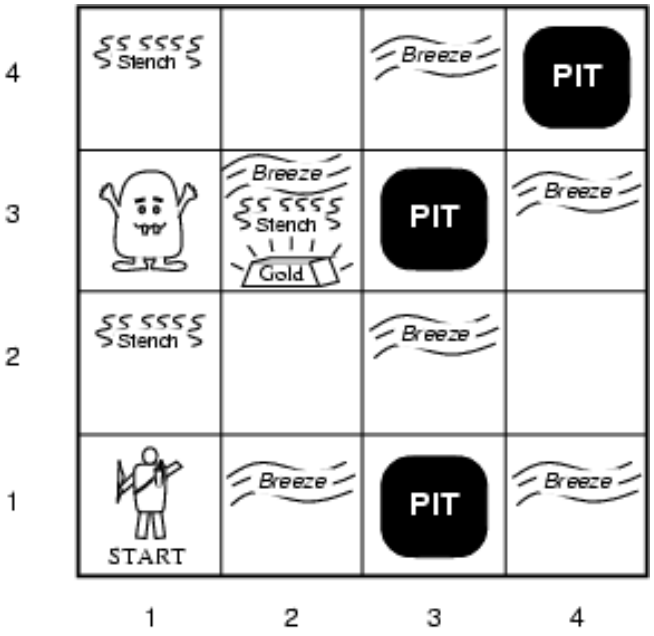
Problem Set 3 – Designing Logical Agents

Wumpus World

As discussed in class, 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 (a good thing to have with US Federal Reserve Policy of monetary stimulus). Although the Wumpus world is rather tame by modern computer game standards, it provides a good test bed for designing intelligent agents. Credits: Mike Genesereth.



PEAS description:

- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
  - The agent always starts at the origin of the Wumpus world
  - There is one Wumpus placed randomly in any room in the Wumpus world except the origin.
  - Each room except the origin has a probability of 0.2 of having a pit.
  - Squares adjacent to the Wumpus are smelly
  - Squares adjacent to the pit are breezy
  - Glitter *iff* gold is in the same square
  - Shooting kills Wumpus if you are facing it
  - Shooting uses up the only arrow

  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
  - You need to define characteristics of environment, i.e., partially observable, etc.
- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

Assignment

a) Create a knowledge-based Wumpus Agent that uses propositional logic to keep track of all relevant facts within its knowledge base. This will require defining propositional rules of the Wumpus World (document in your report). I have provided a Java project that implements a propositional logic knowledge base that can perform inference using forward chaining (aima-propositional-logic.zip). You can use this software to define your initial knowledge base, add new knowledge as you explore your environment, and perform inference to identify safe rooms to move to as part of your search procedure.
b) Create two Wumpus search agents that use the Knowledge base. As your agents explore the Wumpus world, they add percepts to the KB and perform inference on the KB to determine if neighboring rooms (the fringe) are safe. One agent should just stop if it cannot identify a safe room to move to (Chicken Little Agent). The other agent should randomly select a room (Rambo Agent) if it cannot determine if any adjacent room is safe. For best performance you will need a search algorithm that can backtrack. BFS or DFS should work fine, however your fringe should prioritize the safety of the fringe rooms.
c) Perform several random trials of both agents and analyze their respective performance, i.e., success rate, steps to goal, and identification of un-winnable environments.
d) Bonus:
    a. Enhance your Knowledge-based agent to use a probability score to select the "safest" room when the knowledge agent cannot identify with certainty that a room is safe.
    b. Surprise me!

Tips on creating a Knowledge-based Agent

- Unzip aima-propositional-logic.zip
- Run the demo: aima.logic.propositional.PLFCEntailsDemo
- Sample usage from example in class:

```
private static PLFCEntails plfce = new PLFCEntails();

KnowledgeBase kb = new KnowledgeBase();
kb.tell(" (P => Q)");
kb.tell("((L AND M) => P)");
kb.tell("((B AND L) => M)");
kb.tell("( (A AND P) => L)");
kb.tell("((A AND B) => L)");
kb.tell("(A)");
```

```
kb.tell("(B)");
String query = "Q";
System.out.println( plfce.plfcEntails(kb, query) );
```

Submission:
Quality of reports and analysis will be rewarded. Submit report, results, and analysis as a PDF file with your name, class, and problem set number along with your project archive to Blackboard. A demo is required!