Escape Wumpus World

April 24, 2023

Abstract

This report investigates the development of a plan for an agent to escape from the Wumpus world. The Wumpus world contains various obstacles and corresponding states to overcome them. Through exploration of these obstacles and their solutions, a plan will be formulated for the agent to escape the Wumpus world. Pre-existing Wumpus world examples have been used in this problem. With proper planning, a feasible escape route can be found for every Wumpus world scenario.

1 Introduction

For the agent to exit a given map, it is sufficient for them to move and reach the designated coordinates. However, in this problem, there are certain obstacles that impede the agent's progress while navigating through the map. There are actions that can be taken to eliminate some of these obstacles. This report aims to discover a plan that will enable the agent to escape from the Wumpus world by overcoming these obstacles with the correct actions.

Research question How can an agent escape from the Wumpus world by overcoming its obstacles?

Contribution There are two key contributions in this report:

1. The development of a viable plan for escaping the map.
2. The identification of appropriate actions to overcome the obstacles in the map.

Overview In section 2, the problem is discussed in detail. Section 3 covers the constraints and rules of the problem. Section 4 lists the actions that can be taken to escape the map. Section 5 provides an example simulation that represents the constraints and actions. In section 6, algorithms that can be used for the solution are discussed. The report concludes with section 7.

2 Problem description

In this problem, we have an agent represented by the starting position 'S' located within a 12x8 grid map of a Wumpus world. The agent's goal is to escape from this Wumpus world, but there are obstacles that the agent must overcome. One type of obstacle is represented by walls denoted by 'X' which the agent cannot pass through. The other type of obstacles are cells containing objects that the agent can only enter by applying the actions specified in Section 4 and following the rules listed in Section 3. If the rules are not followed, the agent cannot move into those cells. These obstacles include a wumpus denoted by 'W', a door denoted by 'D', a pit denoted by 'P', a crate denoted by 'C', and a half crate denoted by 'H'. There are also cells containing objects that do not hinder movement, such as an arrow denoted by 'A' and a key denoted by 'K'. The agent can also move into empty cells. All of these objects can be present in a single map or in some maps, fewer objects may be present. Figure 1 shows an example map with all these objects. The goal of this problem is to escape from the map. Before diving into how the obstacles can be overcome, we will first explain the movements' nomenclature to better understand the escape plan in Figure 1.

Our agent has four possible movements in the grid map: 'north', 'south', 'east', and 'west'. The agent's current position is represented by 'S' and is located at cell $\langle 5, 2\rangle$. The possible movements from this cell are:

- Move 'north' to cell $\langle 5, 1\rangle$.
- Move 'south' to cell $\langle 5, 3\rangle$.

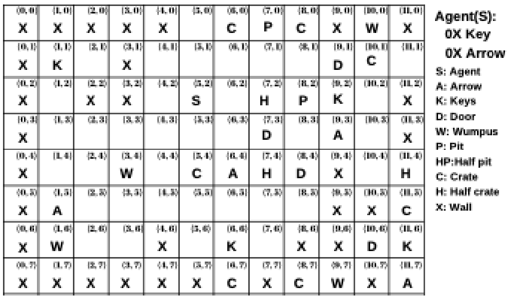

Figure 1: Sample Wumpus world map

- Move 'east' to cell $\langle 6, 2\rangle$.
- Move 'west' to cell $\langle 4, 2\rangle$.

A map representing these coordinates is shown in figure 2. These movements are denoted based on the cardinal directions: 'north' for moving up, 'south' for moving down, 'east' for moving right, and 'west' for moving left.

In Figure 1, our agent denoted as 'S' starts at the position $\langle 5, 2\rangle$ and when it moves in the north direction, and then in the north direction again, it reaches the coordinate $\langle 5, 0\rangle$ which is at the edge of the map. The direction of the last movement can be decided based on rule-based decisions. For example, one can exit the map by moving north at the $\langle 5, 0\rangle$ cell. However, there is a more practical way to exit the map for this problem. One can exit the map by using the same direction as the last movement towards the edges of the map. In the current example, the last movement was in the north direction to reach the point $\langle 5, 0\rangle$. Similarly, moving north can be used to exit the map and achieve the goal.

3 Constraints and rules

The objects mentioned in the problem definition need to be further introduced in order to better understand the constraints and rules. In this section, the rules and constraints that cannot be directly expressed in this class will be explained through the objects.

3.1 Object Based

The The object 'X' is represented as a wall, and agent cannot move to the cell where this object is located. It is strictly impossible to change or convert this object into a movable cell with any other action or rule.

The object 'A' is represented as an arrow. There is no constraint to move to the cell where this object is located. As a rule, if our agent moves to this cell, it automatically picks up the arrow. After that, the object is no longer in the cell, and our agent owns the arrow.

The object 'K' is represented as a key, and like the arrow, there is no constraint to move to the cell where this object is located. Similarly, if our agent passes through this cell, it gains ownership of the key, and the object is no longer in the cell.

The object 'D' is represented as a door. Our agent cannot enter the cell where the door object is located. To move to this cell, our agent needs a key. Before the movement action, the key must be used to open the door and make the cell movable. After opening the door, our agent no longer owns the key used to open the door, and movement to the open door is possible.

The object 'C' is represented as a crate, and direct movement to the cell where this object is located is not possible. However, it is possible to reach the cell where the object is by pushing it. The rules for pushing this object are shared in the general rules section.

The object 'H' is represented as a half crate, and it has similar properties and constraints to the crate. When pushed onto a pit, it separates into two half pits, and the details of this separation are explained in the pit section. The most important distinction that can be mentioned in this section is that two half crates can be pushed simultaneously.

The object 'P' is represented as a pit, and our agent cannot directly enter the cell where this object is located. To enter this cell, the agent needs to use a crate or half crate. If a crate is pushed into a pit, both the crate and the pit disappear, and the cell becomes available for the agent to move. If a half crate is pushed into a pit, the half crate disappears, and the pit becomes a half pit. Like a pit, a half pit is not suitable for the agent to move, and a half crate is required for the agent to move to a half pit. If a half crate is pushed into a half pit, both the half crate and the half pit disappear, and the cell becomes available for the agent to move.

The object 'W' is represented as a wumpus. Our agent cannot enter the cell where the wumpus object is located. To move to this cell, our agent needs a arrow. Before the movement action, the arrow must be used to shoot the wumpus and make the cell movable. After shooting the wumpus, our agent no longer owns the arrow used to shoot the wumpus.

3.2 General Rules

In order for the agent to perform its actions on the map, some rules need to be defined. With these rules, the agent can determine its movement areas and reach the correct outcome. Place: Firstly, cells on the map must be defined. These cells should be our constraints for movement and there should be no movement area defined outside these cells.

Connected: Firstly, in an empty map, the cells from which our agent can move should be determined. If this rule is not defined, our agent can move randomly. However, in our problem, our agent has a movement capacity only in the north, south, east, and west directions. For example, in Figure 2, our agent is located at point ⟨5, 2⟩. The cells that will be connected at this point are ⟨5, 1⟩, ⟨5, 3⟩, ⟨6, 2⟩, and ⟨4, 2⟩. However, reaching the cell ⟨6, 1⟩ requires a diagonal movement, which is not possible for our agent. Therefore, this cell should not be connected. Every connection of each cell should be specified based on this rule.

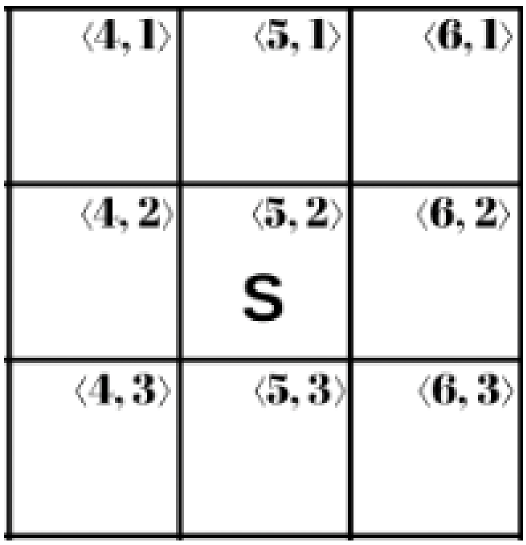

Figure 2: Shows the map. Movements: north, south, east, west.

Pushable: In order to be pushed, crates and half crates must meet certain conditions. Some of these conditions are expressed when describing the movements in section 4. However, the most important rule that needs to be expressed in this section is that the cell where the agent is located, the cell where the object is located, and the cell where the object will be pushed must be in the same direction. At the same time, the order should be agent, object, goal cell to push or goal cell to push, object, agent. An example of an appropriate position is given in Figure 3. Figure 4 cannot be pushed because it is not in the same direction, and Figure 5 cannot be pushed because the order is incorrect.

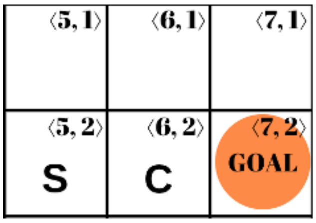

Figure 3: Proper Pushing Example

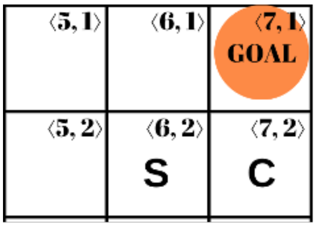

Figure 4: Improper Direction Example

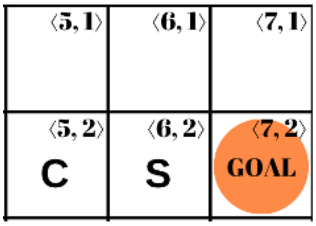

Figure 5: Improper Order Example

Double pushable: It has similar features to pushable but is expected to comply with this rule when two half Crates are to be pushed. As with pushable, there should be a direction between the cells. However, since there are two half Crates in this, the order should be as follows: agent, half Crate, half Crate, goal cell to push or goal cell to push, half Crate, half Crate, agent. An appropriate situation is shown in Figure 6.

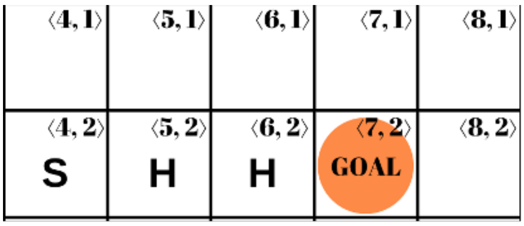

Figure 6: Double Pushable Half Crate Rule Example

4 Actions

In order to move our agent between cells and overcome obstacles in the cells to reach the goal, we need actions that include rules and constraints. With these actions, we can create our plans and implement our movement steps.

Move: With this action, a simple movement from one cell to another is targeted. In the example in Figure 7, the position of the robot is $\langle 9,\ 1\rangle$ and there is an arrow at $\langle 10,\ 1\rangle$. The connected condition between $\langle 9,\ 1\rangle$ and $\langle 10,\ 1\rangle$ is satisfied and it can be moved. At the same time, there is no object that is not suitable for movement based on the conditions described in sections 2 and 3 in the cell where it will move, which is ⟨10, 1⟩. There is an arrow, but as explained, the arrow is not an obstacle for movement. The completed state of the movement is shown in Figure 8. When the movement is completed, the agent is not in ⟨9, 1⟩ but in the cell ⟨10, 1⟩. The cell to be moved can be completely empty or contain a key and an arrow. If one of the key or arrow objects is in the moved cell, this object is automatically owned by the agent. In our example, the arrow in cell ⟨10, 1⟩ is no longer there when the movement is completed, and the number of arrows owned by the agent has increased by one.

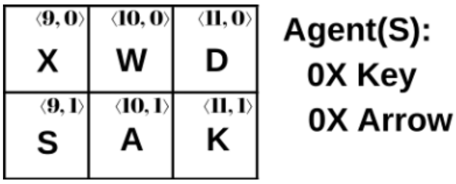

Figure 7: Simple movement from ⟨9, 1⟩ to ⟨10, 1⟩ with arrow pick-up.

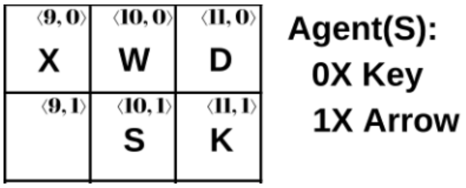

Figure 8: Completed state of the movement.

Shoot: With this action, the aim is to hit the wumpus in one of the cells that has the connected cell property for the cell where the agent is located, and make the cell suitable for movement. To complete this action, the agent also needs at least one arrow. In Figure 9, as seen, the cell ⟨10, 1⟩ where the agent is located and the cell ⟨10, 0⟩ where the wumpus is located satisfy the connected rule. At the same time, the agent has one arrow. The completed state of the action is shown in Figure 10. When the action is completed, the agent will no longer have the arrow used to hit the wumpus, so the number of arrows owned by the agent has decreased by one. The wumpus will no longer be in cell ⟨10, 0⟩, it has been destroyed. The cell ⟨10, 0⟩ will now be available for movement.

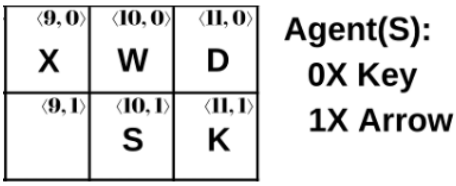

Figure 9: Hitting the Wumpus with Arrow.

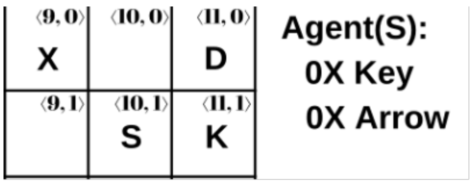

Figure 10: Completed State of the Shoot Action.

Unlock: With this action, the aim is to unlock the door and make it suitable for movement in one of the cells that has the connected cell property for the cell where the agent is located, provided that there is a door. To be able to apply this action, the agent must have at least one key. In Figure 11, our agent is located in cell $\langle 11, 1\rangle$ and the cell $\langle 11, 0\rangle$ where the door is located satisfies the connected rule, and at the same time, our agent has one key. The state after the action is completed is shown in Figure 12. When the action is completed, the agent will no longer have the key used to unlock the door, so the number of keys owned by the agent will decrease by one. The cell $\langle 11, 0\rangle$ where the door is located will no longer have the door, as the door will be destroyed as soon as it is unlocked. There is no possibility of the door closing again. The cell $\langle 11, 0\rangle$ will now be suitable for the agent to move.

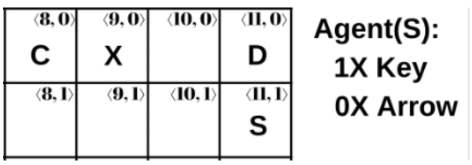

Figure 11: Agent unlocking the door.

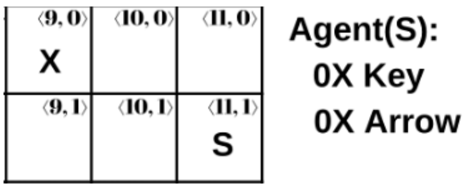

Figure 12: Resulting state after unlocking the door.

Push: There can be three different initial states for this action.

The first one could be pushing a Crate in one of the cells that satisfies the connected condition with the cell where the agent is located. In this case, the cell where the agent is located, the cell where the Crate is located, and the cell where the Crate will be pushed to must satisfy the pushable rule. In Figure 13, the cell where the agent is located is indicated as $\langle 5, 1\rangle$, the cell where the Crate is located is $\langle 6, 1\rangle$, and the cell where the Crate will be pushed to is $\langle 7, 1\rangle$. In our example, these three cells satisfy the pushable rule. This action has two possible outcomes depending on whether there is a pit in the cell where the Crate will be moved. In our example, there is no Pit in the cell $\langle 7, 1\rangle$. After this action is completed, the Crate will now be in the cell $\langle 7, 1\rangle$ and will no longer be in the cell $\langle 6, 1\rangle$. The cell $\langle 6, 1\rangle$ will now be a valid cell for the agent to move to. The agent will now be in the cell $\langle 6, 1\rangle$ and will no longer be in the cell $\langle 5, 1\rangle$. The initial cell $\langle 5, 1\rangle$ where the agent was located will now be empty. The situation that occurs when the action is completed is shown in Figure 14.

In the case where there is a Pit, the Crate is pushed to the cell with the Pit. As an example in Figure 15, the cell where the agent is located is indicated

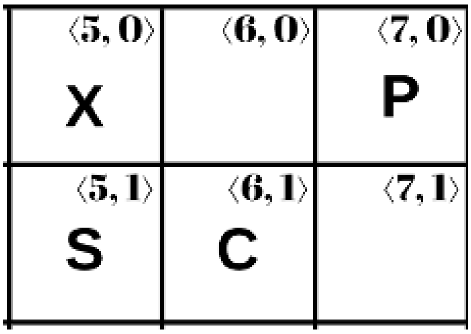

Figure 13: Pushing a Crate to an adjacent cell.

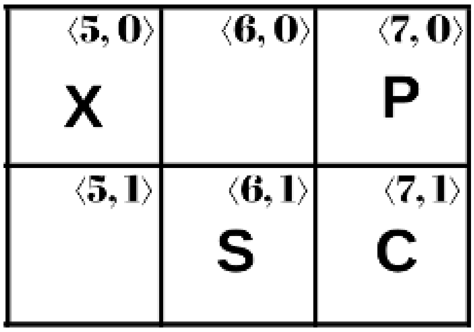

Figure 14: Result of pushing a Crate to an adjacent cell.

as $\langle 5,\ 1\rangle$, the cell where the Crate is located is $\langle 6,\ 1\rangle$, and the target cell to be pushed is $\langle 7,\ 1\rangle$, similar to the previous example, but there is a Pit in the cell $\langle 7,\ 1\rangle$. When the action is completed, the Crate is pushed to the cell $\langle 7,\ 1\rangle$. In this case, both the Crate and the Pit are destroyed. Both the cell $\langle 6,\ 1\rangle$ where the Crate was initially located and the cell $\langle 7,\ 1\rangle$ are now empty and can be moved into. The agent is initially located in the cell $\langle 6,\ 1\rangle$ where the Crate is and is no longer located in the cell $\langle 5,\ 1\rangle$. The situation that occurs when the action is completed is shown in Figure 16.

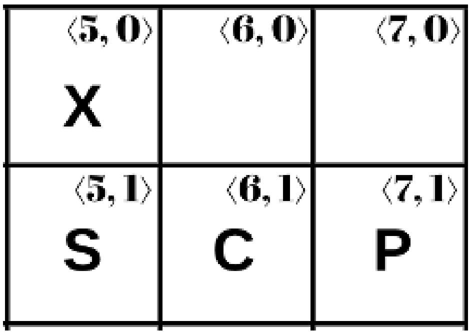

Figure 15: Pushing a Crate onto a Pit.

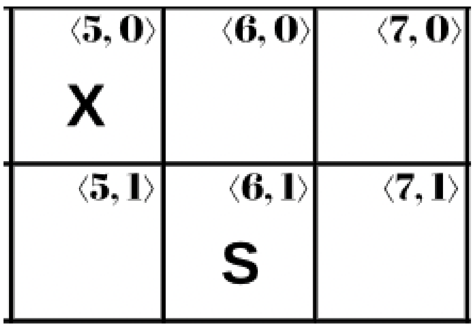

Figure 16: Result of Pushing a Crate onto a Pit.

The second scenario involves pushing a half Crate in one of the cells that satisfies the connected condition with the cell where the agent is located. In order to perform this action, the cell where the agent is located, the cell where the half Crate is located, and the cell where it will be pushed must satisfy the pushable rule. After this rule is met, the movement can be executed. Depending on whether there is a pit in the cell where the half Crate will be moved, this action may result in three outcomes.

In the first case, there is no pit. As an example in Figure 17, the cell where the agent is located is ⟨5, 1⟩, the cell where the half Crate is located is ⟨6, 1⟩, and the cell where it will be pushed is ⟨7, 1⟩. These three cells satisfy the pushable rule. After this action is completed, the half Crate is now located in cell ⟨7, 1⟩ and is no longer in cell ⟨6, 1⟩. Cell ⟨6, 1⟩ is now a movable cell for the agent. The agent is now located in cell ⟨6, 1⟩ and is no longer in cell ⟨5, 1⟩. The cell ⟨5, 1⟩, where the agent was initially located, is now empty. The situation that occurs when the action is completed is shown in Figure 18.

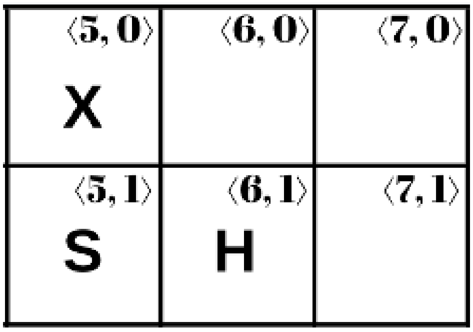

Figure 17: Pushing a half Crate.

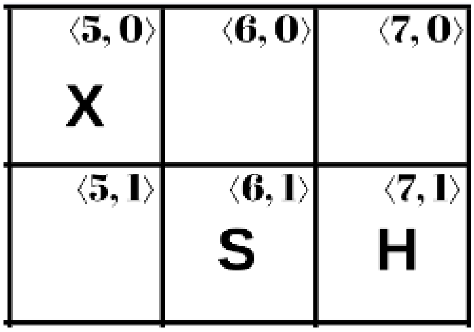

Figure 18: Resulting situation after pushing a half Crate.

If there is a pit in the cell, then the half Crate is pushed into the pit cell. However, unlike the previous scenario, the pit is not eliminated. In this case, the half Crate is destroyed and the cell it was in becomes movable. However, the pit still remains impassable and becomes a half pit. To eliminate the half pit and make the cell movable, another half Crate is needed. The example for the first scenario is shown in Figure 19 before and after the action is completed. Finally, there is the case where there is a half pit. In this case, a similar outcome occurs as when the Crate is pushed into the pit. The half pit and half Crate are destroyed, the cells they were in become movable, and the agent moves to the cell where the half Crate was. The situation that occurs when the action is completed is shown in Figure 20.

The final scenario is when two half Crates are pushed simultaneously. For this to happen, the connected condition must be established between the agent and the half Crate, and the agent must satisfy the double pushable rule for the two half Crates and the target cell. This scenario can also be divided into three cases based on whether the target cell is empty, contains a pit, or contains a half pit.

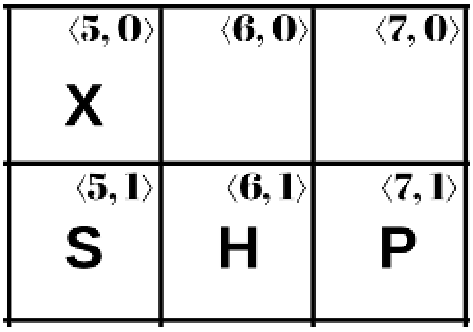

Figure 19: Pushing the Crate into the pit.

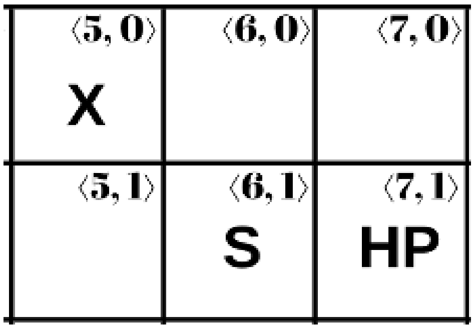

Figure 20: After pushing the half Crate into the half pit.

If the target cell is empty, the half Crate furthest from the agent moves to the empty cell, leaving a half Crate in the previously empty cell, which is now immovable. An example is shown in Figure 21, where the agent is in cell ⟨5, 1⟩, and there are half Crates in cells ⟨6, 1⟩ and ⟨7, 1⟩. The target cell is ⟨8, 1⟩. All four cells, ⟨5, 1⟩, ⟨6, 1⟩, ⟨7, 1⟩, and ⟨8, 1⟩, satisfy the double pushable rule. After the action is completed, the half Crate in cell ⟨7, 1⟩ will be pushed to cell ⟨8, 1⟩, which will become immovable. The half Crate in cell ⟨6, 1⟩ will be pushed to cell ⟨7, 1⟩, leaving cell ⟨6, 1⟩ empty and movable. The agent will move to cell ⟨6, 1⟩ and will no longer be in cell ⟨5, 1⟩. The situation that occurs when the action is completed is shown in Figure 22.

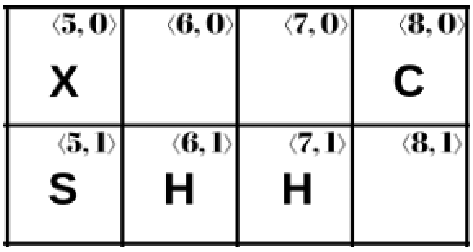

Figure 21: Example of double pushable for half Crates..

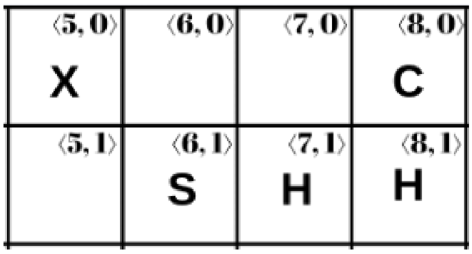

Figure 22: Result of the double pushable for half Crates..

In the case where there is a Pit, the half Crate farthest from the agent is pushed into the cell containing the Pit. In the new situation, the pushed half Crate disappears, and the Pit is updated as a half Pit. An example is given in Figure 23. The agent is located in cell ⟨5, 1⟩, and there are half Crates in cells ⟨6, 1⟩ and ⟨7, 1⟩. Our target cell is ⟨8, 1⟩, which contains a Pit. Cells ⟨5, 1⟩, ⟨6, 1⟩, ⟨7, 1⟩, and ⟨8, 1⟩ comply with the double pushable rule. Upon completion of the action, the half Crate in cell ⟨7, 1⟩ will be pushed to cell ⟨8, 1⟩, and the Pit in cell ⟨8, 1⟩ will be updated to a half Pit. The half Crate in cell ⟨6, 1⟩ will be pushed to cell ⟨7, 1⟩. Cell ⟨6, 1⟩ will be empty and movable. The agent will move to cell ⟨6, 1⟩ and will no longer be in cell ⟨5, 1⟩. The situation that occurs when the action is completed is shown in Figure 24.

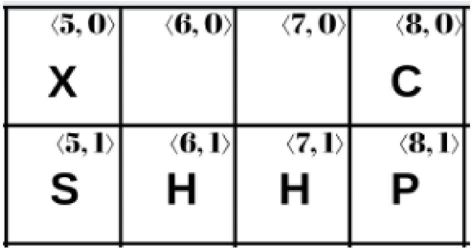

Figure 23: Half Crate Pushed into Pit.

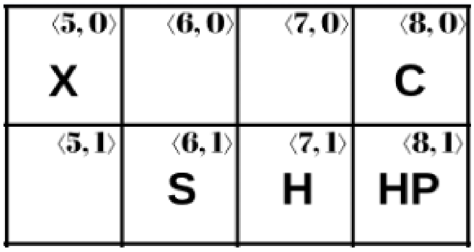

Figure 24: Result of Pushing Half Crate into Pit.

In the case of Half Pit, the half Crate farthest from the agent is pushed to the cell where the Half Pit is located. In this situation, both the Half Pit and the half Crate in the cell where the Half Pit is located will disappear. An example is given in Figure 25, where our agent is located in cell ⟨5, 1⟩, and there are half Crates in cells ⟨6, 1⟩ and ⟨7, 1⟩. Our target cell is ⟨8, 1⟩, where there is a Half Pit. Cells ⟨5, 1⟩, ⟨6, 1⟩, ⟨7, 1⟩, and ⟨8, 1⟩ all comply with the doublepushable rule. After the action is completed, the half Crate in cell ⟨7, 1⟩ will be pushed to cell ⟨8, 1⟩, causing the Half Pit and the half Crate in cell ⟨8, 1⟩ to disappear. Cell ⟨8, 1⟩ will be empty and movable. The half Crate in cell ⟨6, 1⟩ will be pushed to cell ⟨7, 1⟩, and cell ⟨6, 1⟩ will be empty and movable. Our agent will move to cell ⟨6, 1⟩ and will no longer be in cell ⟨5, 1⟩. The situation that occurs when the action is completed is shown in Figure 26.

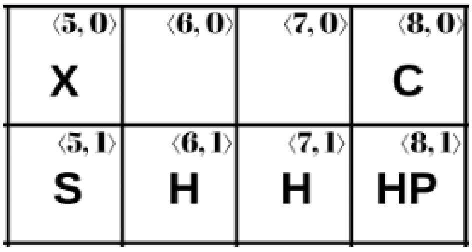

Figure 25: Half Pit and Half Crate pushing Example.

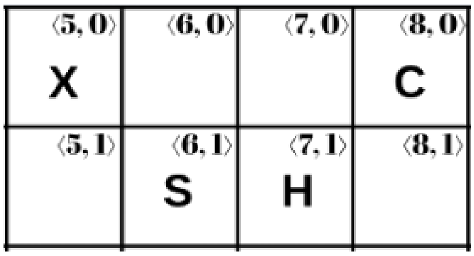

Figure 26: Resulting Situation after Action Completion Half Pit and Half Crate pushing.

5 Sample Wumpus Map Escape Plan

A representational example that utilizes actions and constraints has been presented. In this example, all 4 actions have been attempted to be used while representing as many constraints as possible, although not all of them could be represented.

The starting position is shown in Figure 27, and the goal is to reach a cell located on the edge of the map and exit from there.

The agent has moved eastward in Figure 28 using a push action, and has reached cell ⟨6, 2⟩. During this movement, 2 half crates were pushed, satisfying the double pushable condition. The half crates were pushed to cell ⟨8, 2⟩ where there was a pit, causing both the half crate and the pit to disappear and become a half pit. The half crate in cell ⟨6, 2⟩ was pushed to cell ⟨7, 2⟩.

⟨0,0⟩ X	⟨1,0⟩ X	⟨2,0⟩ X	⟨3,0⟩ X	⟨4,0⟩ X	⟨5,0⟩ X	⟨6,0⟩ C	⟨7,0⟩ P	⟨8,0⟩ C	⟨9,0⟩ X	⟨10,0⟩ W	⟨11,0⟩ X
⟨0,1⟩ X	⟨1,1⟩ K	⟨2,1⟩	⟨3,1⟩ X	⟨4,1⟩	⟨5,1⟩ X	⟨6,1⟩	⟨7,1⟩	⟨8,1⟩	⟨9,1⟩ D	⟨10,1⟩ C	⟨11,1⟩
⟨0,2⟩ X	⟨1,2⟩	⟨2,2⟩ X	⟨3,2⟩ X	⟨4,2⟩ C	⟨5,2⟩ S	⟨6,2⟩ H	⟨7,2⟩ H	⟨8,2⟩ P	⟨9,2⟩ K	⟨10,2⟩ A	⟨11,2⟩ X
⟨0,3⟩ X	⟨1,3⟩	⟨2,3⟩	⟨3,3⟩	⟨4,3⟩ X	⟨5,3⟩ P	⟨6,3⟩	⟨7,3⟩ D	⟨8,3⟩	⟨9,3⟩	⟨10,3⟩	⟨11,3⟩ X
⟨0,4⟩ X	⟨1,4⟩	⟨2,4⟩	⟨3,4⟩ W	⟨4,4⟩	⟨5,4⟩ C	⟨6,4⟩ A	⟨7,4⟩ H	⟨8,4⟩ D	⟨9,4⟩ X	⟨10,4⟩	⟨11,4⟩ H
⟨0,5⟩ X	⟨1,5⟩ A	⟨2,5⟩	⟨3,5⟩	⟨4,5⟩	⟨5,5⟩	⟨6,5⟩	⟨7,5⟩	⟨8,5⟩	⟨9,5⟩ X	⟨10,5⟩ X	⟨11,5⟩ C
⟨0,6⟩ X	⟨1,6⟩ W	⟨2,6⟩	⟨3,6⟩	⟨4,6⟩ X	⟨5,6⟩	⟨6,6⟩ K	⟨7,6⟩	⟨8,6⟩ X	⟨9,6⟩ X	⟨10,6⟩ D	⟨11,6⟩ K
⟨0,7⟩ X	⟨1,7⟩ X	⟨2,7⟩ X	⟨3,7⟩ X	⟨4,7⟩ X	⟨5,7⟩ X	⟨6,7⟩ C	⟨7,7⟩ X	⟨8,7⟩ C	⟨9,7⟩ W	⟨10,7⟩ X	⟨11,7⟩ A

Agent(S):
0X Key
0X Arrow
S: Agent
A: Arrow
K: Keys
D: Door
W: Wumpus
P: Pit
HP:Half pit
C: Crate
H: Half crate
X: Wall

Figure 27: Starting Position.

⟨0,0⟩ X	⟨1,0⟩ X	⟨2,0⟩ X	⟨3,0⟩ X	⟨4,0⟩ X	⟨5,0⟩ X	⟨6,0⟩ C	⟨7,0⟩ P	⟨8,0⟩ C	⟨9,0⟩ X	⟨10,0⟩ W	⟨11,0⟩ X
⟨0,1⟩ X	⟨1,1⟩ K	⟨2,1⟩	⟨3,1⟩ X	⟨4,1⟩	⟨5,1⟩ X	⟨6,1⟩	⟨7,1⟩	⟨8,1⟩	⟨9,1⟩ D	⟨10,1⟩ C	⟨11,1⟩
⟨0,2⟩ X	⟨1,2⟩	⟨2,2⟩ X	⟨3,2⟩ X	⟨4,2⟩ C	⟨5,2⟩	⟨6,2⟩ S	⟨7,2⟩ H	⟨8,2⟩ HP	⟨9,2⟩ K	⟨10,2⟩ A	⟨11,2⟩ X
⟨0,3⟩ X	⟨1,3⟩	⟨2,3⟩	⟨3,3⟩	⟨4,3⟩ X	⟨5,3⟩ P	⟨6,3⟩	⟨7,3⟩ D	⟨8,3⟩	⟨9,3⟩	⟨10,3⟩	⟨11,3⟩ X
⟨0,4⟩ X	⟨1,4⟩	⟨2,4⟩	⟨3,4⟩ W	⟨4,4⟩	⟨5,4⟩ C	⟨6,4⟩ A	⟨7,4⟩ H	⟨8,4⟩ D	⟨9,4⟩ X	⟨10,4⟩	⟨11,4⟩ H
⟨0,5⟩ X	⟨1,5⟩ A	⟨2,5⟩	⟨3,5⟩	⟨4,5⟩	⟨5,5⟩	⟨6,5⟩	⟨7,5⟩	⟨8,5⟩	⟨9,5⟩ X	⟨10,5⟩ X	⟨11,5⟩ C
⟨0,6⟩ X	⟨1,6⟩ W	⟨2,6⟩	⟨3,6⟩	⟨4,6⟩ X	⟨5,6⟩	⟨6,6⟩ K	⟨7,6⟩	⟨8,6⟩ X	⟨9,6⟩ X	⟨10,6⟩ D	⟨11,6⟩ K
⟨0,7⟩ X	⟨1,7⟩ X	⟨2,7⟩ X	⟨3,7⟩ X	⟨4,7⟩ X	⟨5,7⟩ X	⟨6,7⟩ C	⟨7,7⟩ X	⟨8,7⟩ C	⟨9,7⟩ W	⟨10,7⟩ X	⟨11,7⟩ A

Agent(S):
0X Key
0X Arrow
S: Agent
A: Arrow
K: Keys
D: Door
W: Wumpus
P: Pit
HP:Half pit
C: Crate
H: Half crate
X: Wall

Figure 28: Eastward Movement with Push Action, Double Pushable Condition and Pit Interaction.

In Figure 29, our agent has applied the push action again in the east direction. Since only one half crate was pushed, the pushable condition was sought to be satisfied. The half crate in cell ⟨7, 2⟩ was pushed to cell ⟨8, 2⟩, and cell ⟨7, 2⟩ became empty. At the same time, both the half crate and half pit in cell ⟨8, 2⟩ disappeared because a half crate was pushed onto the half pit, leaving cell ⟨8, 2⟩ empty and movable. In the final state, our agent is in cell ⟨7, 2⟩.

In Figure 30, our agent is moving from cell ⟨7, 2⟩ to cell ⟨8, 2⟩ with the move action in the east direction. As a condition, the connected rule of cells has been taken into account.

In Figure 31, our agent has moved eastward from cell ⟨8, 2⟩ to cell ⟨9, 2⟩ using the move action. The agent has paid attention to the connected rule while making the move. Upon completion of the move, the agent now possesses the Key in cell ⟨9, 2⟩, and the Key is no longer present in the cell.

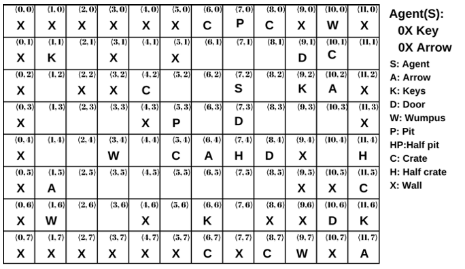

Figure 29: Pushing Half Crate in East Direction.

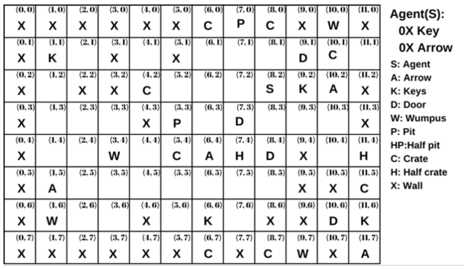

Figure 30: Agent Moving from ⟨7, 2⟩ to ⟨8, 2⟩ with Eastward Move Action.

In Figure 32, our agent has applied the unlock action in the northward direction. The agent has followed the connected rule while doing so. Additionally, the agent must have at least one Key to perform this action, which is satisfied as the agent has collected a Key in the previous action. Upon completion of the action, the door in cell ⟨9, 1⟩ will disappear, making it accessible for movement. Moreover, the number of Keys held by the agent will decrease by one.

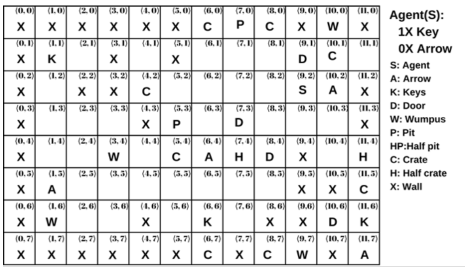

Figure 31: Agent's Eastward Movement and Obtaining the Key.

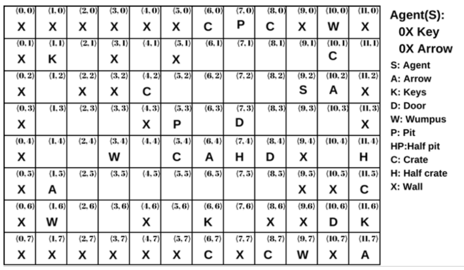

Figure 32: Unlocking a door with a key in northward direction.

In Figure 33, our agent has moved northward from cell ⟨9, 2⟩ to cell ⟨9, 1⟩ using the move action. The agent has been mindful of the connected rule while completing this action.

In Figure 34, our agent has used the push action to push the crate in cell ⟨10, 1⟩ to the eastward direction, resulting in the crate being moved to cell ⟨11, 1⟩ and becoming immovable. The agent has paid attention to the pushable rule while completing this action. Upon completion of the action, our agent is now located in the cell ⟨10, 1⟩, which has become empty and accessible due to the movement of the crate.

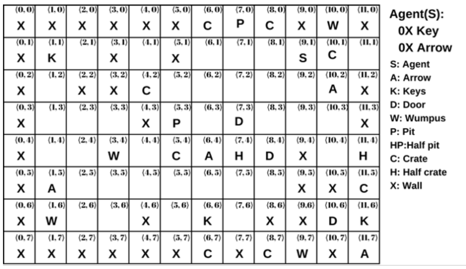

Figure 33: Northward Movement from Cell ⟨9, 2⟩ to Cell ⟨9, 1⟩.

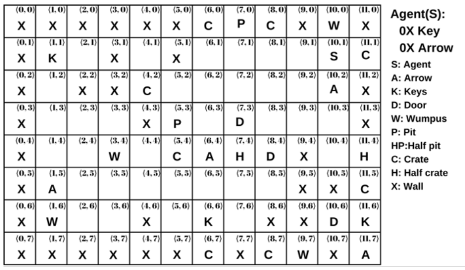

Figure 34: Pushing Crate to the East.

In Figure 35, our agent has performed a move action and moved from cell ⟨10, 1⟩ to cell ⟨10, 2⟩ in the south direction. During this movement, the agent has acquired the Arrow located in cell ⟨10, 2⟩ and the Arrow is no longer present in that cell. The connected rule was followed while performing this action.

In Figure 36, our agent has performed a move action and moved from cell ⟨10, 2⟩ to cell ⟨10, 1⟩ in the north direction. The connected rule was followed during this movement.

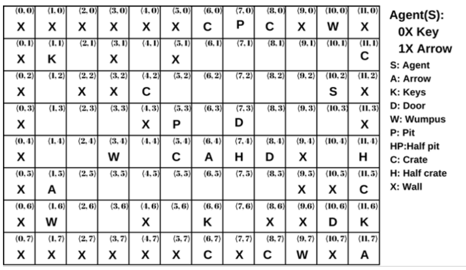

Figure 35: Agent's Southward Move and Arrow Acquisition.

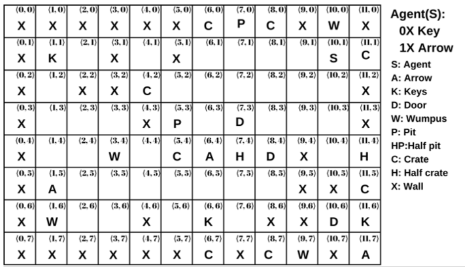

Figure 36: Agent moves north from ⟨10,2⟩ to ⟨10,1⟩.

In Figure 37, our agent is using the shoot action to hit the wumpus in the northern direction located at ⟨10, 0⟩. The connected rule and having at least one arrow rule were considered for this action. After this action, the ⟨10, 0⟩ cell is now empty and can be moved into. However, the number of arrows the agent has has decreased by one.

In Figure 38, our agent is using the move action to move from ⟨10, 1⟩ cell to the ⟨10, 0⟩ cell in the northern direction. The connected rule was taken into account while applying this action.

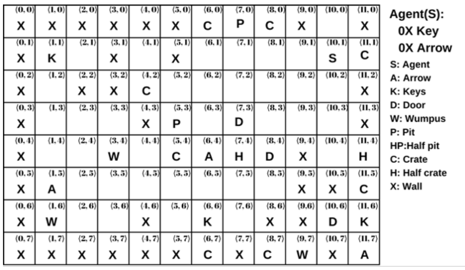

Figure 37: Agent shooting the wumpus in the north direction .

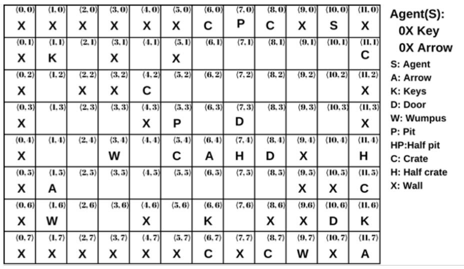

Figure 38: Agent's Northward Move from ⟨10, 1⟩ to ⟨10, 0⟩.

Our agent has reached one of the cells on the edges of the map. Therefore, by making a final move in the direction of north, which was the direction of the last action, the agent exits the map and the goal is achieved. The final map is indicated in Figure 39.

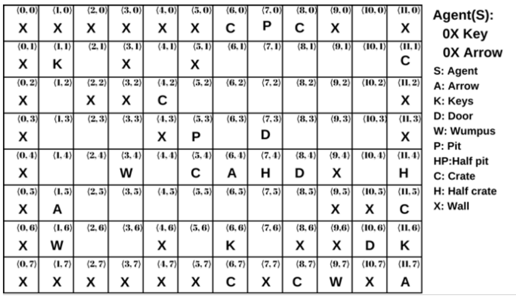

Figure 39: Agent Reaches Map Edge and Achieves Goal.

6 Finding Solution

The goal of the project is to develop a plan for an agent escaping from the Wumpus world. A* search, best first search, and greedy search algorithms were investigated to find the plan. Among these algorithms, greedy search was chosen.

A* Search: The A* search algorithm is commonly used for finding the shortest path problem. This algorithm considers the difference between the cost of a node to reach the target node and the cost of the path it passed from the starting node. This way, the most cost-effective path to reach the target node is found. However, this algorithm can sometimes work very slowly, especially when working with large datasets.

Best First Search: Best first search is commonly used in finding the best path problems. This algorithm takes into account the distance to the target node and prioritizes the closest nodes. However, in some cases, going to the closest nodes may require a longer path in the end.

Greedy Search: Greedy search uses a heuristic function to find the next best move. This function estimates the distance or cost to the target and prioritizes the most suitable node. This algorithm works very quickly and skips high-cost calculations. However, greedy search may not always be the best path and can lead to misleading results.

The goal of the project is to develop a plan for an agent escaping from the Wumpus world. The global shortest path is not targeted. Considering the hardware and time constraints to be used in the project, Greedy search was preferred. Greedy search works very quickly and skips high-cost calculations. In addition, the most suitable heuristic function can be selected for the Wumpus world, and greedy search prioritizes the best nodes using this function.

7 Conclusion

In this report, we aimed to develop an agent planning strategy to escape from the Wumpus world. Considering the hardware and time constraints, the Greedy search method was preferred. Greedy search works quickly by skipping high-cost calculations. Additionally, the most suitable heuristic function for the Wumpus world can be selected, and Greedy search prioritizes the best nodes using this function.

This report also provided information about some of the challenges specific to the Wumpus world. For example, the agent needs to detect the dangers around it to avoid the Wumpus. To overcome these challenges, a set of rules was established when determining the agent's movements and decisions. These rules were designed to help the agent escape from the Wumpus world.

In conclusion, this report provides a basic framework that can be used to develop an agent planning strategy to escape from the Wumpus world. With further development, this strategy can be used in other challenging environments, such as the Wumpus world, and provide solutions for similar problems.

References

[1] Yusra Alkhazraji, Ron Alford, Ben Johnson, and Wheeler Ruml. Pyperplan. https://doi.org/10.5281/zenodo.3700819, 2020. doi: 10.5281/zenodo.3700819.

[2] Repository for Assignment 6: Find a plan to leave the Wumpus cave. https://gitlab.rrze.fau.de/wrv/AISysProj/ws2223/a6-escape-wumpus-world/assignment.

[3] Fast Downward. https://www.fast-downward.org/, (visited on 02/15/2022).