

## Assignment 2: A Better Vacuum Robot Planner

### CS 730/830, Fall 2015

Electronic submission due at **11:30pm on Mon, Sept 14**  
Hardcopy submission due at **start of class on Tue, Sept 15**

#### Overview

You will extend your vacuum robot planner with more advanced search algorithms that should allow it to tackle much larger problems.

#### Input

Your program should now accept up to two command-line arguments:

**algorithm** one of **depth-first**, **depth-first-id** (depth-first iterative deepening), **uniform-cost**, or **a-star**

**heuristic** if the algorithm is **a-star**, this second argument will specify either **h0** ( $h(n) = 0$ ), **h1** (a heuristic you design), or **h2** (an even better heuristic you design). Your heuristics must be admissible.

#### Output

Same as before.

#### Execution

The reference solution actually has three informed heuristics implemented (**h1–h3**) and then two additional ones (**h4** and **h5**) for the graduate extension (described below).

#### Submission

Submit a brief write-up with your final hardcopy solution answering the following questions:

1. Explain each heuristic function you devised and prove that each is admissible.
2. Describe any implementation choices you made that you felt were important. Mention anything else that we should know when evaluating your program.
3. What is the time and space complexity of each algorithm you implemented? Which algorithms are admissible?
4. Provide empirical results confirming your answers to the previous question.
5. What suggestions do you have for improving this assignment in the future?

Attach to the write-up a transcript of your code solving the example cases using the validator, and a listing of your source code (2 pages per page, as with **a2ps -2**).

#### Graduate Extensions

Those in 830 must extend the base assignment in two ways. First, your planner should support two additional algorithms **ida-star** (iterative-deepening A\*) and **greedy** ( $f(n) = h(n)$ ). Second, you should implement an additional admissible heuristic function **h3** that takes the battery into account somehow.

## Evaluation

Rough guide to grading:

- 0** nothing
- 1** something but basically nothing
- 2** write-up is correct but no code works
- 3** DFID works but A\* doesn't. write-up has no discussion of heuristic anything.
- 6** Multiple significant problems.
- 7** A\* returns non-admissible solutions.
- 8** Significant problem.
- 9** Code works fine, but is not super-awesome. Write-up is fine but not super-awesome.
- 10** Everything runs smoothly and correctly. The implementation roughly on par with the reference solution even for large problems. Write-up is clear and convincing, with no errors.