the problem. On the other hand, the programs must deal with an uncontrolled space of all kinds of voices and accents. The voice activated directory assistance programs used by telephone companies, which must deal with a large and changing vocabulary are certainly AI programs.
   - This is borderline. There is something to be said for viewing these as intelligent agents working in cyberspace. The task is sophisticated, the information available is partial, the techniques are heuristic (not guaranteed optimal), and the state of the world is dynamic. All of these are characteristic of intelligent activities. On the other hand, the task is very far from those normally carried out in human cognition.

2. **Question 2.4** Many of these can actually be argued either way, depending on the level of detail and abstraction.

   - A. Partially observable, stochastic, sequential, dynamic, continuous, multi-agent.
   - B. Partially observable, stochastic, sequential, dynamic, continuous, single agent (unless there are alien life forms that are usefully modeled as agents).
   - C. Partially observable, deterministic, sequential, static, discrete, single agent. This can be multi-agent and dynamic if we buy books via auction, or dynamic if we purchase on a long enough scale that book offers change.
   - D. Fully observable, stochastic, episodic (every point is separate), dynamic, continuous, multi-agent.
   - E. Fully observable, stochastic, episodic, dynamic, continuous, single agent.
   - F. Fully observable, stochastic, sequential, static, continuous, single agent.
   - G. Fully observable, deterministic, sequential, static, continuous, single agent.
   - H. Fully observable, strategic, sequential, static, discrete, multi-agent.

3. **Question 2.7** Refer to tables 1 and 2.

4. **Question 3.8**

   - Any path, no matter how bad it appears, might lead to an arbitrarily large reward (negative cost). Therefore, one would need to exhaust all possible paths to be sure of finding the best one.

function GOAL-BASED-AGENT(percept) **returns** an action
persistent: *state*, the agentfs current conception of the world state
model, a description of how the next state depends on current state and action
utility function, a description of the agentfs utility function
plan, a sequence of actions to take, initially empty
action, the most recent action, initially none
state⟵ UPDATE-STATE(state, action, percept ,model)
if GOAL-ACHIEVED(state,goal) **then return** a null action
if plan is empty **then**
plan ⟵ PLAN(state,goal ,model)
action ⟵ FIRST(plan)
plan ⟵ REST(plan)
return action

Table 1: A goal-based agent.

function UTILITY-BASED-AGENT(percept) **returns** an action
persistent: *state*, the agentfs current conception of the world state
model, a description of how the next state depends on current state and action
utility function, a description of the agentfs utility function
plan, a sequence of actions to take, initially empty
action, the most recent action, initially none
state⟵ UPDATE-STATE(state, action, percept ,model)
if plan is empty **then**
plan ⟵ PLAN(state,utility function,model)
action ⟵ FIRST(plan)
plan ⟵ REST(plan)
return action

Table 2: Utility-based agent.

- Suppose the greatest possible reward is c. Then if we also know the maximum depth of the state space (e.g. when the state space is a tree), then any path with d levels remaining can be improved by at most cd, so any paths worse than cd less than the best path can be pruned. For state spaces with loops, this guarantee doesn't help, because it is possible to go around a loop any number of times, picking up c reward each time.
- The agent should plan to go around this loop forever (unless it can find another loop with even better reward).
- The value of a scenic loop is lessened each time one revisits it; a novel scenic sight is a great reward, but seeing the same one for the tenth time in an hour is tedious, not rewarding. To accommodate this, we would have to expand the state space to include a memory-a state is now represented not just by the current location, but by a current location and a bag of already-visited locations. The reward for visiting a new location is now a (diminishing) function of the number of times it has been seen before.
- Real domains with looping behavior include eating junk food and going to class.

5. **Question 3.9**

- a. Here is one possible representation: A state is a six-tuple of integers listing the number of missionaries, cannibals, and boats on the first side, and then the second side of the river. The goal is a state with 3 missionaries and 3 cannibals on the second side. The cost function is one per action, and the successors of a state are all the states that move 1 or 2 people and 1 boat from one side to another.
- c. It is not obvious that almost all moves are either illegal or revert to the previous state. There is a feeling of a large branching factor, and no clear way to proceed.