CSC 520 Homework Assignment 1

Due: September 7, 2017
Unity ID: rsdates

1. **[20 points]** Suppose we have an agent that works to make an office building 'intelligent'. This means that the agent is responsible for automatically heating, cooling, and lighting areas of the building where there are people present. Sensors can tell the agent where people are currently. The agent must turn lights on when people enter a room and turn them off again within 2 minutes of the people leaving the room. The agent must keep the areas with people currently present at an appropriate temperature, say within a set four-degree range. If the room is too cold or too hot for more than 20 minutes, people will leave and the company will lose money. Heating and cooling rooms takes some time, depending on the size of the room. Therefore, the room may not be within the four-degree range when a person enters, but should get within the range before the 20-minute time limit. Answer the following questions about the agent.

   (a) Define a PEAS specification for the agent.

   Answer

PEAS Specification				
Agent Type	Performance Measure	Environment	Actuators	Sensors
Office Building 'intelligent'	Efficient heating, cooling(20 minutes), and lighting(off 2 minutes)	office building, employees	Timers, light switches, temperature adjustors	cameras, RFID, Sonar

   (b) Is it sufficient for the agent to be simple reflex? Why or why not?

   Answer

   Yes, I believe this agent could sufficiently be a simple reflexive agent, becasue the criteria to perform an action can be defined on an "if- then" bases. For example "if a person enters a room, then turn the lights on." There may be a challenge to be simple reflexive when dealing with the temperature adjustor depending on the performance of the heating/cooling system.

   (c) Would it be beneficial for the agent?s performance if it randomly heated or cooled rooms where there are no people currently? Identify possible disadvantages to this sort of random action.

   Answer

   No, I feel that randomly heating or cooling rooms where no one is present would not be beneficial. The disadvantages weigh more than the advantages of this particular case. This is because an "office building" environment is not enough context for size and therefore risk the case of wasting energy for rooms that people will never enter.

   (d) Suggest one improvement to the agent design. Since every improvement carries drawbacks, what are the drawbacks to yours?

   Answer

   An improvement to the agent design, could be that there is a predetermined schedule to heat or cool certain rooms. The drawback from this change would be that human error caused by lack of updating this schedule could result in the system inefficiently regulating temperature in unoccupied rooms.

2. **[10 points]** Answer the following questions based on the reading materials provided about IBM's Watson.

   (a) Describe a PEAS specification for Watson

   Answer

PEAS Specification				
Agent Type	Performance Measure	Environment	Actuators	Sensors
Watson	Win at Jeopardy (fast response, correct answers, correct bid without risk)	question that is being asked, other competitors,	synthesis of information retrieval, natural language processing, knowledge representation and reasoning, machine learning, computer-human interfaces	microphone, monitor, "ringer/buzzer"

(b) Describe Watson's environment. (Fully/partially observable, Deterministic/stochastic, etc.)

Answer

The current question that is asked along with the other competitors make up Watson's environment. It can be described also as being fully observable since Watson has the question as a text input. Watson also has a stochastic element, due to the fact that the answers of the competitors are unpredictable(random). This also plays into the fact that the environment is dynamic as well since the competitors may answer whenever.

(c) Discuss at least three separate aspects of the Jeopardy problem domain together with the hardware and/or software design choices in Watson that are rational given those problem aspects. 1. There are different types of questions that can be asked in Jeopardy. This made it necessary for Watson to be able to infer the question type through parsing, decomposition, and source acquisition etc technologies.

2. Competitors will answer the Jeopardy questions fairly fast in a matter of seconds. This required that Watson needed to be fast which is why having a paralleled computing system was a rational decision.

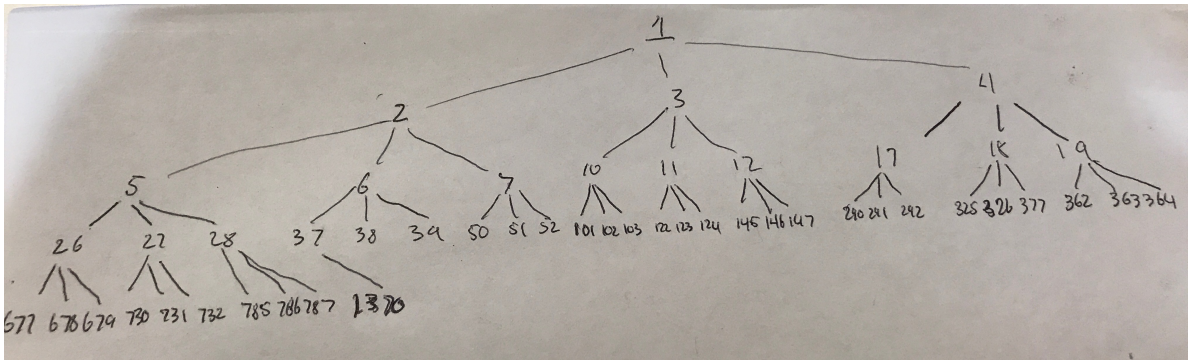
3. In Jeopardy participants are not able to research an answer. This is why having a large efficient local database was necessary.

(d) Describe the DeepQA approach developed for Jeopardy and name the six architectural roles that are designed in this model.

Answer

The DeepQA is a massively parallel probabilistic evidence-based architecture that includes over 100 different techniques for analyzing natural language, identifying sources, finding and generating hypotheses, finding and scoring evidence, and merging and ranking hypotheses. The 6 architectural roles that are designed in this model are the following: Content Acquisition, Question Analysis, Hypothesis Generation, Hypothesis and Evidence Scoring, Final Merging and Ranking

3. **[20 points]** Consider a state space where the start state is labeled 1 to each state k has three successors: labeled $(k^2) + 1, (k^2) + 2$, and $(k^2) + 3$ respectively. Draw the portion of the state space for states 1 to 1370.



(a) Suppose the goal state is 101. List the order in which states will be visited for the breath-first search.

Answer

1,2,3,4,26,27,28,37,38,39,50,51,52,101

(b) Suppose the goal state is 101. List the order in which states will be visited for the depth-limited search with limit 3.

Answer

if we consider the root node as level 0 then the answer is as follows:

1,2,5,26,27,28,6,37,38,39,7,51,52,3,10,101

However if we consider the root node as level 1 the search algorithm never finds the goal node

1,2,5,6,7,3,10

(c) Suppose the goal state is 101. List the order in which states will be visited for the iterative deepening search with initial cutoff 1 and cutoff increment 1.

Answer

1,

1,2,3,4

1,2,3,4,5,6,7,10,11,12,17,18,19

1,2,3,4,5,6,7,10,11,12,17,18,19,26,27,28,37,38,39,50,51,52,101

4. **[10 points]** Which agent structure is appropriate for each of the following agents, considering their task environment? Give a brief explanation for your answer. Room temperature controller

(a) Room temperature controller

Answer

I think this type of agent would best serve to have the simple reflexive agent structure. This is becasue most cases of a room temperature controller can be described in an "if-then" fashion. "if a person is present, then turn on the system"

(b) Taxi driver

Answer

I think this type of agent would best serve to have the goal-based agent structure. A taxi driver agent should be aware of the goal or end destination in order to determine the most efficient route to take.

(c) Tic-tac-toe player

Answer

I think this type of agent would best serve to have the learning agent structure. This is because in tic-tac-toe the agent will be against another competitor which implies a stochastic input from

(d) Face recognition system

Answer

I think this type of agent would best serve to have the model-based reflexive structure. This is becasue I view everyones face, as a model with states. This structure allows the agent to maintain or update the models states. for a face recognition system the system will be able to recognize the individuals face, but also improve the recognition as the individuals face changes (i.e ages).

5. **[40 points]** In a language of your choice (Java, Python, or C++), implement the Depth-First Search and Breadth-First Search algorithms. Your code should keep track of nodes expanded and should be able to compute the length of this list. Then run your algorithms on the Romanian road map. To save a bit of typing, you may use this file for the cities and roads: [roads.pl]. This is a Prolog source file, and this assignment does not use Prolog, so you will have to modify it for use with your code. Notice also that Assignment 1 does not use the road distances, or the longitude/latitude of the cities.

(a) [10 points] Consider the path from Fagaras to Dobreta and the path from Dobreta to Fagaras. Run your algorithms and show the paths returned by DFS and BFS results for each case. How do the solution paths compare for the two algorithms? Give an explanation for what you observe.

Answer

The following information shows how the paths compare to each other.

$Path : Fagaras - Dobreta$

DFS

The expansion count is 5

The path is ['fagaras', 'bucharest', 'giurgiu', 'pitesti', 'craiova', 'dobreta']

The length of the list is 6

BFS

The expansion count is 14

The path is ['fagaras', 'bucharest', 'sibiu', 'giurgiu', 'pitesti', 'urziceni', 'arad', 'oradea', 'rimnicu vilcea', 'craiova', 'hirsova', 'vaslui', 'timisoara', 'zerind', 'dobreta']

The length of the list is 15

$Path : Dobreta - Fagaras$

DFS

The expansion count is 4

The path is ['dobreta', 'craiova', 'pitesti', 'bucharest', 'fagaras']

The length of the list is 5

BFS

The expansion count is 9

The path is ['dobreta', 'craiova', 'mehadia', 'pitesti', 'rimnicu vilcea', 'lugoj', 'bucharest', 'sibiu', 'timisoara', 'fagaras']

The length of the list is 10

The path of Dobreta to Fagaras shows a quicker path for both the DFS and BFS algorithms. Also my program shows there are less expansions on that path as well. I feel like the fact that we aren't considering the edge length and also that the algorithm is based on alphabetical order in regards to expansion is a large dictator to the results.

$Executiontrace$

```
Roberts-MacBook-Pro-2:SearchRomania robertdates$ python SearchRomania.py bfs fagaras dobreta
The expansion count is 14
The path is ['fagaras', 'bucharest', 'sibiu', 'giurgiu', 'pitesti', 'urziceni', 'arad', 'oradea', 'rimnicu_vilcea', 'craiova', 'hirsova', 'vaslui', 'timisoara', 'zerind', 'dobreta']
The length of the list is 15
You have mail in /var/mail/robertdates
Roberts-MacBook-Pro-2:SearchRomania robertdates$ python SearchRomania.py dfs fagaras dobreta
The expansion count is 5
The path is ['fagaras', 'bucharest', 'giurgiu', 'pitesti', 'craiova', 'dobreta']
The length of the list is 6
Roberts-MacBook-Pro-2:SearchRomania robertdates$ python SearchRomania.py bfs dobreta fagaras
The expansion count is 9
The path is ['dobreta', 'craiova', 'mehadia', 'pitesti', 'rimnicu_vilcea', 'lugoj', 'bucharest', 'sibiu', 'timisoara', 'fagaras']
The length of the list is 10
Roberts-MacBook-Pro-2:SearchRomania robertdates$ python SearchRomania.py dfs dobreta fagaras
The expansion count is 4
The path is ['dobreta', 'craiova', 'pitesti', 'bucharest', 'fagaras']
The length of the list is 5
Roberts-MacBook-Pro-2:SearchRomania robertdates$
```

(b) [10 points] Is there a case where Depth-First performs worse than Breadth-First (in terms of number of cities visited in the path, not the distance)? If yes, what is the case? If not, explain why.

Answer

Yes, by looking at the path from zerind to oradea you can see this type of case happen. This is because the algorithm favors the alphabetical order when choosing how to expand. So for DFS it looks seeks out the left most path, even though the destination is right beside it, the algorithm is unaware that it only needs one hop to find goal. However the BFS algorithm expands and therefore considers both directions and so it finds the goal destination right away.

(c) [10 points] Is there a case where Breadth-First performs worse than Depth-First (in terms of number of cities visited in the path, not the distance)? If yes, what is the case? If not, explain why.

Answer

Yes, by looking at part a of this question both of the cases the breadth-first performs worse than the depth-first. To be specific the cases are Dobreta to Fagaras as one, and Fagaras to Dobreta as the other.

(d) [10 points] For the same graph, perform a hand-execution of Depth-First Iterative Deepening (DFID) with increment and cutoff initialized to 1, starting at Pitesti. List the nodes in the order expanded for the first five iterations of DFID, and the state of the data structure (stack) after each iteration. Expand the nodes alphabetically and insert them in nondecreasing alphabetical order. How does this list compare with the list of expansions in Breadth-First Search?

Answer

The picture shows a DFID expansion trees at each iteration focused on the inner loops (DFS) aspect of the algorithm. There is also shown the stack after each iteration of incrementing the cutoff. Notice that the stack is empty each iteration, becasue the stack will be empty due to the fact that the DFS part of the algorithm is completed. This is in contrast to a DFS list that would be much smaller. This list is partially due to fact that the DFID fully repeats the DFS process over each iteration, thus the list will contain more because of the repetition in the search.