Experiment No: 2

Write a Prolog program to implement Water-Jug Problem.

Date: _____

Competency and Practical Skills: Knowledge of logic programming, Basics of search Techniques

Relevant CO: CO1, CO5

Objectives: To create a functional and efficient Prolog program for solving logic puzzles.

Equipment/Instruments: Personal Computer, SWI-Prolog Interpreter

Theory:

The Water-Jug Problem is a classic problem in Artificial Intelligence that involves two jugs of different sizes and the task of measuring a specific amount of water using these jugs. The problem can be formulated as follows:
Given two jugs of sizes x and y, where x is greater than y, and an amount z of water to be measured, the task is to find a sequence of steps to measure exactly z liters of water using only these two jugs.

The problem can be solved using various AI algorithms such as Depth-First Search, Breadth-First Search, or heuristic search algorithms like A* search. The Breadth-First Search algorithm is commonly used to solve the Water-Jug Problem.

To solve the problem using Breadth-First Search, we start with the initial state where both jugs are empty and generate all possible next states from this state by applying the allowed actions such as filling a jug, emptying a jug, or pouring water from one jug to another. We then add these generated states to a queue and continue generating states and adding them to the queue until we find the desired goal state, which is when one of the jugs contains exactly z liters of water. Once the goal state is reached, we backtrack to find the sequence of steps taken to reach the goal state.

State Representation and Initial State – we will represent a state of the problem as a tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug.
Note $0 \leq x \leq 4$, and $0 \leq y \leq 3$.
Our initial state: (0,0)
Goal Predicate – state = (2,y) where $0 \leq y \leq 3$.

Operators –

- Fill the 4-gallon jug: This operator fills the 4-gallon jug to its maximum capacity. The resulting state will be (4, y), where y is the amount of water in the 3-gallon jug.
- Fill the 3-gallon jug: This operator fills the 3-gallon jug to its maximum capacity. The resulting state will be (x, 3), where x is the amount of water in the 4-gallon jug.
- Empty the 4-gallon jug: This operator empties the 4-gallon jug completely. The resulting state will be (0, y), where y is the amount of water in the 3-gallon jug.
- Empty the 3-gallon jug: This operator empties the 3-gallon jug completely. The resulting state will be (x, 0), where x is the amount of water in the 4-gallon jug.
- Pour water from the 4-gallon jug into the 3-gallon jug: This operator pours water from the 4-gallon jug into the 3-gallon jug until the 3-gallon jug is full or the 4-gallon jug is empty. The resulting state will be (x - (3 - y), 3) if the 4-gallon jug becomes empty or (4, y) if the 3-gallon jug becomes full.
- Pour water from the 3-gallon jug into the 4-gallon jug: This operator pours water from the

3-gallon jug into the 4-gallon jug until the 4-gallon jug is full or the 3-gallon jug is empty. The resulting state will be (4, y - (4 - x)) if the 3-gallon jug becomes empty or (x, 3) if the
4-gallon jug becomes
full.

Safety and necessary Precautions:

It is important to handle edge cases such as unsolvable puzzles, invalid puzzle states, and memory limitations to avoid program crashes or incorrect results.

Procedure:

1. Define the initial state using a Prolog predicate.
2. Define the goal state predicate.
3. Define the goal state predicate.
4. Define the operators
5. Define the search algorithm.

Observation/Program:

<Write Prolog code and analyse goal statements>

Conclusion:

Quiz: (Sufficient space to be provided for the answers)

1. What search algorithm can you use to find a solution to the Water-Jug Problem, and how would you implement it in Prolog?

2. How would you test the Prolog program to ensure that it works correctly?

Suggested Reference:

1. http://lpn.swi-prolog.org/lpnpage.php?pageid=online
2. "Artificial Intelligence" -By Elaine Rich And Kevin Knight (2nd Edition) Tata Mcgraw- Hill
3. "PROLOG Programming For Artificial Intelligence" -By Ivan Bratko(Addison-Wesley)

References used by the students: (Sufficient space to be provided)

Rubric wise marks obtained:

Rubrics	Knowledge (2)		Problem Recognition (2)		Logic Building (2)		Completeness and accuracy (2)		Ethics (2)		Total
	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	Good (2)	Avg. (1)	
Marks											