Artificial Intelligence Project 2: Wumpus World

Group 11: Lisa Peters, Janette Rounds & Monica Thornton

Abstract

In this work, we examine the concepts of logical agents, knowledge representation, reasoning, and inference via the Wumpus World problem. This project focuses on the design, implementation, and evaluation of two agents who use disparate strategies to explore the Wumpus world. One of the agents perceives the world through its reactions to the environment it is navigating, and the second agent uses first-order logic (FOL) to try to reason about the environment. These explorers were evaluated on worlds that vary in size from 5×5 to 25×25, and a number of performance measures were gathered as the explorers navigated the different worlds. These measures were employed to compare the relative performance of each agent. Contrary to our expectations, our FOL agent did not outperform the reactive agent. While there were a few exceptions where the FOL agent did better than the reactive agent, these were few, and overall, the FOL agent did slightly worse than the reactive agent.

1. Introduction

This document discusses our implementation of the Wumpus World, a game in which an explorer traverses a two dimensional square grid in order to find gold. The original version of this game, as implemented by Gregory Yob, used a "squashed dodecahedron" as a map, but otherwise Yob's version of the game is largely similar to the version of the game presented in Russell and Norvig (Ahl & North, 1979; Russell & Norvig, 2003). The version of the game implemented for this work is based strongly on the specifications presented by Russell and Norvig, with a few key differences. These differences are mainly additional difficulties introduced to make the problem harder for explorers. In addition to the formidable wumpii and pits of the original world, we also have obstacles the explorer must move around in his search. Also, rather than just one wumpus per world in our formulation, the wumpii are less solitary, and each cave world can have multiple wumpii. Our ambition was to create an agent that uses first-order logic (FOL) to reason about his surroundings and compare his performance to a less logical agent who simply wanders about the world. The data we use to test these explorers is generated through a world generation process with a number of stochastic elements. The intent with generating worlds in this manner is to get a variety of different Wumpus worlds with varying degrees of difficulty.

Hypotheses regarding the expected performance of the explorers is presented in Section 2, while an overview of the implemented algorithms is provided as Section 3 of this work. This is followed by a description of our experimental approach and the performance measures used to gauge explorer performance in Section 4. The experimental results are presented in Section 5 of this work, and the results are discussed in greater detail in Section 6.

2. Hypotheses

The two explorers implemented in this work have a number of fundamental differences with respect to how they explore and interact with the world. As a result, we expect to find performance differences between the two explorers. With respect to the performance of each explorer, we hypothesize that because the Informed Dude can discover pits and wumpii through reasoning (as opposed to dying) and that he will be able to explore more safely, thus having fewer deaths overall than Reactive Dude. When discussing the performance of the explorers, we are taking the average over multiple runs, because the stochastic nature of the explorers (especially Reactive Dude), could unfairly bias the results in the event of a particularly good or bad choice of moves. Additionally, we hypothesize that the Informed Dude will explore fewer cells than the Reactive Dude, because the Reactive Dude will be less direct in his exploration. Due to the penalties imposed on dying and extraneous moves, we hypothesize that the average cost of Reactive Dude will be higher than the average cost of Informed Dude. Additionally, we hypothesize that in worlds with lower probabilities of wumpii, pits, and obstacles, the explorers will have fewer deaths and fewer moves, and hence lower associated costs for both explorers.

3. Algorithms

The two versions of the explorer employed in this work have a number of key differences, and these differences inform the hypotheses presented in the previous section. Information about the implementation of each of the explorers is presented in the following subsections, in addition to details regarding the resolution and unification algorithms that drive the Inference Engine. Additional details regarding the data used to test these explorers, and the corresponding world generation process, can be found in Section 4 of this work.

3.1 Reactive Dude

The Reactive Dude was implemented according to the specifications laid out in the design document. At the start of the exploration, the dude is placed at location $(0, 0)$ facing south, with access to a blank map and a number of arrows corresponding to the number of wumpii in this particular world. This begins a loop of exploration that ends when the explorer gets stuck, or reaches the goal state, which corresponds to the cell where the gold is located. At each round, the choice of what cell to move into was picked from categories prioritized as follows: 1) unexplored and safe cells (i.e. ones where the explorer does not sense a breeze or a stench through their percepts), 2) potentially unsafe but unexplored cells, often resulting in deaths by pit or wumpus, 3) killing a wumpus, and lastly 4) backtracking across previously explored cells. This order was chosen to force as much exploration as possible, as safely as possible.

If more than one choice in a category existed, a random selection was made. At the end of each round, the map and statistics available to the explorer were updated according to what was found in the cell he moved into. Additionally, if a breeze or stench was perceived, the surrounding cells were updated to reflect possible wumpus or pits. In the event where an explorer is unable to move from their starting location, the loop of exploration is terminated. For example, Figure 1 depicts one such world, where the explorer cannot move out from the

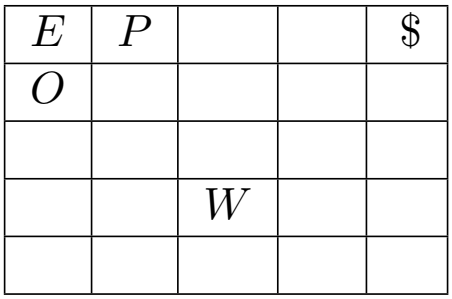

Figure 1: An example of a 5×5 Wumpus world instance in which the explorer (E) is incapable of reaching the goal state (\$) because of the pit (P) in $(0, 1)$ and the obstacle (0) in $(1, 0)$

starting location $(0, 0)$ due to a pit in $(0, 1)$ and an obstacle in (1,0). These impediments prevent the explorer from ever reaching the goal state. This example demonstrates how even a small world with a low probability of obstacles, pits and wumpii can be problematic for the explorers. There are other ways that a world can be unsolvable (i.e. the goal state surrounded by pits or obstacles), and these other types of unsolvable worlds can be discovered through manual inspection by the user.

3.2 Informed Dude

As compared to Reactive Dude our second explorer, Informed Dude, attempts to reason about the environment in order to determine the best action or sequence of actions given the current state of the world. In order to be successful, this logic-based agent must adopt and manage its attitudes about the world, reason about the world based on its knowledge, and perform actions that will bring about its desired outcomes (Bringsjord, 2008). In his work on logic-based AI, Bringsjord makes the claim the most important attitudes this agent has correspond to what the agent *believes* and what the agent *knows*. In the context of the Wumpus world, when the knowledge-based explorer believes something, they can ASK the knowledge base a question via a query α. If α is logically entailed by the knowledge base, an affirmative answer will be returned and the agent's next actions can be planned accordingly. In the event that the explorer is unable to infer anything meaningful about the environment, they use their knowledge about the safe cells and unvisited cells to make their decision in a more stochastic manner to force exploration. Similarly, when our agent knows something about the environment, they can add the assertion to the knowledge base using the $TELL$ command.

The implementation of the Informed Dude used in this work is based on the HYBRID-WUMPUS-AGENT algorithm provided by Russell and Norvig (Russell & Norvig, 2003). Although the algorithm employed in this work was modified to use first-order logic (FOL) rather than propositional logic, it still uses the knowledge base to reason about the state of the world, and applies domain-specific knowledge to decide what actions are best suited to the current environment. After an update to the knowledge base, the inference engine uses resolution and unification to discover any new rules or predicates that can be added to the knowledge base. This is a recursive learning process, where these new or modified

rules may update the knowledge base through the addition or removal of relevant facts or rules. These updates to the knowledge base help the agent to determine the wisest course of action with respect to the best move given the known state of the world.

Because FOL allows the use of quantifiers, variables, and predicates, and due to the decision to build our knowledge base of Horn Clauses, FOL is well-suited to maintain and ascertain knowledge about the Wumpus world. Although not strictly required by the project specifications, Horn clauses were selected because they are a fairly natural representation language, and there are a number of efficient inference algorithms that work well in tandem with Horn clauses (Levy & Rousset, 1998). The inference engine implemented in this work uses resolution and unification to continually update the knowledge base as the explorer gathers new knowledge from his percepts and the environment. Unification, as described by (Russell & Norvig, 2003) allows us to use multiple rules to draw conclusions about the safety (or lack-there-of) of particular cells in the world. We also use resolution, which uses a method of proof by contradiction to infer facts about the Wumpus world (Russell & Norvig, 2003). This allows us to search for proofs of logical entailment using inference, rather than enumerating different versions of the model, which would be an expensive and unwieldy alternative that would not scale well.

3.2.1 Predicates

As implemented in the program associated with this project, the Informed Dude has a knowledge base consisting of nine predicates, and these predicates comprise the facts that govern the Wumpus world (which are stored as Horn clauses). Depending on the explorer's knowledge and interactions with the environment, these facts can be either *true*, $false$ or *unknown*. The $SAFE(x, y)$ predicate signifies that the cell at location (x, y) contains no pit, obstacle or wumpus. $BREEZE(x, y)$ indicates that one or more of the neighboring cells is a pit. Similarly, $STENCH(x, y)$ signals that one or more of the neighboring cells contains a live, hungry wumpus. $BUMP(x, y)$ signifies that there is an impassable obstacle at location (x, y). Once the explorer has discovered the location of a pit, obstacle, or wumpus (either through reasoning based on percept information, a bump on the head, or through an untimely end), they can update their knowledge base with an atomic sentence like $PIT(x, y)$, $WUMPUS(x, y)$ or $OBSTACLE(x, y)$. The process of unification, described in the next subsection of the paper, provides a method to replace the general x and y locations with constants such as the explorer's x and y location at the moment when that information was acquired. Additionally, $GLIMMER(x, y)$ means there is gold in cell (x, y), and $FACING(d)$ indicates whether the adventurer is currently facing north, south, east or west. An explorer can also update their knowledge base by asserting information corresponding to the negation of any of the above. That is, $!STENCH(x, y)$ provides the explorer with the valuable information that $!WUMPUS(x+1, y)$ and $!WUMPUS(x-1, y)$ and $!WUMPUS(x, y+1)$ and $!WUMPUS(x, y+1)$.

The explorer also has a limited number of actions within the Wumpus world, each with an associated cost. The explorer can $GRABGOLD(x, y)$ only when he shares the cell with the gold, and that action costs one point, but gains the explorer 1000 points. The explorer can also $TURNL90(x, y)$ or $TurnR90(x, y)$ and $MOVEFORWARD(x, y)$, all for a one point penalty. Additionally, the explorer can $SHOOTARROW(x, y)$ for a

one point penalty, and if the wumpus is killed, the explorer gains ten points. Any move that results in the death of the explorer is an immediate 1000 point penalty. Whether they correspond to objects or relations, the constants and predicates stored in the knowledge base are essential to the processes of unification and resolution, described briefly in the following two subsections.

3.2.2 Resolution

Resolution is an inference rule that, when coupled with a complete search algorithm, is capable of producing a complete inference algorithm. In other words, as a complete inference algorithm, resolution can derive any sentence α that is entailed by the knowledge base. As previously mentioned, we specified our knowledge base in terms of Horn clauses, and the resolution process we employed utilizes clauses (that is, disjunctions of literals) along with a literal to try to derive a proof by contradiction. Our implementation of resolution was strongly rooted in the forward chaining algorithm presented by Russell and Norvig (Russell & Norvig, 2003).

In forward chaining, a list of the clauses processed so far is maintained in the knowledge base, and complementary clauses (clauses that unify with the negation of the query α) are used to eliminate clauses in order to produce new resolvent clauses. The integration of new facts requires a method for ambiguity resolution, and we handled this resolution via the expensive, but relatively intuitive process of pattern matching (Bos, 2004). Forward-chaining is advantageous in that every atomic sentence that can be derived from the given facts will be derived. However, one of the disadvantages of this approach is that many of the derived inferences are not relevant to the current query α.

3.2.3 Unification

As a reminder, in FOL, literals are complimentary if one unifies with the negation of another, making unification an integral part of first-order inference algorithms. The unification algorithm takes two sentences, p and q, and returns a substitution list (or unifier) θ that makes p and q look identical. More formally, $UNIFY(p, q) = \theta$ where $SUBST(p, \theta) = SUBST(q, \theta)$. Again, the strategy employed in this work to accomplish unification is based on the algorithm presented by Russell and Norvig (Russell & Norvig, 2003). The unification algorithm takes the inputs p and q and compares their structures element by element to build θ. At the end of the unification process, the substitution string θ is returned to the resolution algorithm where the knowledge base is updated based on the values stored in θ.

4. Methods

Prior to the implementation and evaluation of the explorers described in the previous section, we needed to generate the necessary data to test their performance according to a number of specified metrics. To do this, we implemented a Wumpus world generator, which stochastically generated pits according to an input probability P_{pit}, obstacles according to an input probability P_{obs} and wumpii according to input probability P_{wumpus}. This represents a significant departure from the Wumpus world as described by Russell and Norvig (who did not include obstacles, and had only one Wumpus per world), and makes each of

World Size	Total Cells	Probability Range to Test
5	25	(0.077, 0.120, 0.153, 0.230)
10	100	(0.030, 0.082, 0.163, 0.245)
15	225	(0.012, 0.083, 0.165, 0.248)
20	400	(0.009, 0.083, 0.166, 0.249)
25	625	(0.006, 0.083, 0.167, 0.250)

Table 1: Information on each of the Wumpus worlds tested in this work, including the size of the world, the number of cells, and P_{pit} P_{obs}, P_{wumpus} values tested for each size.

the worlds more difficult for the explorer. Variations between worlds led to problems with varying degrees of difficulty, so to reduce the impact of any very difficult (or very easy) caves, we varied the probabilities P_{pit} P_{obs}, P_{wumpus} to produce four different graphs for each of the five specified cave (world) sizes. This led to some unsolvable graphs, where the explorers could not reach the gold due to the placement of hazards in the cave environment. The cave sizes used in this work are 5×5, 10×10, 15×15, 20×20, and 25×25, which provides a total of 20 graphs for testing. To more fairly compare the efficacy of the explorers, each explorer was given ten chances on each of the caves. Additionally, to prevent a single world from flummoxing a particular explorer and adversely impacting their performance, we ran each explorer on the same set of worlds. Table 1 provides additional information on the caves used for testing, as well as the maximum and minimum probabilities used for each size of cave.

To evaluate the performance of each explorer, a number of statistics were gathered that relate to the explorer's ability to navigate through the world. These statistics included the total number of moves the explorer made on each journey, the total cost of the exploration (according to the payoff and penalty functions described in the specifications, the number of times the explorer found the gold, number of times the explorer killed a wumpus, the number of times the explorer fell into a pit, the number of times the explorer was killed by the wumpus, and the total number of cells explored. In Sections 5 and 6 of this work we explore how these metrics align with our expectations presented in Section 2.

5. Results

In the course of running our algorithms on a variety of Wumpus Worlds, we discovered that there were several possible outcomes. The first outcome is the most ideal. The Dude is able to reach the gold and is not blocked by pits or obstacles. The second outcome is that the adventurer is unable to move out of the cell at $(0, 0)$. We include this type of world in the numbers we report here. The final possible outcome is that the world is unsolvable, but the adventurer cannot detect this, so the algorithm gets stuck in an infinite loop. Unsolvable worlds occur when the explorer is constrained to a part of the map because of line (or circle) of pits and obstacles or the gold is unreachable for similar reasons. We did not report the results of this last type of outcome, as there were no results to report. This is why the table for world size 10×10 has fewer probabilities. There were only unsolvable worlds with infinite loops generated with probability 0.245 at size 10×10. Of all world sizes, we found that worlds of size 10×10 generated the highest number of unsolvable worlds.

Dude Type	World Prob	Moves	Cost	Cells Explored	% Gold Found	Total Deaths	Pit Deaths	Wumpii Deaths	Killed Wumpii
Inf	0.077	20.26	-733.20	6.16	1.0	1.70	0.68	1.02	0.48
Reac	0.077	21.14	-574.48	6.54	1.0	1.54	0.66	0.88	0.46
Inf	0.12	24.1	-2042.950	3.975	0.75	2.75	1.55	1.20	0.900
Reac	0.12	20.875	-1982.175	4.075	0.75	2.70	1.45	1.25	0.875
Inf	0.153	40.32	-4553.58	3.32	0.8	5.28	2.82	2.46	2.08
Reac	0.153	33.96	-4352.26	4.06	0.8	5.10	2.48	2.62	2.20
Inf	0.23	7.55	-1585.275	0.20	0.25	1.825	1.30	0.525	0.40
Reac	0.23	6.15	-1507.425	0.25	0.25	1.750	1.25	0.500	0.25

Table 2: Moves, Cost, Cells Explored, Gold Found, Deaths, and Killed Wumpi for both Informed Dude and Reactive Dude for several probabilities used to generate 5×5 worlds

Dude Type	World Prob	Moves	Cost	Cells Explored	% Gold Found	Total Deaths	Pit Deaths	Wumpii Deaths	Killed Wumpii
Inf	0.03	128.66	-2314.38	41.86	1.0	3.08	1.50	1.58	0.62
Reac	0.03	139.14	-2592.34	43.64	1.0	3.34	1.62	1.72	0.88
Inf	0.082	128.88	-6814.36	26.66	1.0	7.58	4.08	3.50	2.00
Reac	0.082	118.32	-6151.72	24.40	1.0	6.94	3.68	3.26	2.24
Inf	0.163	133.83	-11220.67	11.033	1.0	11.967	5.900	6.067	5.733
Reac	0.163	139.23	-13682.23	14.633	1.0	14.433	6.933	7.500	4.633

Table 3: Moves, Cost, Cells Explored, Gold Found, Deaths, and Killed Wumpi for both Informed Dude and Reactive Dude for several probabilities used to generate 10×10 worlds

We performed two-sample student's t-tests using a Confidence Level of 0.95 comparing moves, cost, cells explored, percentage of gold found, total number of deaths, number of deaths by pits, number of deaths by wumpi, and killed wumpi between the Reactive Dude and the Informed Dude. We only found statistically significant differences in the size $20X20$ worlds and only then for a few specific measures, those being cells explored, moves, deaths by pits, and death by wumpi. The relevant comparisons are bold in the Table 5. Other than tables of size $20X20$ neither algorithm performed statistically significantly better or worse than the other.

6. Discussion

Work has been done to show that (not surprisingly), as the probability P_{pit} increases, an explorer has fewer possible scenarios to reach the goal state, making exploring difficult and more deadly, as the agent has fewer locations it can determine to be safe (Sardina & Vassos, 2005). Given that our formulation of the Wumpus world differs from the one employed by Sardina and Vassos in that we also increase the probability P_{wumpus} and introduce P_{obs},

Dude Type	World Prob	Moves	Cost	Cells Explored	% Gold Found	Total Deaths	Pit Deaths	Wumpii Deaths	Killed Wumpii
Inf	0.012	187.52	-2567.40	99.30	0.96	3.24	1.58	1.66	0.68
Reac	0.012	261.28	-2476.86	103.54	0.96	3.00	1.52	1.48	0.98
Inf	0.083	242.88	-16772.62	51.30	0.94	17.34	8.72	8.62	4.44
Reac	0.083	306.62	-17354.00	57.44	0.94	17.80	9.00	8.80	6.12
Inf	0.165	245.867	-16111.7	13.3	0.633	16.3	8.033	8.600	6.100
Reac	0.165	182.767	-17127.1	18.333	0.633	17.467	8.867	8.267	5.667
Inf	0.248	2.5	-503.65	0	0	0.5	0.5	0	0
Reac	0.248	2.5	-503.5	0	0	0.5	0.5	0	0

Table 4: Moves, Cost, Cells Explored, Gold Found, Deaths, and Killed Wumpi for both Informed Dude and Reactive Dude for several probabilities used to generate 15×15 worlds

Dude Type	World Prob	Moves	Cost	Cells Explored	% Gold Found	Total Deaths	Pit Deaths	Wumpii Deaths	Killed Wumpii
Inf	0.009	396.22	-3736.5	**169.04**	0.98	4.02	2.0	**2.02**	**0.6**
Reac	0.009	457.98	-4643.5	**211.56**	0.98	4.82	2.3	**2.52**	**1.4**
Inf	0.083	319.1	-21110.46	62.76	0.94	21.54	10.70	10.84	5.42
Reac	0.083	365.5	-21973.64	68.94	0.94	22.32	10.78	11.54	7.76
Inf	0.166	**992.375**	-50121.12	41.100	0.8	49.125	26.375	22.750	20.775
Reac	0.166	**589.700**	-44466.55	43.675	0.8	44.350	22.625	21.725	18.800
Inf	0.249	**15.4**	-3217.2	2	1	4.2	**1.6**	2.6	2
Reac	0.249	**18.0**	-4018.6	2	1	5.0	**2.0**	3.0	2

Table 5: Moves, Cost, Cells Explored, Gold Found, Deaths, and Killed Wumpi for both Informed Dude and Reactive Dude for several probabilities used to generate 20×20 worlds

Dude Type	World Prob	Moves	Cost	Cells Explored	% Gold Found	Total Deaths	Pit Deaths	Wumpii Deaths	Killed Wumpii
Inf	0.006	537.20	-3448.12	265.60	0.88	3.48	1.76	1.72	0.78
Reac	0.006	574.82	-4304.36	299.34	0.88	4.28	2.18	2.10	1.18
Inf	0.083	474.42	-28860.56	84.64	0.8	34.9	14.46	14.54	9.02
Reac	0.083	572.76	-34925.26	104.56	0.8	29.0	17.48	17.42	13.12
Inf	0.167	1021.40	-65046.15	56.15	1.2	64.05	31.65	32.40	27.15
Reac	0.167	1400.85	-65484.85	60.05	1.2	63.70	31.75	31.95	28.85
Inf	0.250	3	-1003.55	0	0	1	1	0	0
Reac	0.250	3	-1003.50	0	0	1	1	0	0

Table 6: Moves, Cost, Cells Explored, Gold Found, Deaths, and Killed Wumpi for both Informed Dude and Reactive Dude for several probabilities used to generate 25×25 worlds

the number of locations the explorer can determine to be safe is likely made even smaller through the introduction of these additional (often lethal) impediments.

As discussed in results, Reactive Dude and Informed Dude performed the the same statistically. However, of interest to us, we noticed a large discrepancy of performance in measure of time between Informed Dude and Reactive Dude, especially at high probabilities and large worlds. For example, on running through all probabilities of size 20 worlds, the Informed Dude took over 2 hours, a feat Reactive Dude completed in under 5 minutes. Additionally, Informed Dude entered an infinite loop on our third test world of size 25, a world confirmed to be solvable, indicating an error in Inference Engine. With respect to our Informed Dude we found that despite testing and modifying our agent, debugging our knowledge base, and testing our inference engine, the agent did not perform as well as expected on a number of key measures. This gulf between our expectations and the explorer's performance could be due to a number of implementation issues, including tactical errors in resolution or unification, an insufficient number of rules to accurately describe the domain, or an ordering of the rules that made inference difficult. In his work on perpetual self-aware cognitive agents, Michael Cox presents a number of common issues related to expectation failure, which occurs when a system expects one explanation to be the case, when in reality another explanation proved true instead (Cox, 2007). These issues include contradictions between the expected and actual explanation, novel situations and erroneous associations - any (or all) of which could be the culprit when our Informed Dude appeared to be performing unexpected or foolhardy actions.

References

Ahl, D. H., & North, S. (1979). *More Basic Computer Games*. Creative Computing Press, Morristown, NJ, USA.

Bos, J. (2004). Computational semantics in discourse: Underspecification, resolution, and inference. *Journal of Logic, Language and Information*, *13*(2), 139–157.

Bringsjord, S. (2008). The logicist manifesto: At long last let logic-based artificial intelligence become a field unto itself. *Journal of Applied Logic*, *6*(4), 502 – 525. The Philosophy of Computer Science.

Cox, M. T. (2007). Perpetual self-aware cognitive agents. *AI magazine*, *28*(1), 32.

Levy, A. Y., & Rousset, M.-C. (1998). Combining horn rules and description logics in CARIN. *Artificial Intelligence*, *104*(1), 165 – 209.

Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (3rd edition). Pearson Education.

Sardina, S., & Vassos, S. (2005). The wumpus world in indigolog: A preliminary report. In *Proceedings of the Workshop on Non-monotonic Reasoning, Action and Change at IJCAI (NRAC-05)*, pp. 90–95.