Faculty of Computer Science & Information Technology

2023-2027

Programming for Artificial Intelligence

Lab

Task 3

The Superior University

Submitted To:

Mr. Rasikh Ali

Submitted By:

Masooma Zahra

Roll No:

SU92-BSAIM-F23-088

Section:

4B

Department:

SE

Overview:

The **Water Jug Problem** is a classic AI search problem where two jugs of given capacities must measure a specific amount of water using allowed operations. This program uses **Depth-First Search (DFS)** to explore possible water transfers between jugs. If a solution is found, it prints the step-by-step process; otherwise, it reports failure.

Code Overview:

This code solves the **Water Jug Problem** using **Depth-First Search (DFS)**. It works by:

- Exploring different ways to **fill, empty, or transfer** water between jugs.
- Checking if the target amount is reached at each step.
- **Backtracking** when a dead-end state is encountered.
- Printing the **step-by-step** solution if a valid sequence is found.

Functions:

```
def isgoal(state, goal):
```

- Checks if either jug contains the **target amount** of water.

Parameters:

- `state`: A tuple representing the current water levels in both jugs.
- `goal`: The desired amount of water to be measured.

Returns:

- `True` if the goal is reached, otherwise `False`.

```
def getsuccessors(state, jug1cap, jug2cap):
```

- Generates all **possible next states** by:
  - **Filling** either jug.
  - **Emptying** either jug.
  - **Pouring** water from one jug to another.

Parameters:

- `state`: A tuple representing the current water levels in both jugs.
- `jug1cap`: Capacity of the **first jug**.
- `jug2cap`: Capacity of the **second jug**.

Returns:

- A **list of possible next states**.

```
def dfs(jug1cap, jug2cap, goal):
```

- Implements **Depth-First Search (DFS)** to find a sequence of steps that measure the target amount.
- Uses a **stack** to explore states and a **set** to track visited states.
- **Backtracks** if a state has already been visited.

Parameters:

- `jug1cap`: Capacity of the **first jug**.
- `jug2cap`: Capacity of the **second jug**.
- `goal`: The desired amount of water to be measured.

Returns:

- A **list of steps** if a solution exists, otherwise `None`.

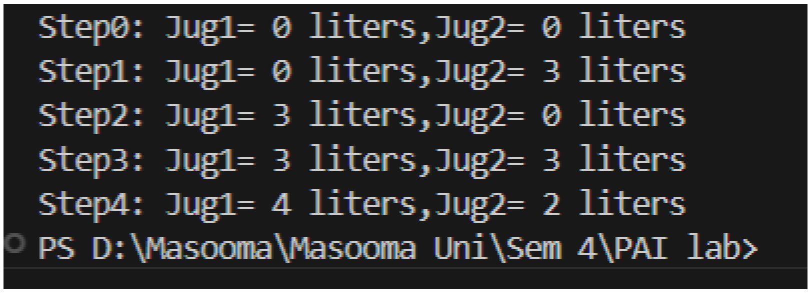
```
def printsol(solution):
```

- Prints the **step-by-step process** of reaching the goal.
- Displays the water levels in both jugs at each step.
- If no solution is found, prints `"No solution found"`.

Parameters:

- `solution`: A list of steps representing the solution sequence.

Output:



```
Step0: Jug1= 0 liters,Jug2= 0 liters
Step1: Jug1= 0 liters,Jug2= 3 liters
Step2: Jug1= 3 liters,Jug2= 0 liters
Step3: Jug1= 3 liters,Jug2= 3 liters
Step4: Jug1= 4 liters,Jug2= 2 liters
PS D:\Masooma\Masooma Uni\Sem 4\PAI lab>
```