Science and Engineering Faculty
IFN680 – Introduction to Artificial Intelligence

Assignment 1

The Wumpus World – a Probability-based Agent

(Semester 2, 2016)

Team Members
Frederick Heidrich, N9753095
Anh Khoa, Tran, N09540377

18 September 2016

Lecturers:
Dr. Frederic Maire
Dr. Yue Xu

Tutor:
Mr Dominic Francis

Table of Contents

1 Approach

We decided to adopt the conditional independence approach according to (Russell, Norvig, & Davis, 2010, p.500). The essence of this method is instead of considering all unvisited rooms in our joint distribution, we only consider all unknown rooms outside the query rooms (called fringe in our code) independent of known rooms. Below is the describe of the mathematics behind this approach.
Starting with the given equation:

$$P\left(P_q \middle| known\_PW, known\_BS\right)$$
$$= \alpha \sum_{unknown} P(known\_BS|\, P_q, unknown, known\_PW)P(P_q, unknown, known\_PW)$$

We applied production rule on the unknown variable, which gives us:

$$= \alpha \sum_{fringe} \sum_{other} P(known\_BS|P_q, fringe, other, known\_PW)P(P_q, fringe, other, known\_PW)$$

Fringe is defined as all the available rooms the agent can access directly or indirectly. And other is the rest of *unknown* that does not include *fringe*.

Because *known_BS* and other are independent from each other:

$$= \alpha \sum_{fringe} \sum_{other} P(known\_BS|P_q, fringe, known\_PW)P(P_q, fringe, other, known\_PW)$$

Rearranging the terms:

$$= \alpha P(known\_BS)P(P_q) \sum_{fringe} P\left(known\_BS|P_q, fringe, known_{PW}\right) P(fringe) \sum_{other} P(other)$$

$$= \alpha' P\left(P_q\right) \textstyle\sum_{fringe} P\left(known\_BS|P_q, fringe, known_{PW}\right) P(fringe) \qquad (1.1)$$

Notice that in the last term, we folded *P(known_BS)* and alpha into the normalizing constant and used the fact that $\sum_{other} P(other)$ equals 1.
That is, by contrast to a summation of all unknown rooms in the original equation, this new summation only includes 4 or 8 terms over the frontier variables, with four being there are 2 rooms in the fringe and 8 being there are 3 rooms in the fringe.

Given the scenario of a 4x4 maze where there are still 12 unknown rooms, while the original equation executes a summation over 2^{12} = 4096 terms, equation (1.1) only executes a summation over 8 terms. For this reason, we also proposed that our approach speeds up the computational process and, therefore, makes the probability-based approach even more superior than the logic-based approach.

2 Example Scenarios

We included 4 out of our vigorous tests to illustrate that the probability-based approach is more efficient than the logic-based approach. Specifically, our examples will illustrate that the probability-based approach will either allow the agent to reach the gold with less steps compared to the logic-based, or allows the agent to reach the gold while the logic-based cannot.

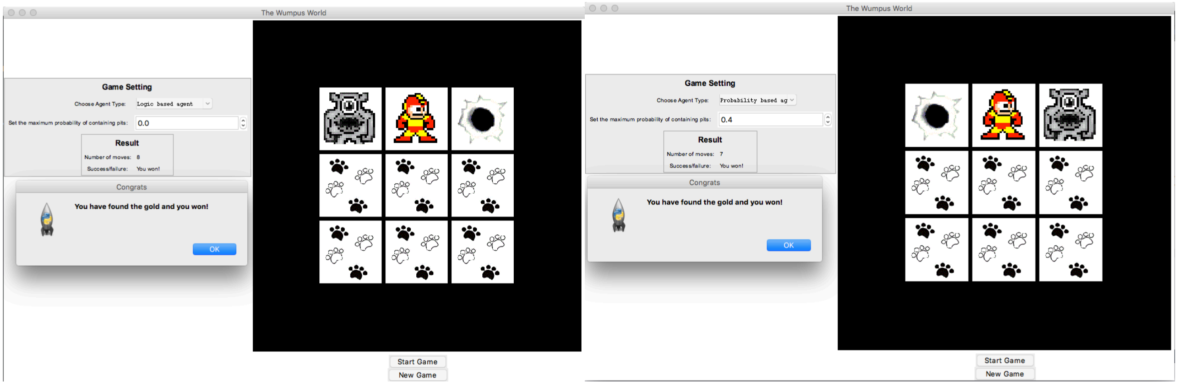

Figure 1. Left figure represents logic-based approach, right figure represents probability-based approach

In the example in Figure 1, while both approaches allow the agent to reach the gold, the agent using the probability-based only needed to take 7 steps while the agent using the logic-based needed to take 8 steps.

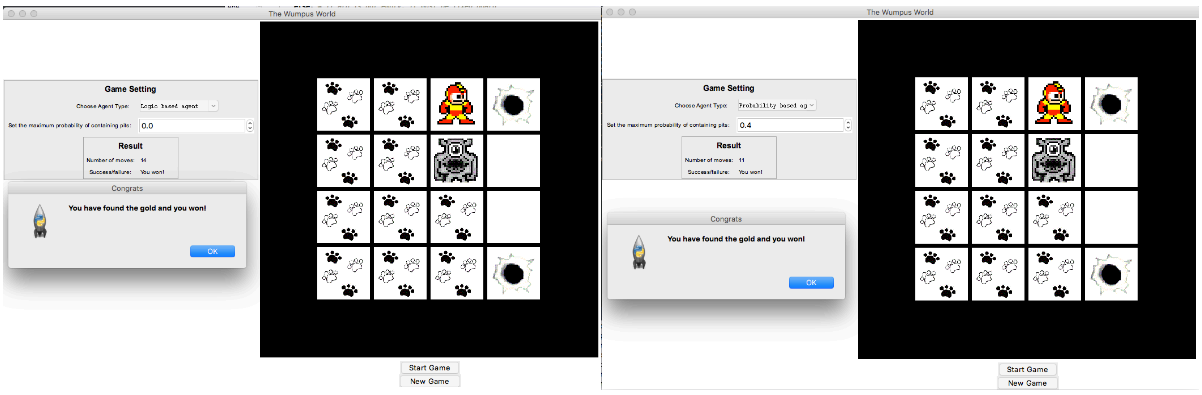

Figure 2. Left figure represents logic-based approach, right figure represents probability-based approach

In the example in Figure 2, while both approaches allow the agent to reach the gold, the agent using the probability-based only needed to take 11 steps while the agent using the logic-based needed to take 14 steps.

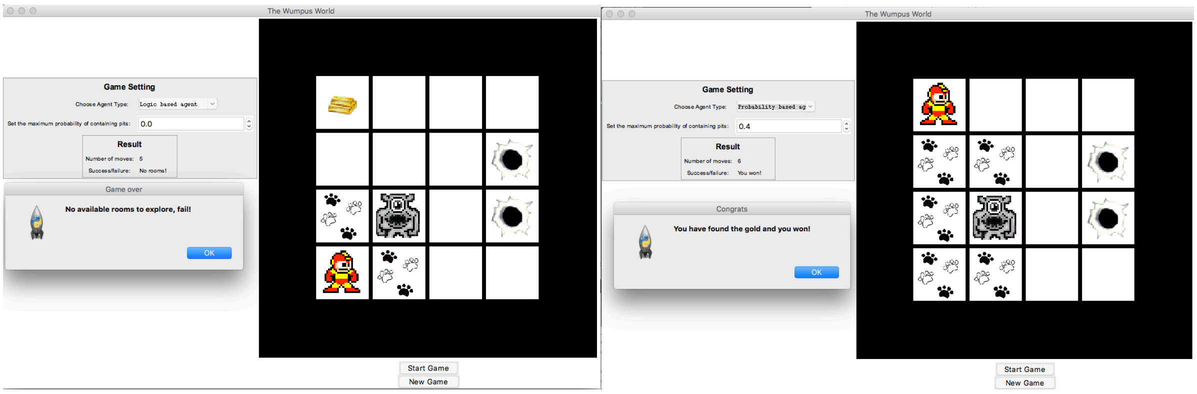

Figure 3. Left figure represents logic-based approach, right figure represents probability-based approach

In the example in Figure 3, the logic-based agent failed to reach the gold and declared loss after 3 steps. On the other hand, the probability-based agent was able to reach the gold after 6 steps.

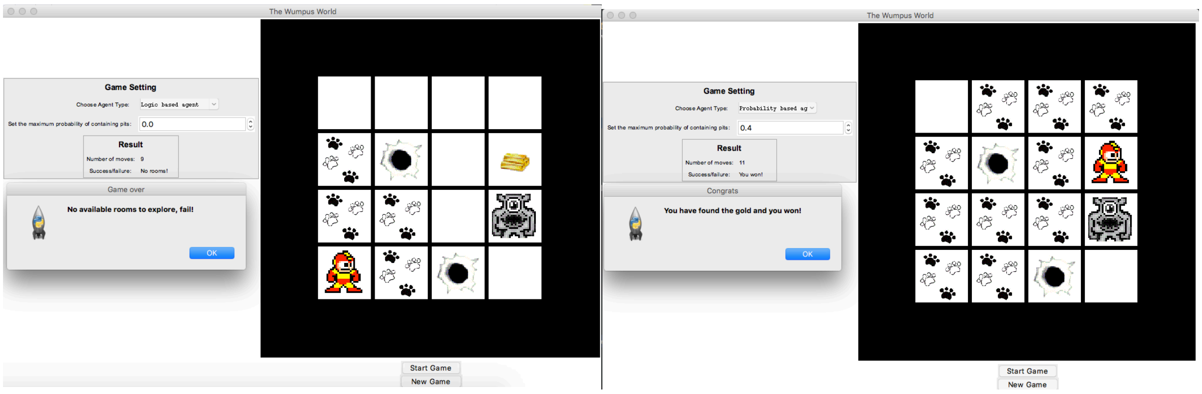

Figure 4. Left figure represents logic-based approach, right figure represents probability-based approach

In example 4, the logic-based agent failed to reach the gold and declared loss after 9 steps. On the other hand, the probability-based agent was able to reach the gold after 11 steps.

3 Limitations of Probability-based Approach

Apart from the aforementioned scenarios, we also decided to include a few scenarios where the probability-based agent failed to choose a safe room or the room with the gold.

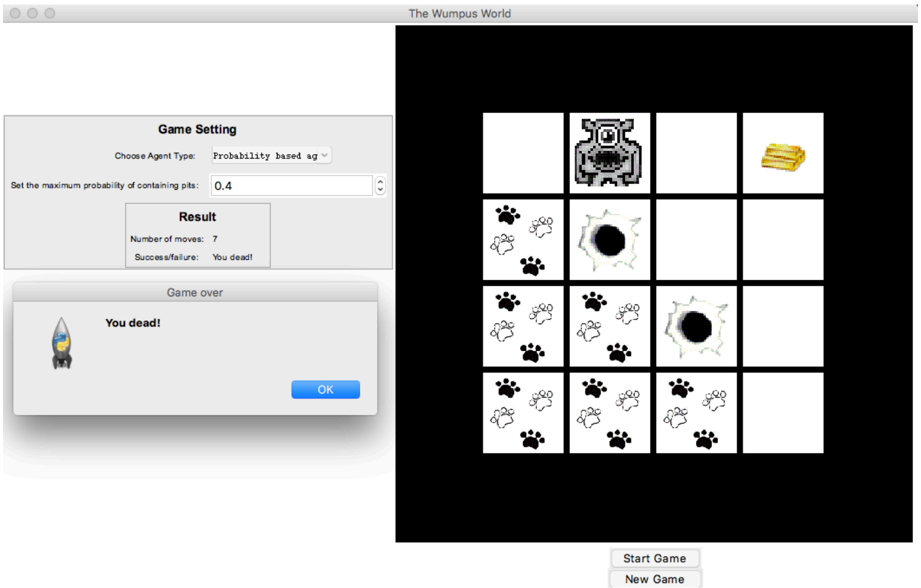

Figure 5. 4x4 maze with wumpus/pits in room (1,2), (2,2) and (3,3). Gold is in room (1,4)

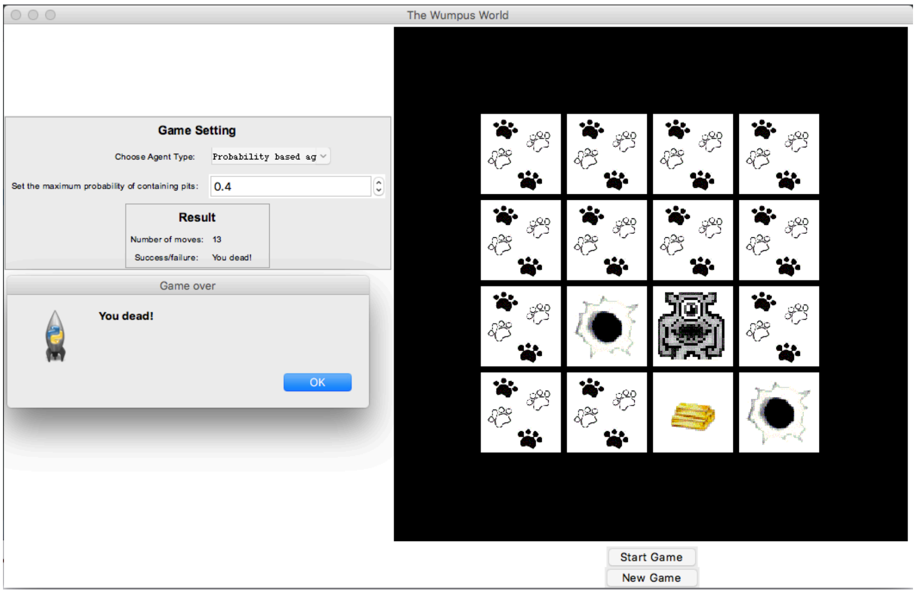

Figure 6. 4x4 maze with wumpus/pits in room (3,2), (3,3) and (4,4).
Gold is in room (4,3)

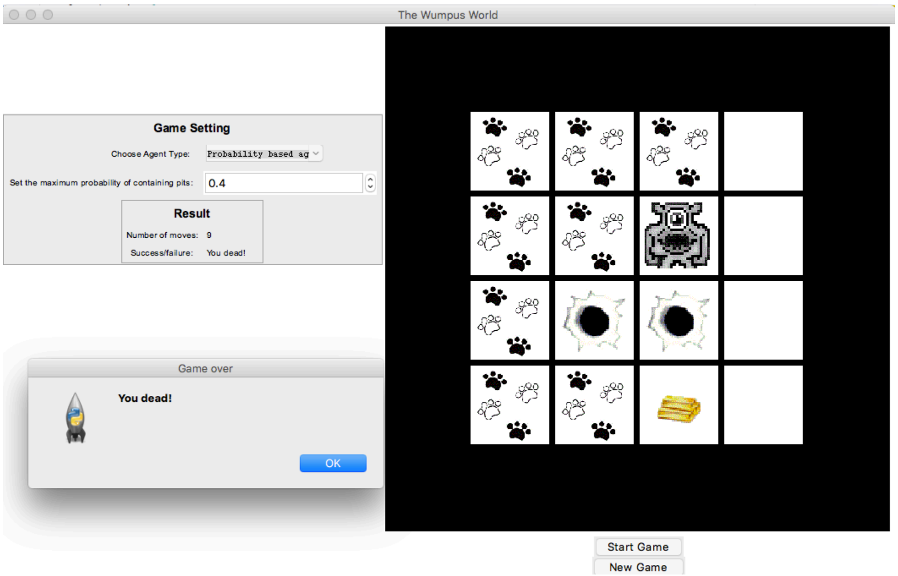

Figure 7. 4x4 maze with wumpus/pits in room (2,3), (3,2) and (3,3).
Gold is in room (4,3)

To provide a perspective on why there is such failure with the probability-based approach, we need to look into the path of the agent in each scenario, as well as the state of the agent right before it made the last and losing move.

In the scenario in Figure 5, the only safe path for the agent is almost surrounded by the wumpus/pits. Only when the agent has reached room (4,4) that it is possible to determine safe path. In this scenario, the agent followed this path: (1,4) -> (2,4) -> (3,4) -> (2,3) -> (1,3) -> (1,2). As the agent reached room (1,2), it had determined that there is a breeze in each of these rooms: (1,2), (2,3), and (3,4), and no pit or wumpus has been detected.

The fringe in this scenario included room (1,1), (2,2), (3,3) and (4,4). By using either equation (1.1) or the given equation, the agent would detect that all room has an equal chance of 0.0256 of having a pit/wumpus. This means the chance that the agent chooses the wrong room was actually 50%.

The same logic can be applied for the scenarios in Figure 6 and Figure 7.

4 Statement of Completeness

We have completed the joint distribution part of code in the function called *Pit-Wumpus_probability_distribution*(*self*, *width*, *height*) which returns a joint probability distribution object based on the fringe rooms in our approach and the function *next_room_prob*(*self*, *column*, *row*) which uses the joint probability distribution of function 1 to select a room based on the variable *max_pit_probability* given by the game. Both functions are completed in the file *probability_based_move.py*.

Both Fred and Khoa worked on the development of the conditional approach that results in Equation 1.1. While Fred focused on turning the algorithm to code in the *next_room_prob*(*self*, *column*, *row*) function, Khoa focused on creating the test environment that resulted in the 4 examples and limitations of the probability-based approach. Both contributed equally to the report.

5 Reference

Russell, S., Norvig, P., & Davis, E. (2010). *Artificial intelligence*. Upper Saddle River, NJ: Prentice Hall.