CS 38000 Artificial Intelligence **Amir Mohideen**

Assignment 01

Due on Thursday February 13, 2020 in class. No late assignment is acceptable.

Consider a problem of designing an intelligent agent called the maze-problem-solver.

Given ANY maze configuration, the maze has an entrance (equivalently, called the initial state) for entering the maze, and two exist ] [ (or called final states). The maze-problem-solver can find its ways from entrance to an exit ] [. The goal is that the maze-problem-solver enters the given maze and exit from the maze with minimal time span (in terms of number of squares traversed).

Given any maze configuration, for formulating the problem as a search problem, you require the following components for the maze-problem-solver. Describe the maze-problem solver as:

1. Identify the initial state that best represents the starting conditions, and the goal or condition the maze-problem solver wants to achieve.

Initial state? Position A

Goal state(s)? Reach position G or g

1. Formulate a state space over which the solver performs search. The state space is a way for representing in a computer the states and sequences of transitions (actions) of the real problem.

State Spaces?

(no of boxes travelled, array of all the letter checkpoint locations travelled, direction its travelling)

1. Formulate actions which caused the state space transitions that allow the solver to move between different states. The actions reflect the actions one can take in the real problem but operate on the state space instead.

Actions?

Top, Bottom, Left, Right, Reset

For designing a maze-problem-solver intelligent agent, which could act rationally, the rationality at any given time depends on the PEAS. Define the PEAS for this maze-problem-solver:

Performance measure?

Environment? Complete the following form.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Environment | Observable | Deterministic | Episodic | Static | Discrete | Agents |
| Unknown | Fully | deterministic | episodic | static | discrete | Single-agent |

Actuators: moving from one box to another, rotating and moving

Sensors: Wall sensor, letter checkpoint location identifier

**The following graph is a maze example: Assume A is the entrance. G and g are exits.**

**] [**

**] [**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **g** |
| **Y** |  |  |  |  |  |  |  |  | **b** |  |  |  |  |  |  |  |  | **f** |  |
|  |  |  |  |  |  |  |  |  |  | **Z** |  |  |  |  |  |  |  |  | **t** |
| **C** |  |  | **D** | **d** |  |  |  |  |  |  |  | **H** |  |  |  | **O** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | **e** |  |  |  | **a** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **B** |  |  |  |  |  |  |  | **E** |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **F** |  |  |  |  |  |  |  |  |  | **T** |  | **V** |
|  |  |  |  |  |  |  |  |  | **h** |  |  |  |  |  |  |  | **U** |  |  |
|  |  |  |  |  |  |  |  | **n** |  |  |  | **N** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **J** |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **R** |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | **M** |  |  |  | **Q** |  |  |  | **r** |  |  |  |
|  |  |  |  |  |  |  |  | **m** |  |  |  |  | **P** |  |  |  |  |  |  |
| **A** |  |  | **K** |  | **L** |  |  |  | **q** |  |  |  |  |  | **s** | **S** |  | **W** | **G** |

Animal, Attractive, Beautiful, Boy

**It should go in one direction until it faces a wall. When it does, it should change direction. If there is no wall, proceed. If not, change direction. Also, each time when changing direction, make sure u do not go back to the same step. If you do, record the steps taken and restart in a diff direction from a checkpoint. In this way record all steps. And proceed until you come across g or G**

**CS 380 As 01 01292020**