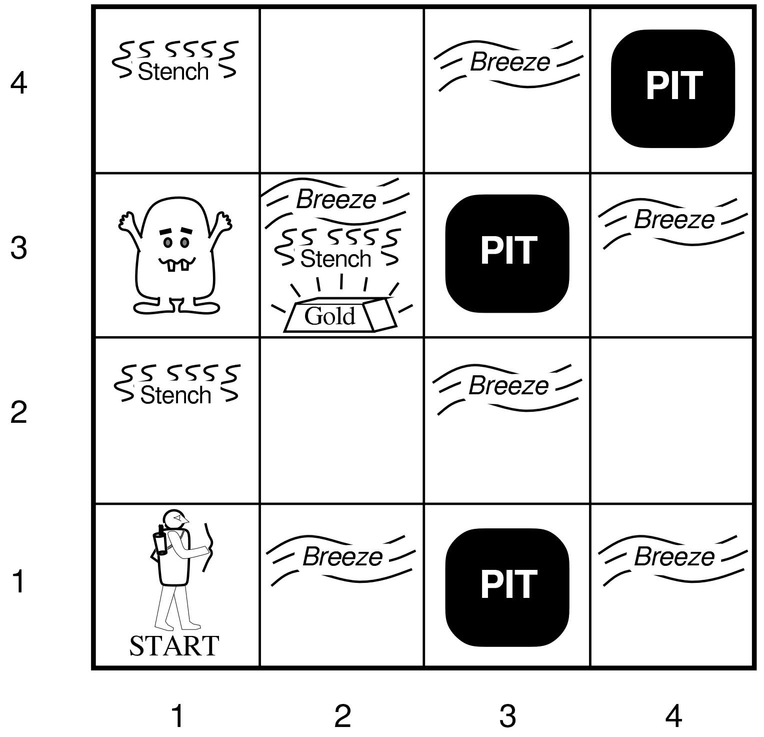
AI Project: Wumpus World in JESS[1]



In this assignment, you will utilize Jess to represent knowledge about the Wumpus World, and to reason about it.

The core system for the Wumpus World is available through the file `ww.jess`, in combination with some sample files that provide facts for a particular instance of the Wumpus World, such as `cave0.jess`, `cave1.jess`, `cave2.jess`, `cave3.jess`. These files are available as a single zip file from the course website.

The log below shows an example of the WW simulator for the `cave0.jess` instance. It assumes that you're running Jess from the directory that contains the Wumpus World files.

```
% jess

Jess, the Rule Engine for the Java Platform
Copyright (C) 2008 Sandia Corporation
Jess Version 7.1p2 11/5/2008
```

[1] This text was taken with minor modifications from Prof. Kurfess' artificial intelligence course at the California Polytechnic State University, San Luis Obisopo, to be found at `http://http://users.csc.calpoly.edu/~fkurfess/Courses/CSC-481/W10/Assignments/Wumpus-World.html`

```
This copy of Jess will expire in 1412 day(s).
Jess> (batch ww.jess)
TRUE
Jess> (batch cave0.jess)
TRUE
Jess> (reset)
TRUE
Jess> (run)
GENESIS...
Xena enters the caves at (1,1).
Adding adj asserts for a 4 by 4  world.
SIMULATING...
SENSING...
Xena notices (1,2) nearby.
Xena notices (2,1) nearby.
Xena sees no glitter.
Xena feels no breeze in (1,1).
Xena smells nothing.
THINKING...
No stench in (1,1) means no wumpus in (1,2).
There's no breeze in (1,1) so there's no pit  in (1,2).
Xena somewhat wants to go to (1,2).
With neither wumpus nor pit, (1,2) is safe.
Xena strongly wants to go to (1,2).
There's no breeze in (1,1) so there's no pit  in (2,1).
No stench in (1,1) means no wumpus in (2,1).
Xena somewhat wants to go to (2,1).
With neither wumpus nor pit, (2,1) is safe.
Xena strongly wants to go to (2,1).
Seeing no glitter, Xena knows there is no gold in (1,1).
Since Xena is in (1,1) and not dead, it must be safe.
(1,1) is safe, so there's no pit or wumpus in it.
PLANNING...
ACTING...
Xena goes to (2,1).
SIMULATING...
SENSING...
Xena sees no glitter.
Xena feels a breeze in (2,1).
Xena notices (2,2) nearby.
Xena notices (3,1) nearby.
Xena smells nothing.
THINKING...
```

Your Tasks

The example system works fine in most of the provided cave instances, but not for `cave3.jess`. You need to modify the system as described below.

Improve the Hunter's Reasoning Ability

The basic method to use Jess for reasoning is by designing a set of rules that examine certain conditions, and then draw conclusions to perform actions. The current code has these rules to draw conclusions from sensing a stench in a cave.

```
(defrule evaluate-stench
  (task think)
  (cave (x ?x) (y ?y) (stench TRUE))
  (adj ?x ?y ?x2 ?y2)
  ?f <- (cave (x ?x2) (y ?y2) (has-wumpus UNKNOWN))
  =>
  (printout t "With stench in (" ?x "," ?y
              "), maybe the wumpus is in (" ?x2  "," ?y2 ")." crlf)
  (modify ?f (has-wumpus MAYBE)))
```

It marks every adjacent cave to the stench as possibly containing a wumpus (except for caves that we already know are wumpus free or have a wumpus in them). But, we need to make a stronger statement. Not only *could* one of the caves have a living wumpus, one of them *must* have it. As we learn more information about the cave, we may be in a situation in which there is only one adjacent cave left that might have a wumpus. If so, then it does have a wumpus. Your task is to add a rule or rules to allow the system to deduce the exact location of the wumpus when possible. Do the same for pits. You might find Jess's query facility useful in this; check out the manual at `http://www.jessrules.com/jess/docs/71/`.

New Top Level Goal – Killing the Wumpus

The hunter in the current system has just one goal – to explore the caves until she finds and picks up the gold, then leave the caves. Extend the system to add another goal – eliminating the wumpus. To do this, you can add another slot to the hunter to represent the number of arrows she has:

```
(deftemplate hunter "A hunter"
  (slot agent (default "Xena"))
  (slot x (type INTEGER))
  (slot y (type INTEGER))
  (slot gold (default 0) (type INTEGER))
  (slot alive (default TRUE))
  (slot arrows (type INTEGER) (number 1))
```

Add a new action, `shoot`, which uses up an arrow. If the hunter is in a cave immediately adjacent to a cave with a wumpus, then that wumpus will die a horrible, and deserved, death. Dead wumpi will continue to stink, but they are not dangerous. If the hunter enters a cave with a wumpus corpse, she will not die, although the stench will be hard to get out of her clothes. The goal of the hunter is now to find some gold *and* kill the wumpus (in either order) and then leave the caves.

Improve the Hunters Ability to Go to a Given Location

Study the hunters behavior in the world defined by `cave3.jess`. You will notice that the technique we have used to get the hunter to a distant cave does not work and she is stuck in a cave and cannot figure out what to do. Add improved rules to deal with such cases so that the hunter can form a goal to go to a distant cave and will do so, if possible. For this task you can (and should) exploit the fact that you can add additional pieces of information to the knowledge base, such as routes, or first steps in the right direction, when going towards a certain goal tile.

What are you allowed to modify?

Much of the code deals with simulating the world. Please pay special attention to which parts belong to the simulation and which parts belong to the thinking of the hunter. These can be relatively easily distinguished by paying attention to the `task` premise. In some rare cases it might be necessary however to also make modifications to the simulation, for example to be able to simulation the hunter's shooting of an arrow.

What to Hand in

Please prepare an informative presentation, explaining the game mechanics to your colleagues and document your decisions, please also exemplify the necessary changes to the code and a quick “live demo” of one of the caves.

Your submission (to be turned in electronically) should consist of a ZIP file that in turn contains one PDF with code segments that show your modificiations to the original code, and a brief description of why this code achieves the objectives of the exercise as well as the presentation slides and the code itself.

Acknowledgement

This assignment is based on a homework assignment in the CMSC471 Artificial Intelligence – Fall 2001 course at University of Maryland, Baltimore Counties Homework Nine - Jess meets the Wumpus, and a revised version of the assignment from the University of Queensland’s course COMP 3701, with minor adaptations by Prof. Kurfess.