Finn O’Leary

Intro to AI 449

Professor Scott Weiss

5/8/2024

# Final Project Writeup

The program that I decided to create for my final project in Intro to AI was a propositional logic program that was made to beat the game Wumpus World! My Agent utilizes propositional logic to move throughout Wumpus world to find the gold and get back home safely. Once a move is made, the agent updates its knowledge base based on what it perceives in the given cell. If there is a breeze, it knows there is a possible pit in all adjacent spaces that have not yet been deemed safe from a pit, and if there is no breeze in the current cell it marks all adjacent cells safe from having a pit. It does the same thing for the Wumpus and the stenches, if we are in a cell with a stench, we mark a possible Wumpus in the adjacent cells, not yet deemed safe of Wumpus. After updating the KB, we then resolve the knowledge base to break it down with the new clauses that were added and see what information we gained from the new clauses. If I gain definite knowledge of the Wumpus location, I will shoot the Wumpus, assuming we are in an adjacent cell. The agent keeps moving through all of the guaranteed safe moves and if it's in a spot that has no new safe move, we go back a move. Once we are back a move we evaluate to see if there are any nonvisited safe moves in adjacent squares, to do this we check if our knowledge base doesn't have a visited marker for any of the adjacent cells and if the given cell has no Wumpus and no Pit. If we keep going back, due to there being no safe moves and we eventually reach home, this means there is no guaranteed safe move, so we need to find one. At this point, I evaluate the knowledge base using my evaluate function and pick what is deemed

the safest possible move and move to that spot. This function checks if we know the location of a pit and if we do, we traverse to an adjacent non-visited square and continue. This cycle keeps going until we perceive glitter meaning there is Gold in our current cell, and in this case, we follow all of our steps backward and go home. Once we are home we display the path that our agent took to navigate through the world.

To use the system that I created is very simple. The seed is set to randomly generate a world when you give it no parameters, but I do have some seeds pre-built into it that I can access utilizing a parameter, so I could test the intelligence of my agent. After the world is randomly generated, the user is then prompted to answer the question if they are a computer or a person. If they decide that they are a person, they place the game normally, but if they choose computer then it allows my agent to navigate through Wumpus World and hopefully come out alive and with the gold! If you choose a person, it asks for the person's move or if they want to shoot the wumpus or pick up gold. Once they are dead or home safe with the gold, it displays the moves that were taken during the game and if they won or lost.

My propositional logic algorithm utilizes a HashSet to store all of the clauses that are in my Knowledge Base. I chose a HashSet because I can easily use the contains, remove, and get method to go through the Knowledge to resolve it. It makes my resolutions more time-efficient and faster. My propositional logic works utilizing multiple functions, the update knowledge base, the resolve function for two clauses, and the resolution for the entire Knowledge Base. My updated knowledge base will mark the current cell as visited if it is not visited and will verify that it says that there is no Wumpus or Pit in the current cell. After this, it creates two different clauses to add to the Knowledge Base, one for the Wumpus and one for the pits. It creates these by seeing if it perceived a stench and if it did, we will create a statement utilizing OR, that says

there can be a Wumpus in any adjacent cells that we have not deemed safe from Wumpus. It does the same thing for the pits and the breezes as well. If the current cell has no stench it marks all adjacent cells as not having a Wumpus and if the current cell has no breeze then it marks all the adjacent cells as safe from having a pit. My resolution function takes in two clauses and the kind of clause it is evaluating, a “W” or a “P,” as its parameters. It then breaks the two clauses down into one resolved function if possible and returns it. We then remove the unnecessary clauses and add the resolved ones to the KB. The function that resolves the whole knowledge base, uses two loops that go through the entire KB comparing all the clauses to each other if they are of the same type. After the loops are complete it checks if our Knowledge Base changed and if it didn't, we end the function, if it did we try to resolve it more. We wait until nothing has been changed, because that means we can resolve the Knowledge Base any further.

The Algorithm that I created beats every single game that you throw at it with the exception of a couple. When I randomly generate a board, sometimes it is an unwinnable board, so it does not beat it. It tends to struggle the most when two pits are next to each other or diagonal to each other. This is because it makes our knowledge base misleading. After all, when I declare there is a P11 OR P20 it doesn't take into account that there could be a pit in both when resolving. It will then go back to home when there is no possible safe move and it may deem one of them as safe, not realizing there are two pits adjacent or diagonal to each other. Another thing my agent does is to get home it would take the moves previously taken which is not always the most efficient root. I learned a lot throughout the project. I used many new functions that I didn't know were a thing, examples being the Collections.sort method and I also learned the String.startWith() function that checks if a string starts with something. I also learned that whenever I am going to change vital functions, I need to copy my current function and save it

because there is a chance I will mess up my function too much and forget how to revert it, so copying it and saving it is the safest bet! Overall I loved the project and had a ton of fun thinking it through and working on the program to beat Wumpus World!