Artificial Intelligence

Name : Vivek Modi
Ru ID : 218003270

Assignment 1

Que 1)
Many of the computational models of cognitive activities that have been proposed involve quite complex mathematical operations, such as convolving an image with a Gaussian or finding a minimum of the entropy function. Most humans (and certainly all animals) never learn this kind of mathematics at all, almost no one learns it before college, and almost no one can compute the convolution of a function with a Gaussian in their head. What sense does it make to say that the “vision system” is doing this kind of mathematics, whereas the actual person has no idea how to do it?
→ Mathematical calculations like gaussian or finding a minimum of the entropy is challenging to solve a normal human. It requires some Calculations to make it easy. The use of computational models help humans to do this calculations more easily, effectively and quickly. The vision System here can be linked to as a model called open cv in machine learning where it uses a lot of parameters or so called neurons to process various image related problems. This neurons can be referred as the neurons in a human body. The human eye is linked to a lot of neurons which helps eye to process and see images and objects. This is similar to the computational use of open cv in Machine learning. Concluding, humans can process objects but can not calculate while computational software help humans to process and calculate/interpret images/objects faster.

Que 2)
This exercise explores the differences between agent functions and agent programs.

a) Can there be more than one agent program that implements a given agent function? Give an example or show why one is not possible.

→ Yes, it is possible to have more than one agent program that implements a given agent function. For example, suppose an agent that sums a list of numbers. In this case, there can be multiple possible ways to find the summation of numbers. One case could

be to add the numbers directly, the other ways could be to sort the number sand then sum the numbers.

b) Are there agent functions that cannot be implemented by any agent program?

→ Agent functions do not have logical bases hence It would be difficult to implement some agent programs. Moreover, it also depends on the ability or capacity of an agent to perform any function.

c) Given a fixed machine architecture, does each agent program implement exactly one agent function?

→ Yes, each agent program can implement exactly one agent function because each agent percepts one function at a given time

d) Given an architecture with n bits of storage, how many different possible agent programs are there?

→ If the architecture has n bits of storage, 2^n different agent programs can be possible.

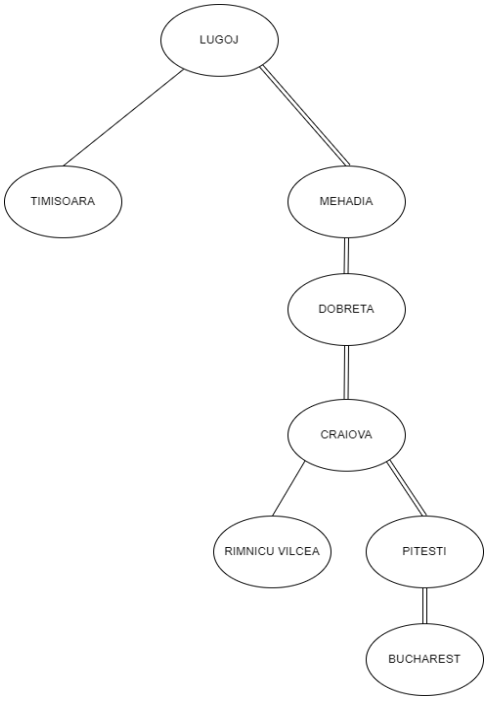
e) Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

→ No, agent function does not change if the machine speeds up. Only the machine will learn faster.

Que 3)

Trace the operation of A* search applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the f, g, and h score for each node.

→



Total Cost | Cost | Heuristic | path traveled
244 0 244 ['Lugoj']
311 70 241 ['Lugoj', 'Mehadia']
387 145 242 ['Lugoj', 'Mehadia', 'Drobeta']
425 265 160 ['Lugoj', 'Mehadia', 'Drobeta', 'Craiova']
440 111 329 ['Lugoj', 'Timisoara']
503 403 100 ['Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti']
504 504 0 ['Lugoj', 'Mehadia', 'Drobeta', 'Craiova', 'Pitesti', 'Bucharest']

Que 4)

Implement DFS, BFS, and A* for the problem of getting to Bucharest from Arad. You can use python or R for the implementation. Please present the path each algorithm will choose.

→ Uploaded on github
https://github.com/viper-vm/Artificial-Intelligence/tree/main/Assignment1

Que 5)

Give the name of the algorithm that results from each of the following special cases:

a) Local beam search with k = 1

→ Hill climbing algorithm

b) Local beam search with one initial state and no limit on the number of states retained.

→ It resembles Breadth-first search if it adds one complete layer of nodes before adding next layer.

c) Simulated annealing with T = 0 at all times (and omitting the termination test).

→ Hill climbing algorithm

d) Simulated annealing with T = ∞ at all times.

→ Depth-first search

e) Genetic algorithm with population size N = 1.

→ Depth-first search