CISC 3410 Fall 2012, Project 1

1 Description

This project uses search techniques to plan the path taken by a robot in a simulated Netlogo world. You will start with a Netlogo program that uses breadth-first search to plan the path for the robot, and you will:

1. Experiment with the program;
2. Extend the program to use more advanced search techniques; and
3. Evaluate how much more efficient the more advanced techniques are.

2 Get ready

1. Download the file `robots.nlogo` from the course web site and open it in Netlogo.
2. Press **setup**.
3. The default is to set up a 25 by 25 patch world, with a randomly positioned robot (little robot symbol) and goal (red patch), and 750 randomly positioned obstacles (black patches).
4. When you press **go** the robot uses breadth-first search over the patches looking for the goal and then follows the path it has found to get to the goal.
5. With the switch **show-visited** on, the set of patches that have been searched are colored yellow.
6. With the switch **show-agenda** on, the set of patches that are on the agenda are colored blue (and they will later change from blue when they are no longer on the agenda).
7. With the switch **show-path** on, once the robot has found a path to the goal, the path is colored purple.
8. **reset** places the robot at its start position and forgets about any search it just did (this is useful in debugging to allow you to do over some situation that was causing problems).
9. Play with the simulation until you are comfortable with what it is doing.

3 Experiment

1. Use **setup** to generate a random location for the robot and the goal.
2. Press **go** and run the robot until it reaches the goal.
3. Make a note of the number of visited nodes.
4. Repeat this 19 times, so you have a set of 20 measurements.
5. Compute the average number of nodes visited and the standard deviation of the number of nodes.

   (If you don't know how to compute the standard deviation, look it up. The computation for a small sample like this is trivial in a spreadsheet).
6. In the **Information** tab, document the results from your experiment.

4 Extend

1. Look at the code in the **Procedures** tab.
2. The key function is `get-plan`, this is the place where the search takes place.
3. You then need to modify the code to do the following:
   - Provide a heuristic that can rate each patch and be used to guide the search for the goal.
   - Give the option of using greedy search as well as breadth-first search to find the goal.
     What I mean by "as well as" is that I want your program to give the option of using either greedy search or breadth-first search.
   - Add A* search as a third search technique (in other words when you are done your program should offer breadth-first search, greedy search and A* search).
4. While you are writing code, you may find it helpful to work with a much smaller world. However, when you are done you need to make sure that your code runs on a 25 by 25 patch world.

5 Evaluate

1. Evaluate your new search techniques by repeating the experiments you did for breadth-first search for greedy and A* search.

   You need to do this on a 25 by 25 world so that your results are comparable with those for breadth-first search.
2. Again calculate the average number of visited nodes and the standard deviation.
3. Add the results to the **Information** tab, and explain the differences (or lack of differences) between the three sets of results.

6 Finish up

1. Make sure you document your code in the **Procedures** tab, and that you write up your experiments in the **Information** tab.
2. For extra credit use the heuristic value to color the nodes on the agenda so that when the switch **show-agenda** is on, the program shows which patches have higher heuristic value.
3. Rename your program `robots-<myname>.nlogo`, where `<myname>` is your name, so my program would be named `robots-parsons.nlogo`.
4. Subit your program through the submission site.