Week 2 - Lab - Welcome to the Grid!

Intro to GridWorld

Welcome to the Week 2 Artificial Intelligence Lab! The purpose of this lab is to introduce you to the GridWorld platform that will provide a concrete environment in which to practice the concepts taught on this unit.

You will need to use a local **swipl** installation in order to run this code as **SWISH cannont be used for this**. You are also encouraged to use a text editor of your choice to edit the files (Both **VSCode** and **Atom** have support for prolog syntax highlighting and SWI has a built in editor called **PceEMacs**). If you have any trouble setting up swipl then please come to the lab or make a post in the forum.

You are provided with two important files: a zip file containing the SWI library code that runs the local webserver you will use to interact with the GridWorld; and an html file containing a detailed manual describing how to use that library (PLEASE DOWNLOAD THE FIRST FILE AND REREAD THROUGH THE SECOND):

- Lab source code (for weeks 2 and 3) - GridWorld Library: Grid_world_lab.zip (https://www.ole.bris.ac.uk/bbcswebdav/pid-9698031-dt-content-rid-52548344_2/xid-52548344_2)
- Lab user guide (for weeks 2 and 3) - GridWorld Manual : GridWorld.html (https://www.ole.bris.ac.uk/bbcswebdav/pid-9698031-dt-content-rid-52548340_2/xid-52548340_2)

The library code is stored in the directory "ailp" (standing for Artificial Intelligence with Logic Programming). The root directory contains 2 files of the form lab_*_1234567.pl where you will need to write your lab solutions by completing the predicate stubs. You may replace the 1234567 in the filenames with *your own* 7-digit student number. Note this number will be important if you do the coursework for this unit but, since these labs aren't assessed, you are free to leave it unchanged or any other NUMBER that you want (just make sure NOT to user your username as only numbers are supported and, if you use letters, the runner will throw an error saying it can't load your submission file).

You won't need to look into the **internals** of the library but you may do so, if you want. One thing you might want to do is temporarily change the default **grid size** from 10 to a smaller number such as 5 to make it easier to debug your code. This can be done by changing "X=10" to "X=5" in lines 52 and 53 of /ailp/library/game_predicates.pl; and by changing "X=25" to "X is "N*N//4" in line 79. This will allow you to make the grid a small as 3*3 (but you may encounter issues if you try to go lower). Please ensure that you fully **restart** Prolog if you make and changes to the library code.

Prolog: Lab Grid

For this exercise you will learn how to move around the grid world and ultimately use you agent to draw a spiral. The grid world is just the name of the environment that these labs will take part in. For this exercise you will be editing lab_grid_1234567.pl. Please read the Introduction to Gridworld.html **guidance notes** linked above. This lab will build on concepts explained in Chapters 3-5 of the Learn Prolog Now! (http://www.learnprolognow.org/) including **recursive** predicates (Chapter 3 (http://www.learnprolognow.org/lpnpage.php?pagetype=html&pageid=lpn-htmlch3)) on **lists** (Chapter 4 (http://www.learnprolognow.org/lpnpage.php?pagetype=html&pageid=lpn-htmlch4)) with **integers** (Chapter 5 (http://www.let.rug.nl/bos/lpn//lpnpage.php?pagetype=html&pageid=lpn-htmlch5)).

Possible Queries

The following queries have been exported by the library code:

Query Name	Description
start	Initiate the web server
shell	Open the command shell
my_agent(-A)	Returns the ID of your Agent
ailp_grid_size(-N)	Returns the size of the grid
get_agent_position(+A,-Pos)	Returns the position of Agent A
agent_do_moves(+A,+L)	Makes Agent A perform the list of moves L

How to run?

In order to run the grid world, you need to open ailp.pl using swipl. This exercise is known as lab grid so we will pass that as a command line argument (e.g. **swipl ailp.pl lab grid**).

Upon loading you will likely see a lot of warnings about singleton variables. This is because the skeleton definitions in lab_grid_1234567.pl have not yet been properly implemented.

Next, Type in **start.** and hit enter to start the webserver. If you press anything but 'N'/'n', then the webpage will open automatically in your default browser. **Hit the run button on the webpage before you continue.**

We will use a shell to initialise the grid so type in **shell.** and then **setup.** in the command prompt. (Note that while the main prolog prompt is **?-**, in the shell it will look like **?**).

After the setup/0 has finished, you can exit the shell using either **stop.** , **CTRL+D** or **CTRL+C** followed by **a** (to abort) at the prompt.

Identifying your Agent

At the top left of the grid, you will be able to see a coloured shape. This is your agent. For the first task in this exercise, you need to find a query that allows you to identify your agent and their current position.

Create predicate m/1

The first step of moving around the grid world is defining the directions that we are able to move. They are **n**, **e**, **s** and **w**. For the first part of this exercise, you will need to define the predicate m/1 such that m(A) is true when A is one of the four valid directions.

Create predicate on_board/1

Next, we need to create the predicate on_board(+P) that tells us if a position P is on the board. Positions are stored as a compound term p(X,Y) where both X and Y range from 1 to the grid size inclusive.

TIP: You can use the predicate ailp_grid_size/1 to find the size of the grid.

Create predicate pos_step/3

For this step we need to find the new position after an agent has made a possible step (although we don't need to check the validity yet!). The full predicate is pos_step(+Pos,+Dir,-NPos).

Create predicate new_pos/3

We are now able to combine the previous steps together to find the new position after a move has been made as well as to check that the move is valid. new_pos(+Pos,+M,-NPos) should be true if moving from Pos in direction M will lead to NPos and NPos is on the board.

Create predicate complete/1

The predicate complete(+L) should be true if the Length of L is equal to the number of cells in the grid. The size of the grid **SHOULD NOT** be hardcoded.

Create predicate spiral/1

The final part of this exercise is to implement the predicate spiral(-L) which will move the agent from the start position p(1,1) towards the center of the grid in a spiral pattern. L should be the list of steps that are taken by the agent to reach the center **including** the start position (You can test this using complete/1)

TIP: In order for the agent to actually move, you will need to use agent_do_moves(A,L) where A is the identifier of your Agent and L is a path **that does not include** your current position.