Lab #1

Thursday, April 4, 2024 5:33 PM

Kassandra Winter
04/04/2024

CSC 480 Kearns Spring 2024

Lab week 1a: Agents, Environments, State Spaces and Search

For each of the following examples what are the task environment attributes? Justify your answer in one or two sentences.

Performance measure
- how is performance measured?

Environment
- Is the environment fully or partially observable?
- deterministic or stochastic,?
- static or dynamic?
- single or multiple agents?

Actions
Sensors

1. A system developed to support **automated customer support** interactions. The system can automatically generate a predefined response containing instructions on resetting the password if a customer's message contains keywords indicating a password reset.

Performance measure : The performance is mesured by how accurately the system is able to detect a keyword (s) within the customer's message. It could also be measured by the accuracy/efficiency of the response generated.

Environment : The environment is partially observable because the system can only catch the given keywords, not the context given around the message.

Actions : The main action performed by the system is generating an instructional response.

Sensors : The system contains sensors that detect keywords within the customer's message.

2. A system to manage the heating and cooling of a large building

Performance measure : The performance is mesured by the system's ability to maintain a consistant desired temperature as well as the system's responsiveness to temperature changes (adjust to heat or cool accordingly).

Environment : The environment is partially observable because it can keep track of the temperature, however cannot take into account external factors (when the building is in use, time of day, etc).

Actions : The main action performed by the system is heating and/or cooling a building

Sensors : The system contains sensors that detect temperature levels of the building

3. A car's mapping subsystem to provide shortest route to a destination from current position

Performance measure : The performance is mesured by the system's ability to provide the shortest route to the user

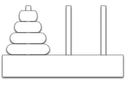
Environment : The environment is partially observable because it has access to a map, and consequently all possible routes to arrive at the desired destination, but may not have access to external factors (recent car crashes, animal crossings, etc).

Actions : The main action performed by the system is calculating the shortest route to a given destination

Sensors : The system contains a GPS sensor that provides location data

CSC 480 Kearns Spring 2024

Lab week 1b: State Spaces and Search



1. **Towers of Hanoi**

The Towers of Hanoi is a famous problem for studying recursion in computer science and recurrence equations in discrete mathematics. We start with *N* discs of varying sizes on a peg (stacked in order according to size), and two empty pegs. We are allowed to move a disc from one peg to another, but we are never allowed to move a larger disc on top of a smaller disc. The goal is to move all the discs to the rightmost peg (see figure).
In this problem, we will formulate the Towers of Hanoi as a search problem. **(5 points)**
Note: The goal here is not to write an algorithm, rather to specify the problem as a search problem.

(a) Propose a **state representation** for the problem
(b) What is the **start state**?
(c) From a given state, **what actions are legal**? What is the goal test?

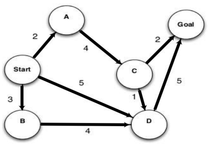
a) A state representation for the problem can be broken into three states, each state represented as the number of discs on a given peg (given each peg is ordered from smallest to largest)

b) The start state is represented by the leftmost peg containing all of the discs (ordered from smallest to largest) and the other two pegs containing none (empty).

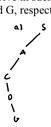
c. From a given peg, any action that involves moving a single disc from one peg to another, given that a larger disc is not allowed to be placed on top of a smaller disc.

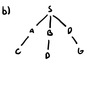
The goal test is when all discs are located on the rightmost peg, sized from smallest to largest.

2. **Search algorithms in action**



For each of the following search strategies, work out the path returned by the search on the graph shown above. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively.
a) Depth-first search.
b) Breadth-first search.
c) Uniform cost search.





CSC 480 Kearns Spring 2024

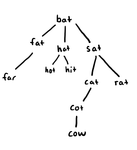
3. **Consider a word puzzle where the objective is to start with a given word and end with another given word by changing a single letter each time and making sure that the intermediate words are real words (they are in the dictionary).** For example, a sequence to derive gold from boat would be: boat, coat, colt, cold, gold.

Part a. Consider the problem where the initial word is bat and the goal word is cow. Draw a partial search space tree with at least 10 nodes and at least 2 levels after the root node. You should show at least one solution. You don't have to show an entire level.

Part b. Which would be better for this problem: depth-first search or breadth-first search? Why? Explain your answer in terms of the branching factor and the size of the total state space.

please see answer here →

a)



b) In this problem, the breadth-first search would be better due to the branching factor being low (limited to a finite number of possible words when changing only a single letter) and the total state space also being finite (making sure it finds the smallest path).

HEAD START ON Lab 2 - Work on DFS and BFS now for best retention. DO NOT SUBMIT WITH YOUR LAB 1 SUBMISSION!

The following will be duplicated and enhanced in Lab 2 and will include Dijkstra and A*. In Lab 2 you will submit you experience with all four algorithms.
The goal is to improve your intuition and understanding of the properties of these search algorithms. Your goal is to understand the advantages and disadvantages of DFS and BFS.

4. **Using the link below explore the simulations for Breadth and Depth First Search . Try out different paths and setups in an attempt to intuitively gather the properties of the two algorithms.**

Use this tool https://algo-vz.netlify.app/ to test BFS and DFS algorithms on the various mazes on the site. At a minimum you should run the simulations on the mazes: Random, Recurisve Division, Simple spiral.

Understand how the different the different grid based environment may influence the algorithms behavior behavior, Your goal is to understand the advantages and disadvantages or DFS and BFS.

In Lab 2 you will compare all four algorithms and submit screenshots that show you understanding of their behavior.