Artificial Intelligence DD2380, HT 2009
Homework 1

Due: See BILDA (around Sept 9)
Where: BILDA.
Scope: This homework covers chapters 1-4.

Instructions

In the spirit of the new goal oriented grading system this homework starts with some general questions (part A) for receiving a passing grade. You are to solve Part A of the homework individually and hand it in (upload to BILDA) individually and the answers must be in your own words. Upload the answers as a single **pdf**-file.

The second part of the homework (Part B) requires you to make an implementation to solve a problem. You are encouraged to solve this part of the homework in pairs. You have to use a programming language that is freely available (Matlab, Python, C/C++, Java) and you have provide the source code as well for testing. The results of Part B should be uploaded to BILDA separately from Part A. Each upload should be **a SINGLE zip-file** with the following structure.

```
myanswers.pdf  README  code/
```

where the pdf-file contain the written answers, the README file gives instruction for how to build and run the code found in the directory code. Only upload one zip-file per pair. Make sure that the first page of the pdf-file with answers clearly states the names and emails of the people behind the report. If you want to remain anonymous in the review process (Part C) make sure that your names are only visible on the first page of the pdf-file. Also write your names in the comment field in the upload page.

The third part cannot be started until after you have handed in Part A and B. More instructions will be posted on the course web page and BILDA for how this process will look.

We expect you to follow the code of honor (http://www.csc.kth.se/utbildning/hederskodex/).

Grading

The max number of points on parts A-C is 100p. The grade is given by

A	$\geq$75
B	$\geq$60
C	$\geq$50
D	$\geq$35
E	$\geq$20

Language

The homework should be written in English.

Part A

Exercise 1, 16p

Each of the below questions is worth max 2p.

1. When can you prune a node in A* and still be sure that the search is still optimal?
2. Does the state of the system change in the vacuum cleaning world with 2 rooms if the robot is in a clean room and perform the Suck operation for one iteration?
3. Why are local maxima so problematic? Mention one way of avoiding local maxima.
4. What is the main drawback of a bidirectional search?
5. Is it necessary for a goal-based agent to know how its actions affect the world?
6. Comment on advantages/disadvantages of randomizing actions for a simple reflex agent
7. Compare the time performance of a Depth First and Breadth First search when the optimal solution (considering the solution with lowest depth as optimal) has to be found
8. Mention an advantage of a goal-based agent compared to a reflex agent.

Exercise 2, 16p

1. Propose your own agent example (not from the book) and make a PEAS description of it with a suitable level of detail. (8p)
2. Describe with your own words how different the design and implementation of a model-based reflex agent and an utility-based agent would be. Do this in general and not specifically for the example you gave above. You can use it to exemplify though if you want.(4p)
3. Discuss the implications of the above properties on the agent behavior. That is, once again in general and not for your particular example from above.(4p)

Exercise 3, 20p

1. You are designing a search algorithm for solving a maze. In that maze, you are allowed to choose one out of three movement patterns. The movement set A has 2 different movements, and the maze can be solved in 1000 movements. The set B has 8 different movements, and the maze can be solved with 4 of those movements. Finally, the set C has 16 different movements, and 2 of those movements make you reach the goal. Your computer has 1000 Mbytes of memory, and storing a maze node in memory consumes 1 Mbyte. Calculate for standard breadth-first, standard depth-first and iterative deepening depth-first:

   (a) Which movement set has the minimum memory consumption for each search method. Is there enough memory available for solving the problem? (8p)

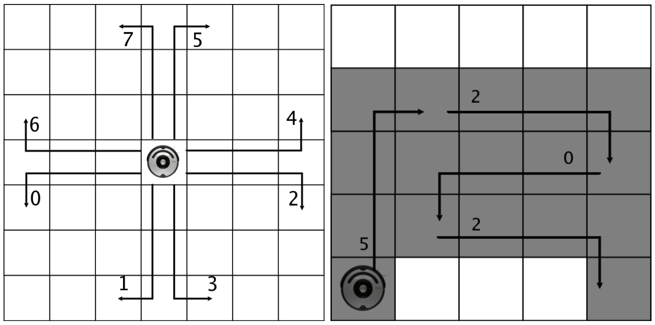
   (b) For set B, estimate the worst-case time spent in solving the maze for each of the search methods. Generating a node takes 0.001 seconds. Consider that there are no repeated states. (8p)

   (c) What would be the worst-case time if there were repeated states? How could they be avoided? (4p)

Part B

Exercise 4, 38p

You bought a cheap robot vacuum cleaner from a web shop. It looks nice, but soon you discover it has one problem: it can't go straight. It performs L-shaped movements, advancing 3 cells in a direction and then turning for one more cell (see 1(a)). However it's pretty efficient cleaning the floor; actually the floor is so clean when the vacuum cleaner passes over it that it's really shiny and slippery. So your task here will be to search a path that covers most of the floor with the particular patterns of movement described below, with the following constraint: the robot shouldn't step on a cell that was cleaned before, since it slips and it's behavior is unpredictable. The world is discretized in cells, and the robot can move in general in 8 ways when it is in a non-border cell. In Figure 1(a) you can see an example of the world with possible movements. The charge station is situated in the bottom left cell, so that is the starting point for the robot. This cell is considered already clean. With this set of movements it is usually impossible to clean all the cells in a squared-room. An example of a path in a 5x5 world can be seen in Figure 1(b). So the statement of the problem would be the following: calculate a path on a board of a given size NxN that leave dirty at most a given number d of cells.



(a) Possible movements from the center square. Each movement cleans four cells

(b) One solution for 5x5: $5, 2, 0, 2$, leaving 8 cells dirty

1. What search method do you think will find a solution in shortest time: breath-first or depth-first? Reason about the answer without taking into account experimental results.(5p)
2. Suggest a heuristic for this problem. Is it admissible? (5p)
3. Write a program that solves this problem with the depth-first method. Execute the program for a $7x7$ board leaving a maximum of 16 dirty nodes, and for a $10x10$ board leaving a maximum of 11 dirty nodes. Write the number of nodes that were expanded in each execution search.(10p)
4. Write a program that solves this problem with the breath-first method. Execute the program with the same scenarios as before. Write the number of expanded nodes and compare them with the numbers you got with depth-first search.(10p)
5. Write a program that solves this problem with the best-first method. Execute the program with the same scenarios as before. Write the number of expanded nodes and compare them with the numbers you got with depth-first search. Will the fastest method always outperform the other one?.(8p)

NOTE: Do not forget to provide the code in the zip-file you upload to BILDA. You can implement the code in Matlab, Python, C/C++, Java. Be sure that your code can compile and run in the computers in the CSC computer rooms.

Part C

Exercise 5, 10p

REVIEW, see course web and BILDA for more instructions