CS3243 Tutorial 1

Eric Han

January 27, 2022

Introduction

Eric Han

4th Year PhD Student

- I cannot be there due to National Service commitments; I will save the rest of the introductions for the next week.
- Today's tutorial will be covered by the relief tutor.
- Feel free to telegram me `@Eric_Vader`/on our group, if you have any concerns.

Expectations

Expectations of you

1. Fill the seats from the front.
2. Come prepared to the Tutorial.
    2.1 You will be asked at random to discuss your answer.
3. Refrain from taking pictures of the slides.
    3.1 Learn to take good notes.
    3.2 The slides will *not* be distributed.
4. Try to be on time.

Any comments or suggestions for the lessons going forward?

Question 1 - Recap

- **Fully / Partially Observable**: Is the complete state of the environment accessible to the agent's sensors
- **Single / Multi-Agent**: Are there more than one actor in the environment? (competitive vs cooperative)
- **Deterministic / Stochastic**: Is the next state determined by the current state and action by the agent?
- **Episodic / Sequential**: Is the next episode dependent on the action taken previously?
- **Static / Dynamic**: Can the environment change while the agent is deliberating?
- **Discrete / Continuous**: Is the state of the environment discretized or varying continuously?

Easiest: Fully, Single, Deterministic, Episodic, Static

Question 1a

Determine the properties of the above problem from the perspective of an intelligent agent planning a solution. Complete the table below.

Question 1a - Answer

Environment Characteristic	Sudoku Puzzle Generation
Fully / Partially Observable	Fully Observable
Single / Multi-Agent	Single agent
Deterministic / Stochastic	Deterministic
Episodic / Sequential	Episodic / Sequential
Static / Dynamic	Static
Discrete / Continuous	Discrete

Explaination on Episodic / Sequential: Depending on the environment formulation, the next episode is dependent on the previous action.

Question 1b

Define the search space for the problem of generating a Sudoku Puzzle by completing the following.

- Give the representation of a state in this problem.
- Using the state representation defined above, specify the initial state and goal state(s).
- Define its actions.
- Using the state representation and actions defined above, specify the transition function T . (In other words, when each of the actions defined above is applied to a current state, what is the resulting state?).

Question 1b - Answer

Sample Environment Formulation:

- **State Space**: $A \in \{0, \cdots, 9\}^{9\times 9}$ where 0 is blank
- **Initial State**: A valid Sudoku Puzzle, completely filled without 0.
- **Final State**: A goal state is $T \in \{0, 1, \cdots\}$ steps away from the inital state.
- **Action**: Removing the value $a \in \{1, \cdots, 9\}$ at (i, j) in A
- **Transition Model**: $A - E_{i,j}(a)$, where $E_{i,j}(a)$ is zeros everywhere but a at (i, j)

Sample: There is no right answer, there are several correct representaions; as long as requirements are fulfilled.

Question 2a

Recap

AIMA Chapter 3, on graph search and tree-like search.

Question

Describe the difference between Tree Search and Graph Search algroithms.

Question 2a

Recap
AIMA Chapter 3, on graph search and tree-like search.

Question
Describe the difference between Tree Search and Graph Search algroithms.

Answer
Graph search will not explore redundant paths, only exploring:

1. unvisited states
2. visited states, but via less than optimal paths

Tree search will explore all paths, including redundant paths.

Question 2b

Recap

Revise DFS and BFS algorithms.

Question

i. Depth-First Search with tree-based implementation
ii. Depth-First Search with graph-based implementation
iii. Breadth-First Search with tree-based implementation
iv. Breadth-First Search with graph-based implementation

Question 2b

Recap
Revise DFS and BFS algorithms.

Question

i. Depth-First Search with tree-based implementation
ii. Depth-First Search with graph-based implementation
iii. Breadth-First Search with tree-based implementation
iv. Breadth-First Search with graph-based implementation

Answer

i. `S-C-E-D-B-E-D-D-A-F-G`
ii. `S-C-E-D-B-A-F-G`
iii. `S-B-C-A-D-E-E-F-D-D-G`
iv. `S-B-C-A-D-E-F-G`

Bonus Qn: Which implementation is typically used in practice?

Question 3

Recap

AIMA Chapter 3, on uniform search

Question

Prove that the **Uniform-Cost Search** algorithm is optimal as long as each action cost exceeds some small positive constant ϵ.

Question 3 - Answer

Given that cost $\ell(a, b) > \epsilon$ between 2 nodes a and b, Invariant hypothesis: For each visited node, $g(u)$ is the least cost from the source s to u.

Base case: When there is one visited node, its tivially true.

Assuming the hypothesis to be true, for all $u \in U$ visited nodes, $g(u)$ is with the least cost. For any V frontier nodes that is unvisited, we choose v where $\ell(u, v)$ is smallest; The cost to v is $g(v) = g(u) + \ell(u, v)$. Assume that $g(v)$ is not the least cost, then there will be a shorter path through some other node w. If w is visited then $g(u)$ must go through w, which contradicts. If w is unvisited, then $\ell(u, w) < \ell(u, v)$ which contradicts as w would have been chosen first.

Hence, if cost $\ell(a, b) > \epsilon$, $g(.)$ is the optimal cost to the source.

Question 4

Formulate the above as a search problem. More specifically, define the following:

- State representation
- Initial state
- Actions
- Transition model
- Step cost
- Goal test

Recap

Question 1b.

Question 4 - Answer

There is no right answer, but minimally:

- **State representation**: describes how pieces are
  - connected to the neighbours
  - represented
- **Initial state**: representation varies but must not have any inital connections.
- **Actions**: considers that legal connections, ie. you cannot simply take any two puzzle pieces.
- **Transition model**: Correctly mutates the current state that maintains semantics.
- **Step cost**: step cost is any positive, non-zero number.
- **Goal test**: checks must consider pieces with 2/3/4 sides are correctly connected.

Absolute positioning is acceptable, but the solution *must* consider orientation.

Question 5

Recap

Question 2b.

Question

Determine the final path found from the start (S) to the goal (G)?

Question 5

Recap
Question 2b.

Question
Determine the final path found from the start (S) to the goal (G)?

Answer
`S-B-F-H-D-G`