

# Logic Programming Assignment

## Individual Assignment

Due: ~~Feb 22<sup>nd</sup>~~ March 1<sup>st</sup> 11:59pm

For this assignment you will be developing a variation of the Wumpus World. The goal of the agent is to navigate through the world and exit the world with as much gold as possible. The world will be composed of squares with pits, gold, and a Wumpus. The environment will be static, i.e. it will not change when the game begins. The neighboring squares to Pits will have a breeze and the ones next to the Wumpus will have a stench, just like in the book. You should assume that the map will be entered through a file called: wumpus\_world.txt (please make sure to use this name since we will be testing your program using our own files and will expect to follow the same format and file name). An example world is described below; please follow the same configuration for your assignment.

In particular, for this assignment, you will need to develop:

- An Agent: this will be your AI character that will need to navigate the world avoiding Pits and the Wumpus but maximizing the amount of gold it can collect before it exists. *You have freedom in terms of how greedy you want the agent to be.* The agent will take the world file as an input (wumpus\_world.txt)
- Generate worlds: for the sake of experimentation, you will need to generate some worlds that you can use as input, please keep track of these worlds and submit them with your assignment. You can generate these by hand or algorithmically. It is up to you.

For the agent, you will implement a hybrid algorithm that uses a combination of logical inference, search, and domain specific routines. You can use a different algorithm of your own design with the following two conditions:

1. You must interface with a knowledge base that implements a tell-ask interface. You should be able to assert percepts through the interface and ask queries about whether a cell is safe. The knowledge base should be able to take as input first order sentences and answer queries in first order logic. For this, you can use a first order theorem prover called Prover-9 (<http://www.cs.unm.edu/~mccune/prover9/>). See the site for more documentation on this theorem prover. However, you can also build your own or use any other theorem provers.
2. You need to implement a search algorithm (A\* or some other search algorithm) to plan a route to a given square and interface it to the agent.

You should develop a report explaining your solution. In your report, you should describe the approaches you tried with pseudocode. Also, describe how you engineered the environment in FOL. Include any other comments or problems you encountered and how you resolved them.

### The world and representation:

- Here is a sample map in txt:  
A11  
B21  
P31  
B41  
S12  
B32  
W13  
S23  
G23  
B23  
P33  
B43  
S14  
B34  
P44  
M44  
GO11 [EOF]
- Semantics of the file representation:
  - Axy = Agent position is x,y (starting position is facing right)
  - Bxy = Breeze at x,y
  - Gxy = Gold at x,y
  - GOxy = Your agent's goal is to reach x,y
  - Mxy = Map of size x,y
  - Pxy = Pit at x,y
  - Sxy = Stench at x,y
  - Wxy = Wumpus at x,y
- For this assignment you will use First Order Logic to represent the environment and for reasoning in the world.

### Game Scoring:

- you get a -1 score for every move (a move from one block to another).
- -100 for shooting an arrow (you get only 1 arrow so use it well).
- 1000 points for gold.
- Game over on dying (by the Wumpus or Pit).

### Code:

- Use Python as the programming language for this assignment. *If you choose to use a programming language other than Python, please email the TA for approval.*
- For the assignment, you can either use GUI or text based interface to show the output.
- The output should show a TRACE of steps (or the solution of A\* and the trace of how the world is explored)

You may read the entire map at one time, but you are not allowed to look at a block until your agent reaches that block. Think means unless you reach 2,3 your program isn't supposed to know that there is Gold at 2,3. (The only exception to this is that you are allowed to read Mxy, Axy and GOxy)

In order to ensure that you aren't reading ahead, we need you to display every step that your agent makes. Ex -

- move 2,1
- found breeze
- move 1,1
- move 1,2
- found stench
- deduced wumpus at 1,3
- deduced pit at 3,1
- move 2,2

**What to submit:**

- Code
- Readme file: includes instructions to run your game
- Report: up to 2-page document describing the approaches you tried with pseudocode. Also, describe how you engineered the environment in FOL. Include any other comments or problems you encountered and how you resolved them.
- Example world files you have experimented with

**Where to submit your assignment:**

- Please submit your deliverables into your SVN repository inside a folder name 'assgn2'.
- The path for the SVN repo is –
  - [https://trac.ccs.neu.edu/svn/aispring2013/bbb\\_aaa](https://trac.ccs.neu.edu/svn/aispring2013/bbb_aaa)
  - Replace 'bbb' with the first 3 characters of your last name
  - Replace 'aaa' with the first 3 characters of your first name (use your middle name here if you have one)

**Grading Criteria:**

- Report highlighting approach and experimentation (10 points)
- Correctness (60 points)
- Code Readability/modularity (10 points)
- Creativity in the solution (20 points), this needs to be highlighted in the report.
- Extra credit will be given to assignments that go beyond the requirements stated here.