IFN 680

The Wumpus World

Conditions:

1. probability of room with Wumpus and pit is 0.2
2. if finding all available room without finding gold (and still alive), return to the original room, the game ended.

PWi,j: a Boolean variable represents location (i, j) contains a pit/Wumpus or not.

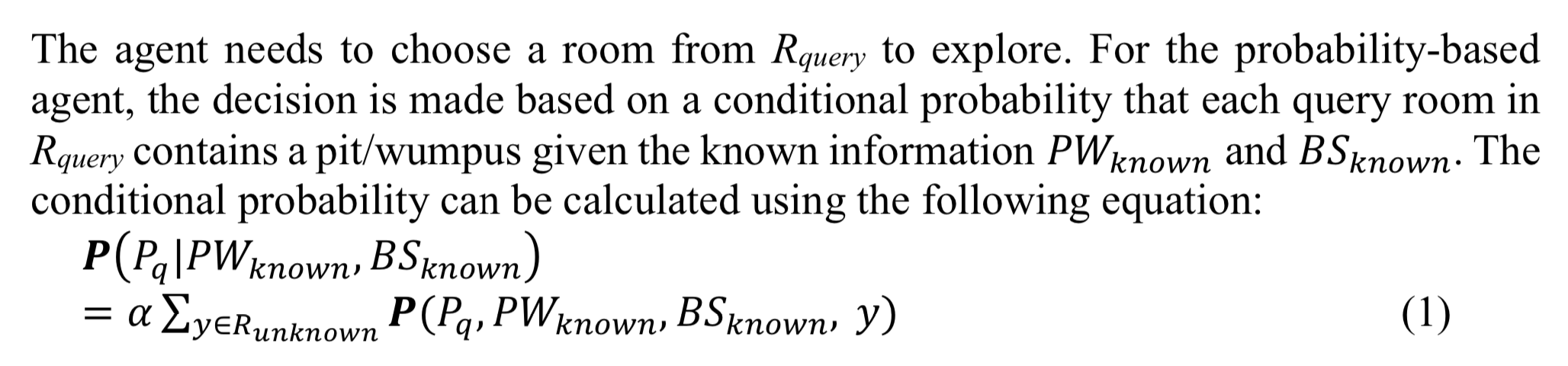
True: contains Pit/Wumpus

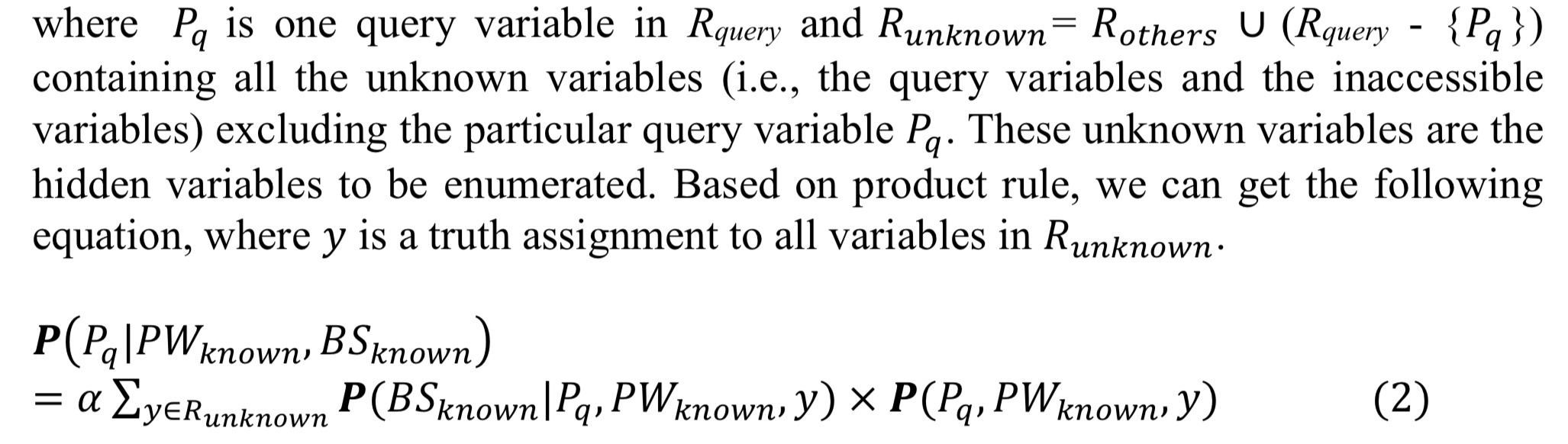
False: vice versa

BSi,j: represents whether location (i, j) contains breeze/stench.

* *Rknown* : a set of rooms that the agent has visited so far. These rooms are safe and the truth value of the pit\_wumpus variables representing these rooms is ‘False’.
  + 𝑃𝑊*known* = {𝑃𝑊1,4 = 𝐹𝑎𝑙𝑠𝑒, 𝑃𝑊2,4 = 𝐹𝑎𝑙𝑠𝑒, ...... ,𝑃𝑊3,2 = 𝐹𝑎𝑙𝑠𝑒}
  + 𝐵𝑆𝑘𝑛𝑜𝑤𝑛 = {𝐵𝑆3,3 = 𝑇𝑟𝑢𝑒, 𝐵𝑆4,4 = 𝑇𝑟𝑢𝑒, 𝐵𝑆1,4 = 𝐹𝑎𝑙𝑠𝑒, ... ... , 𝐵𝑆3,2 = 𝐹𝑎𝑙𝑠𝑒}
* *Rquery* : a set of rooms that adjacent to the agent's current location and that have not been visited by the agent. This part is unknown to the agent, but accessible. We call the rooms in this part query rooms.
* *Rothers*: a set of rooms that the agent has not visited and that are not adjacent to the agent's current location.

**The agent needs to choose a room from *Rquery* to explore.**

****

****

𝜬(𝐵𝑆known |𝑃q , 𝑃𝑊known , 𝑦) is provided as **consistent(self,known\_BS,event)**

known\_BS contains the known breeze\_stench values to the visited rooms and event is a configuration to all rooms in the cave with the known truth values to the variables in 𝑃𝑊known. 𝜬(𝑃*q*,𝑃𝑊*known* ,𝑦) is the joint probability 𝑘𝑛𝑜𝑤𝑛 𝑞 𝑘𝑛𝑜𝑤𝑛 distribution.

The ‘Fixed Board’ option is designed for the convenience of debugging your code.

If you choose the probability-based agent, you need to **specify the maximum probability threshold that a room contains a pit/wumpus**. If the threshold is set **higher**, the robot **will have a higher chance to be killed by the wumpus** or **fall into a pit**. If the threshold is set **too low**, the robot will be **too cautious to explore more rooms** and thus may not be able to find the gold. When the threshold is **set to 0**, the probability- based agent **will perform the same as the logic-based agent**.

**A game ends with three possible outcomes:**

1. the robot found the gold
2. the robot returned to the starting position empty handed without finding the gold
3. the robot was dead (i.e., killed by the wumpus or fell into a pit)

**determine next move**

The provided code allows the agent to conduct a resolution based reasoning to determine next move mainly using three functions in the class, kb\_add() to build its knowledge base, check\_safety() to check whether or not a room is safe (i.e., there is no pit/wumpus in the room), and next\_room() to determine a safe room (next\_room() is in Python file logic\_based\_move.py).

**Useful Variables:**

**Robot variables**o **self.cave**: it is an object of Cave, it is the cave board that the agent is in (‘self’ means this is an instance variable of this class). The two functions that you are required to complete are methods of Robot class. This means that, in the two methods you can get access to the cave via variable **self.cave.**

o **self.visited\_rooms**: it is a set object, it contains a set of (column, row) pairs of all the rooms visited by the agent so far.

o **self.jdp\_PW**: it is an object of JointProbDist. It provides the joint probability distribution of pit/wumpus in the given cave environment.

o **self.max\_pit\_probability**: it is a floating number, it is the maximum probability threshold specified by the user.

o **self.current\_position**: it is a tuple (column, row), it is the location of the room that the agent is in currently.

**Robot functions**o **self.check\_safety(self, column, row):** return True if the room (column, row) is safe (i.e., there is no pit/wumpus in the room), otherwise False.

o **self.consistent(self,known\_BS,event):** return 1 if the truth values in known\_BS are consistent with the values in event, 0 otherwise  
▪known\_BS: a dict containing the visited rooms with their corresponding truth value for breeze/stench;  
▪event: a dict containing the rooms each with an instantiated truth value for pit/wumpus.

o **self.observation\_pits(self,observed\_locations):** return a dict containing a set of var: val pairs indicating whether these rooms contain a pit or not. var is the variable name of a room, val is a truth value.

▪observed\_locations: a set of (column, row) pairs, which have been visited.

o **self.observation\_breeze\_stench(self,observed\_locations):** return a dict containing a set of var: val pairs indicating if each of the rooms has a breeze/stench. var is the variable name of a room, val is a truth value.

▪observed\_locations: a set of (column, row) pairs, which have been visited.

• **Class Cave**This class defines the board for the 2-dimensional cave environment. Each room in the cave is specified by a pair of column and row, (column, row) as showed in Figure 4.

o **self.getsurronding(column, row):** You can use this function via an object of Cave to return the adjacent rooms of a specific location (column, row), e.g., self.cave.getsurrounding(x,y)returns a list of adjacent rooms of (x,y)in cave cave, each room is a pair of (column, row).

cannot use variables \_goldCoor, \_wumpusCoor, and \_pitCoors, which are protected variables and cannot be used in other classes.

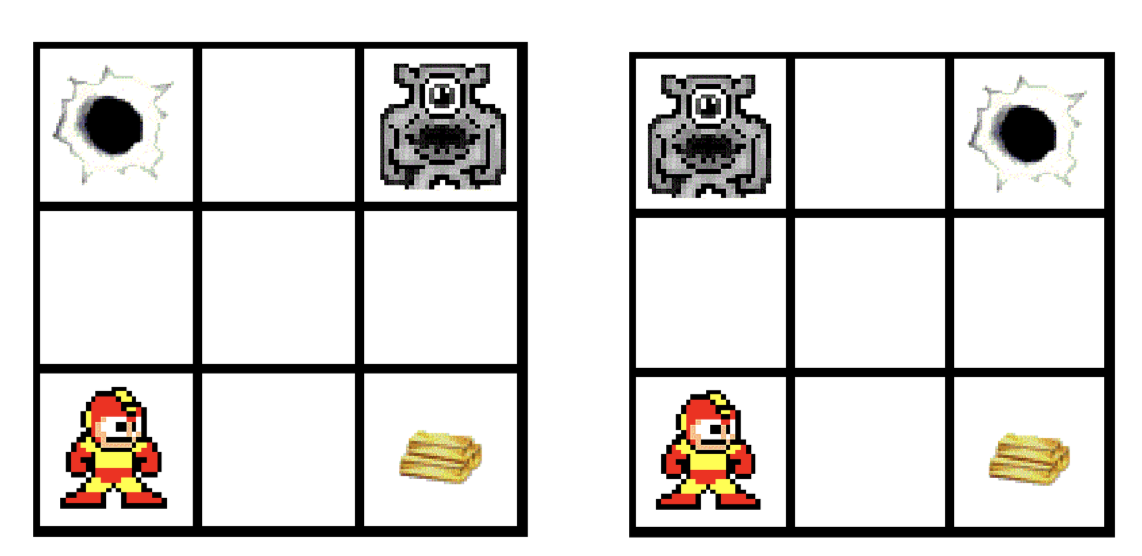
**5. Requirements**

**Code Requirements:**

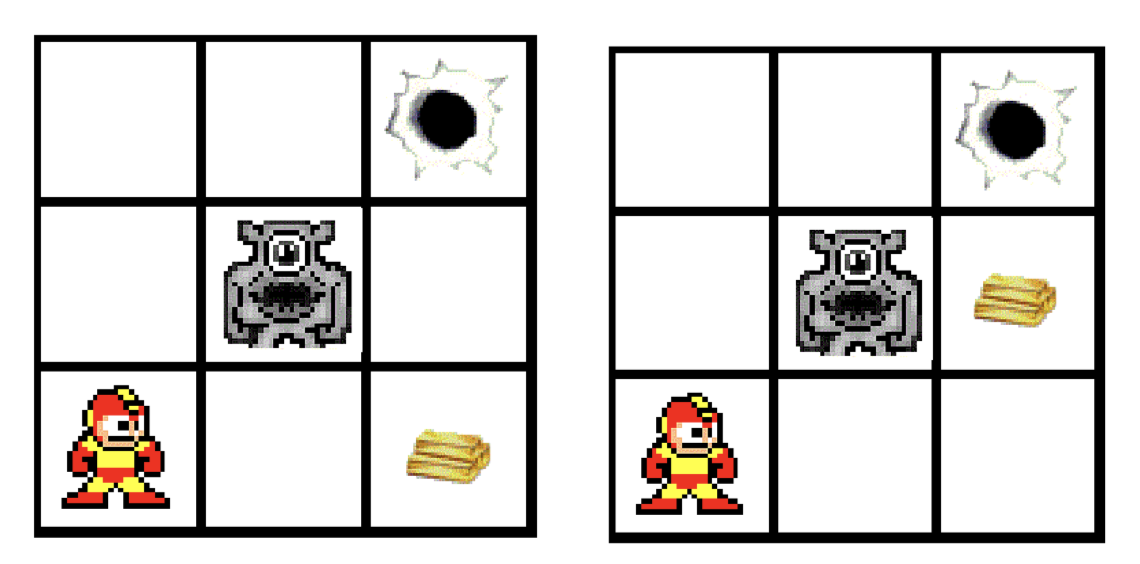
1. Your task is to complete the two required functions in probability\_based\_move.py.
2. You must insert your name and your student ID at the beginning of both files probability\_based\_move.py and the\_wumpus\_world.py. Fail to do so will cause mark deduction. For a team, all team members’ name and ID should be added.
3. Your code must implement a probability based approach to choose a room for the next move. You are encouraged to implement the approach explained in Section 2 (explained in week 5 lecture as well). You can implement other probability based approaches. But it will not bring any extra marks (it may cause mark deduction if it is not correctly implemented).
4. Again, in your code, you cannot use any of the three protected variables defined in class Cave: \_goldCoor, \_wumpusCoor, and \_pitCoors
5. Your code **must be well-presented** and **easy to understand**. **Concise inline comments** are required to explain the purpose of significant code segments and **a brief header comment** is required for each of your methods.
6. 6)  You are required to write a report which contains the following content:

**Report Requirements:**

1. **A statement of completeness** 
   1. Provide a list of functions completed in your probability\_based\_move.py file
   2. If you have modified any part of the code which was not allowed to modify, list the changed part/parts and provide your justification about why you need to do so.
   3. **A workload distribution** over team members in the case of a team work.
2. **Probability-based approach**  
   Indicate which probability-based approach has been implemented in your code, the approach explained in Section 2 or some other approaches. **Provide a brief description** to the approach that you have implemented. You **need to cite at least one reference for the approach** **if you didn’t implement the approach in Section 2**. A **half page (at most one page)** should be sufficient for describing your approach.
3. **Test cases  
   Provide four example scenarios** to show that the **logic-based agent fails to find the gold**, but your **probability-based agent can**. The scenarios **can be 3x3 or 4x4 board**, **one pit or more pits**. You **need to include the screenshots** of the cave boards to show the comparison for each of the examples. The four board configurations must be substantially different. For example, the following two boards are actually identical given that we treat pits and the wumpus the same.



The following two boards are also very similar since the pit and the wumpus positions are identical.



Your code will be tested using some board settings. The four example boards provided by you will also be used to test your code.

**One single zip file containing the following files should be submitted:**

1. A report in pdf format.
2. Your Python files, probability\_based\_move.py and

the\_wumpus\_world.py.