Washington State University
School of Electrical Engineering and Computer Science
Fall 2019

CptS 440/540 Artificial Intelligence
Homework 9
Due: November 7, 2019 (11:59pm)

General Instructions: Your submission for this homework should be a zip file containing your Agent.h and Agent.cc files (or a PyAgent.py file), and an optional readme.txt file. Put your files into one zip file and submit as an attachment under Content → Homework 9 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.

For this homework you will implement an agent to play the Wumpus World game that utilizes concepts from search, logic, and uncertainty. Specifically,

1. Your agent should satisfy all the requirements from Homework 5 (i.e., items 1-3 in Homework 5) with the exceptions described below. You may start from my Homework 5 solution, if you want.
2. The test worlds will all have the same size of 5x5. The gold may be collocated with the Wumpus or a pit (or both). There may not be a safe path to the gold. As with all Wumpus worlds, the (1,1) location never has the gold, the wumpus, or a pit.
3. Since there can now be pits in the world, your agent should compute and output the probability of a pit in each location each time the Process method is called. The probabilities should be correct based on all possible evidence gathered and inferred so far. The last thing your agent should do before returning an action from the Process method is to print out the P($Pit_{x,y}$=true) for all 25 locations (x,y) in the format in the box below. Probabilities should be printed to two decimal places, with one space between them.

```
P(pit):
 0.20 0.20 0.20 0.20 0.20
 0.20 0.20 0.20 0.20 0.20
 0.31 0.20 0.20 0.20 0.20
 0.00 0.86 0.20 0.20 0.20
 0.00 0.00 0.31 0.20 0.20
```

E.g., these probabilities are correct for an agent that has visited (1,1), (1,2) and (2,1), and perceived a breeze in (1,2) and (2,1).

4. As in Homework 5, your agent should eventually visit all safe locations, but a safe location is now defined as a location with P(Wumpus=true) = 0, and P(Pit=true) < 0.5. Note that this may lead your agent to its death by falling into pits, but your agent should then set P(Pit=true) = 1 for these locations for future tries on the same world.

5. Submit a zip file with your Agent.h and Agent.cc files, or PyAgent.py file, along with an optional 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, or PyAgent.py file, into a fresh copy of the simulator code, and compiling and running it on several test worlds. Your agent will be given 10 tries on each test world. Your grade will be based on satisfying the above requirements, *your agent's scores on the test worlds*, and good programming style (see the course website for links to style guides).

Hint: Below is an algorithm for computing the probability of a pit in all locations on the frontier, where the frontier consists of all locations adjacent to locations for which we know a pit does or does not exist.

known = {information about pits in locations for which the probability of a pit is 0 or 1}
breeze = {information about breezes in visited locations}
frontier = {random variables for pits in locations adjacent to locations in *known*}

foreach location (x,y) in *frontier*
{
  $P(Pit_{x,y}=true) = 0.0$
  $P(Pit_{x,y}=false) = 0.0$
  frontier' = *frontier* – $\{Pit_{x,y}\}$

  foreach possible combination *C* of pit=true and pit=false in *frontier'*
  {
    P(*frontier'*) = $(0.2)^T * (0.8)^F$, where T = number of pit=true in *C*, and
                                 F = number of pit=false in *C*
    if *breeze* is consistent with ($C + Pit_{x,y}=true$)
    then $P(Pit_{x,y}=true)$ += P(*frontier'*)
    if *breeze* is consistent with ($C + Pit_{x,y}=false$)
    then $P(Pit_{x,y}=false)$ += P(*frontier'*)
  }

  $P(Pit_{x,y}=true)$ *= 0.2
  $P(Pit_{x,y}=false)$ *= 0.8
  $P(Pit_{x,y}=true) = P(Pit_{x,y}=true) / (P(Pit_{x,y}=true) + P(Pit_{x,y}=false))$ *// normalize*
}