Reza Basseda

Wumpus Game

Problem 1:

The famous Artificial Intelligence [2] describes the Wumpus world problem. The task is to develop a program to solve the Wumpus problem.

Solution:

Introduction:

In artificial intelligence, an **intelligent agent** (**IA**) is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals (i.e. it is rational). Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent, as is a human being, as is a community of human beings working together towards a goal.[1]

According to the above definitions, for the environments in which the complexity is too high to be modeled in terms of abstract states, intelligent agents are required to keep a various of level of knowledge about the environment and decide about their actions according to that knowledge representation.

Due to different levels of complexity, different levels of expressiveness are required for the logic, used to represent the environment. For the simple Wumpus world game, the simple prepositional logic could be applied in order to design an intelligent problem solver agent.
The Wumpus world is a cave consisting of rooms connected by passageways. Lurking somewhere in the cave is the Wumpus, a beast that eats anyone who enters its room. The Wumpus can be shot by an agent, but the agent has only one arrow. Some rooms contain bottomless pits that will trap anyone who wanders into these rooms (except for the Wumpus, which is too big to fall in). The only mitigating feature of living in this environment is the possibility of finding a heap of gold.[2] Figure 1 shows a representation of this problem.

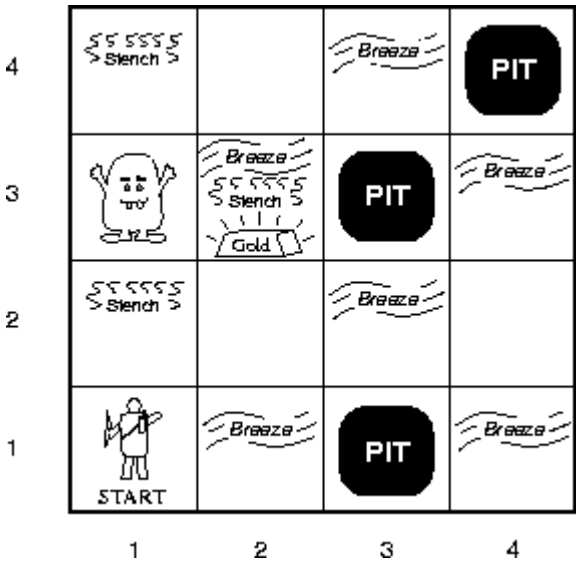

Figure 1: The sample Wumpus world[1]

To design an intelligent problem solver for this problem, we design a simple agent using prepositional logic. Our agent described the environment using a simple preposition of different types as mentioned in Table 1:

Preposition	Meaning
P	Pit
G	Glitter
O	Ok, it is safe.
B	Breeze
S	Stench
V	Visited
W	Wumpus
Table 1: Prepositions of KB	

The agent's knowledge base used theses prepositions for each room to represent the environment and decide the most appropriate action upon to the positions of objects.

Program's Structure:

The program has two main parts which together represent the game played by the agent embedded in the program.

The first is the *cave* object, instantiated from *Cave* class. This object simulates the environment in a

way that it gets actions from agent and provide sensory information in terms of *perception* objects of *Perception* class. The perception objects just provide a set of boolean values for 5 different variables in the sensory tuples.

An *agent* was an object of class *Agent*, consisting of a body which define the search function and action choosing procedures and a *kb* object of *KnowledgeBase* class, representing of agents knowledge about the environment.

For this problem we developed a program, searching rooms in the cave upon to state of the room according to the knowledge acquired during execution. It searches the cave until get the gold using an A* search algorithm. In order to search and find the most appropriate neighbor room to move, we use an *H(n)* function, implemented in four different methods in the program. Those functions are named getHDirect, getHLeft, getHRight and getHBack. While we find the most appropriate room to move, we choose the corresponding set of actions to get that neighbor.

The *KnowledgeBase* class consists of a set of *clause* objects, instantiated from *Clause* class. These clauses represents clauses in the *KB* and each of consists of *prepositions*, objects of class *Preposition*. *KB* consists of several principal rules, which are listed in Table 2. These rules are used to deduce new facts about environment upon to the facts fed by agent according to its perceptions.

`V(x,y) => O(x,y)`
`~P(x,y) & ~W(x,y) => O(x,y)`
`~B(x,y) => ~P(x',y') s.t. (x',y') is (x,y)'s direct neighbor`
`~S(x,y) => ~W(x',y') s.t. (x',y') is (x,y)'s direct neighbor`
Table 2: Rules of KB

Methodology:

We considered our systems in terms of number of facts in *KB* in different program steps. We also considered number of clauses in *KB*. The number of clauses shows that our system can deduce more knowledge for making decision.

We also considered the number of steps of resolution taken for each action decision. These amount of steps would increase if the environment got larger.

Results*:*

To execute the program, the first command line parameter should be entered as an integer to determine the format of out put. If the parameter is 0, the output will be simply the number of facts. If it is 1, it will show the maze and navigator agent.

The execution of program in verbose mode shows a trace of agent's navigation in the cave and the

position of different objects as below. The direction of agent also has been shown in the trace.

```
Time = 0
| | | |*|
|W| |*| |
| | | | |
|>| |*| |
```

In statistics mode of program, you can have different statistics about agent's *KB* during execution. Figure 2 shows how agent's *KB* grows during a sample run. In our experiment agent could grab the gold with 986 scores. Figure 3 also shows the number of steps taken for resolution.

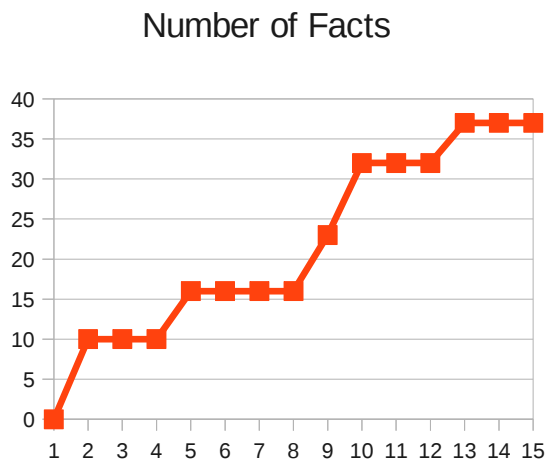

Figure 2: Number of Facts

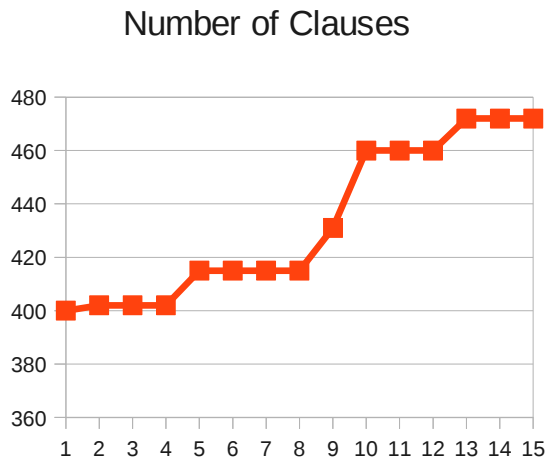

Figure 3: Number of Clauses

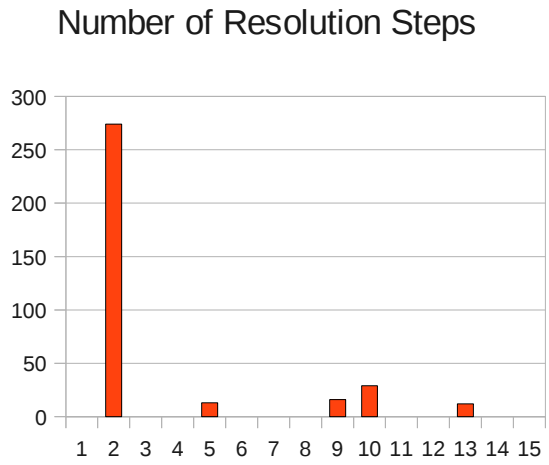

Figure 4: Number of Solution Steps.

Conclusion:

As shown in above graph, the agent's knowledge grows during execution and finally agent could get the gold without being fallen in pits or eaten by Wumpus. The logical agent in the first stages got a lot of information about environment and during its 4 moves it could be able to get 90% of information it required.

If we enlarge the environment, during navigation agent could get more facts based upon the number of

rooms it has been visited.

References:

[1] http://en.wikipedia.org/wiki/Intelligent_agent

[2] Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.