Washington State University
School of Electrical Engineering and Computer Science
Fall 2017

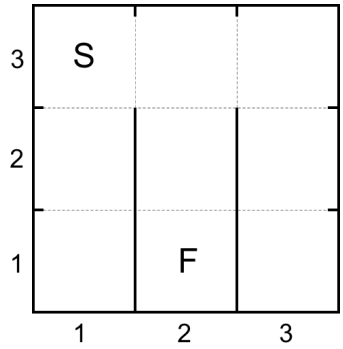
CptS 440/540 Artificial Intelligence
Homework 1
Due: September 7, 2017 (11:59pm)

General Instructions: Put your answers to the following problems into electronic form. Your answers to problems 1-3 should be a PDF document. Your answer to problem 4 should be an Agent.h and Agent.cc file, perhaps with a readme.txt file. Please collect all these homework files into one zip file and submit as an attachment under Content → Homework 1 for the course CptS 440 Pullman (all sections of CptS 440 and 540 are merged under the CptS 440 Pullman section) on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but we will only grade the most recent entry submitted before the above deadline.

1. Go to http://www.mitsuku.com and engage this chatbot in a conversation using the "Chat to me using Flash" option. Enter at least 10 responses, one of which should be "Do you know any Chuck Norris jokes?". When you are done, you can recover your full conversation by clicking "Click here to read the chatlog" below the conversation window. Include your full conversation in the homework and give a short explanation of why you think your conversation shows that Mitsuku does or does not pass the Turing Test.
    a. *CptS 540 students only*. Also ask Mitsuku the question about chromosomes shown as an example in the recent paper "Moving Beyond the Turing Test with the Allen AI Science Challenge" by Carissa Schoenick et al. Be sure to also indicate if Mitsuku gets the correct answer.

2. Consider the task of designing your own chatbot. Describe the performance measure, environment, actuators and sensors of your chatbot. For each of the seven environment properties we discussed in class, indicate how this environment would be described along these seven dimensions and give a short explanation as to why. Finally, of the four types of agents (simple reflex, model-based reflex, goal-based, utility-based), indicate which you think would be best for your chatbot and give a short explanation as to why.

3. Consider the following maze problem, where the goal is to get from the Start location in (1,3) to the Finish location in (2,1). There are four moves available at each location: up, down, left, right. If a move runs into a wall, then you will stay in the same location.



   a. Draw the entire search tree to a depth of 2. As you draw child nodes from left to right, consider actions in the order: up, down, left, right (label branches with U, D, L, R). Note this is a search tree, not a search graph, so locations can be revisited.
   b. In your search tree from part (a), expand the one leaf node that will lead to a solution.
   c. How many nodes would be generated applying breadth-first tree search to this problem? Note that not all generated nodes will be shown in part (a).
   d. Would depth-first tree search terminate on this search problem? Why or why not?
   e. How many nodes would be generated applying iterative-deepening search to this problem? Show your work.

4. A C++ Wumpus World simulator, following the rules from the textbook, is available at http://www.eecs.wsu.edu/~holder/courses/AI/wumpus/wumpus-2.7.zip. For this problem, you will implement a model-based reflex agent to play the Wumpus World game using the following "reflexes" (**and only these**).
   a. If the agent observes the Glitter percept, it should execute the GRAB action.
   b. If the agent has the gold and is in location [1,1], it should execute the CLIMB action.
   c. If the agent observes the Stench percept, and it has an arrow, then it should execute the SHOOT action.
   d. If the agent observes the Bump percept, it should execute the actions TURNLEFT and then GOFORWARD.
   e. Otherwise, the agent should choose an action randomly from TURNLEFT, TURNRIGHT and GOFORWARD.

   Begin by reading the readme.txt file that comes with the simulator. You will be modifying the Agent.h and Agent.cc files to implement your agent. You may also submit your own readme.txt file containing any information you think we may need about your agent. Your agent should not require any user input. Your agent will be tested by copying only your Agent.h and Agent.cc files into a fresh copy of the simulator code, compiling and running it on several test worlds of varying sizes. Your grade will be based on satisfying the above requirements and good programming style.