

CSCE625: Artificial Intelligence

Programming Assignment 4 : Wumpus world in Prolog

Dylan Shell

Oct 16th, 2014

1 Problem Domain

You will be considering the Wumpus world introduced in Russell and Norvig (2009) Chapter 7. For this programming assignment you'll use Prolog's ability to do inference in order to write an agent that will make safe moves within its world.

2 Project Infrastructure

You will need to download and install your own version of SWI-Prolog. It is freely available from <http://www.swi-prolog.org/> for major operating systems. Major Linux distributions have copies available within their package system.

1. Download the following three files:–

- (a) http://robots.cs.tamu.edu/dshell/cs625/prolog/wumpus_world.pl
- (b) <http://robots.cs.tamu.edu/dshell/cs625/prolog/utills.pl>
- (c) http://robots.cs.tamu.edu/dshell/cs625/prolog/my_agent.pl

2. The first two files comprise a simulator for the Wumpus world. The third is an empty agent skeleton program. Your assignment is to flesh out the details in `my_agent.pl`. You are not permitted to make modifications to either of the other two files.

3 Running the code

Start SWI-Prolog:

```
pl
```

Load the world simulator:

```
?- [wumpus_world].
```

Load the agent:

```
?- [my_agent].
```

Run the agent:

```
?- evaluate_agent(1, Score, Time).
```

Correct to: `"wumpus:evaluate_agent(1, Score, Time)"? y`
If you read the output, you'll see the default agent just moves forward. It does this in the world shown in Russell and Norvig's Figure 7.2, which causes the agent's death rather soon. (You use `^D` to exit the programming environment.)

You will need to edit `my_agent.pl` to produce a more rational movement. To do this you will need to define an intelligent `run_agent(Percept, Action)`. The percepts are provided as a list of five elements `[Stench, Breeze, Glitter, Bump, Scream]`, where you receive a "yes" or a "no" in the respective position. There is also an `init_agent` function which you can use for initialization.

The simulator knows the total state of the world, while your agent will know only what it senses through its perceptions. Initially the agent knows only the perceptions in position (1,1) and in order to acquire other information must be moved in other cells of the grid. Your agent must maintain its own state, containing all the information that gradually becomes available. The state will have to be updated with the new perceptions every time an action is executed, and this information must be represented so as to permit reasoning (via inference) as to what operations are safe.

4 Submission

Due date: 30 Oct at 11:59pm.

Submission method: Via e-mail to the TA.

Submit in the following (electronic) form:

- Turn in a zip file which includes all the source files and a documentation like pdf file, with the items below. (Do not describe your results in the e-mail)
- The zip file should be named `student-last-name.hw-number.zip`
e.g., Jim_hw4.zip
- Subject of e-mail should be named as `[csce625]last-name.hw-number`
e.g., [csce625]Jim_hw4

The zip should include a single PDF with the following:

1. A description of how to run the submission.
2. A list of the resources used (*e.g.*, online forums, links to example code on the web, *etc.*).
3. A statement of the Aggie Code of Honor.

It must also include, separately, the code you wrote for this assignment.

You may discuss this openly with your friends and classmates, but are expected to write your own code and compile your submission independently. If in doubt about whether a resource you used should be included in item 4 above, err on the side of caution and include it.