

CSI4106: Introduction to Artificial Intelligence Fall 2017

Assignment 2

Handed in on: November 1st

Due on: November 19th (midnight)

Learning objective: Implement a knowledge base agent in propositional logic and Python

Requirements: The assignment MUST be done in a group of two.

In this assignment, you will implement a simulation of the Wumpus World, build several agents that interact with the world, and collect statistics on their performance. The assignment is broken into 4 parts to guide you along. You must do all four parts.

Part 1

Read carefully <https://github.com/aimacode/aima-python/blob/master/logic.ipynb>

In this assignment, you can reuse any code from AIMA-Python that you see fit to solve the problem. Do not forget to reference it.

Part 2

Write a function that creates a Wumpus World according to the rules described in this assignment. Your simulation should represent the 16 rooms of the Wumpus World, and should

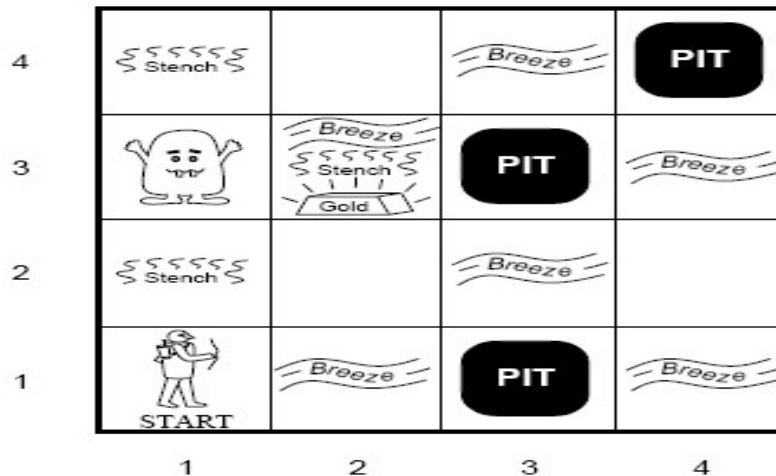
- (a) pick a room for the wumpus,
- (b) pick a room for the gold (not the same room as the wumpus is in),
- (c) generate pits with probability .2 (but never put pits in room [1,1] or in the room containing the gold), and
- (d) determine the sensations (breeze, stench, and glitter) in each room. (The Wumpus World we describe should be the same as that of Russell & Norvig)

Creating the Wumpus World defines the world state, which consists of: (a) an array specifying the contents of each room, (b) the [x,y] coordinates of the room occupied by the agent, (c) the direction the agent is facing (up, down, left, right), and (d) the sensations available to the agent (breeze, stench, glitter, bump, scream).

Part 3

Implement a knowledge-based agent using the pseudo-code shown in the lecture notes. The objective of your agent is to find the gold without being killed. Your agent starts in cell [1,1]. Your agent must use propositional logic to keep track of all relevant facts in its knowledge base.

Your simulation 0 should be as depicted in the following figure:



The actions are: move forward in the current direction, turn left, turn right, grab object in room, and fire an arrow. Firing an arrow will shoot the arrow in the direction the agent is facing, and the arrow will continue until it hits a wall or hits the wumpus. If the wumpus is hit it will die. Killing the wumpus removes the wumpus and all stenches from the world. "Grabbing the object" will have no effect unless the room contains gold, in which the gold is picked up. When the agent tries to walk off the board, it does not move, and it senses a bump. When the wumpus is killed, a scream is generated in all rooms.

The payoffs resulting from an action are as follows:

- +1000 if the gold is picked up;
- 1000 if the agent enters a room containing a pit or the wumpus;
- 10 if an arrow was shot;
- 1 for other action.

You must implement your exploration in a format similar to assignment 1 but this time using the knowledge base. As your agent explores the Wumpus world, it adds percepts to the KB and performs inference on the KB to determine if neighboring rooms (the fringe) are safe. You might consider prioritizing the safety of the fringe rooms. In particular you must

- a) Write a function that identifies the set of possible actions from each given cell
- b) Write a function that takes the current world state and an action of the agent, and returns the new world state and a payoff resulting from the action.

For each cell, before performing any operation, your agent must

1. Update the knowledge base based on its percepts and inferences
2. Determine which cells are safe using propositional inference
3. Determine if the current cell contains the goal.

Part 4

Run **several simulations** (10) of an exploration agent (the most intelligent you can think of) with different environments and report the results of these simulations (payoffs,

solution depth, Time, whether it is optimal, whether it is complete). The first reported results should be the one of simulation 0 as depicted in the figure above.

You can decide to stop a simulation after a number of iterations without a solution.

You should run the simulations 10000 times to determine the **average cumulative payoff** of your agent.

Example of exploration agents:

- DumbAgent: The rules of the agent should be as follows:
 - (1) If the agent senses glitter in the current room, it should pick up the gold.
 - (2) If the agent did not shoot an arrow on the last time step, and the agent senses a stench in the current room, it should shoot an arrow.
 - (3) Otherwise, the agent should choose one of the remaining three actions (turn left, turn right, move forward) at random.
- IntelligentAgent: You can optimize your agent by trying out different programs and seeing which one produces the highest average cumulative payoff (over 10000 runs).

Submission:

You must submit a zip file of your code and report on Brightspace.

The report must:

- briefly and clearly explain your exploration agent. For example, provide a ranked list of condition-action rules (i.e., the top ranked rule is chosen if it applies). You must also clearly explain the inference method(s) used by your agent by referring to the algorithms seen in class.
- provide the average cumulative payoff of your agent.

Marking guidelines

- 1- Working and correct implementation following the guidelines
- 2- Good programming practices (appropriate use of classes, methods, parameters, comments, etc.)
- 3- Highest possible average cumulative payoff as compared to the class. The dumbAgent will be assigned half the marks if correctly implemented.