Mahavir Education Trust's

Shah & Anchor Kutchhi Engineering College,

Chembur, Mumbai 400 088

UG Program in Information Technology

Experiment No.: 2					
Date of Performance:					
Date of Submission:					
Program formation/ Execution/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer (03)**	**Experiment Marks (15)**	**Teacher Signature with date**

Name: Darsh Shah **Class:** BE-6 **Batch:** C **Roll no:** 56

EXPERIMENT 2

Aim: Design Problem definition and formulate it with state space method

Lab outcome No: 7.ITL.703.2

Lab Outcome: Analyse and formalize the problem as a state space graph, design heretic and select amongst different search or games-based techniques to solve them.

Theory:

STATE SPACE METHOD

State Space Method is a process used in the field of computer science, including artificial intelligence (AI), in which successive configurations or states of an instance are considered, with the intention of finding a goal state with a desired property.

Problems are often modelled as a state space, a set of states that a problem can be in. The set of states forms a graph where two states are connected if there is an operation that can be performed to transform the first state into the second.

State space search often differs from traditional computer science search methods because the state space is implicit: the typical state space graph is much too large to generate and store in memory. Instead, nodes are generated as they are explored, and typically discarded thereafter. A solution to a combinatorial search instance may consist of the goal state itself, or of a path from some initial state to the goal state.

Problem Statement:

Water jug problem:

Problem Definition: You are given two jugs, a 4-gallon one and a 3-gallon one. Neither has any measuring mark on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into the 4-gallon jug?

Solution:

The state space for this problem can be described as the set of ordered pairs of integers (x,y)

Where,

x represents the quantity of water in the 4-gallon jug, x= 0,1,2,3,4

y represents the quantity of water in 3-gallon jug, y=0,1,2,3

Start State: (0,0)

Goal State: (2,0)

Production Rules:

<table>
<tr><th>Rule</th><th>State</th><th>Process</th></tr>
<tr><td>1</td><td>(X,Y | X<4)</td><td>(4,Y)
{Fill 4-gallon jug}</td></tr>
<tr><td>2</td><td>(X,Y |Y<3)</td><td>(X,3)
{Fill 3-gallon jug}</td></tr>
<tr><td>3</td><td>(X,Y |X>0)</td><td>(0,Y)
{Empty 4-gallon jug}</td></tr>
<tr><td>4</td><td>(X,Y | Y>0)</td><td>(X,0)
{Empty 3-gallon jug}</td></tr>
<tr><td>5</td><td>(X,Y | X+Y>=4 ^ Y>0)</td><td>(4,Y-(4-X))
{Pour water from 3-gallon jug into 4-gallon jug until 4-gallon jug is full}</td></tr>
<tr><td>6</td><td>(X,Y | X+Y>=3 ^X>0)</td><td>(X-(3-Y),3)
{Pour water from 4-gallon jug into 3-gallon jug until 3-gallon jug is full}</td></tr>
<tr><td>7</td><td>(X,Y | X+Y<=4 ^Y>0)</td><td>(X+Y,0)
{Pour all water from 3-gallon jug into 4-gallon jug}</td></tr>
<tr><td>8</td><td>(X,Y | X+Y <=3^ X>0)</td><td>(0,X+Y)
{Pour all water from 4-gallon jug into 3-gallon jug}</td></tr>
<tr><td>9</td><td>(0,2)</td><td>(2,0)
{Pour 2 gallon water from 3 gallon jug into 4 gallon jug}</td></tr>
</table>

Initialization:

Start State: (0,0)

Apply Rule 2:

(X,Y | Y<3) -> (X,3)

{Fill 3-gallon jug}

Now the state is (X,3)

Iteration 1:

Current State: (X,3)

Apply Rule 7:

(X,Y | X+Y<=4 ^Y>0)   (X+Y,0)

{Pour all water from 3-gallon jug into 4-gallon jug}

Now the state is (3,0)

Iteration 2:

Current State : (3,0)

Apply Rule 2:

(X,Y | Y<3)   ->   (3,3)

{Fill 3-gallon jug}

Now the state is (3,3)

Iteration 3:

Current State:(3,3)

Apply Rule 5:

(X,Y | X+Y>=4 ^ Y>0)   (4,Y-(4-X))

{Pour water from 3-gallon jug into 4-gallon jug until 4-gallon jug is full}

Now the state is (4,2)

Iteration 4:

Current State : (4,2)

Apply Rule 3:

(X,Y | X>0)   (0,Y)

{Empty 4-gallon jug}

Now state is (0,2)

Iteration 5:

Current State : (0,2)

Apply Rule 9:

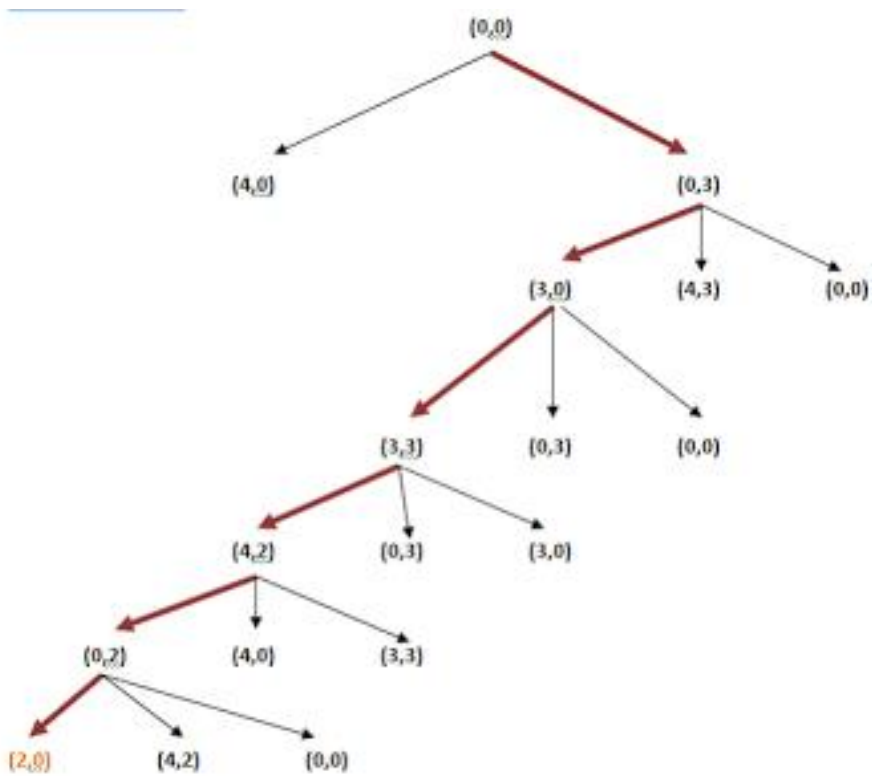
(0,2)   (2,0)

{Pour 2 gallon water from 3 gallon jug into 4 gallon jug}

Now the state is (2,0)

GOAL ACHIEVED

State Space Tree:



Missionaries and Cannibal Problem:

Problem Definition: Three missionaries and three cannibals wish to cross the river. They have a small boat that will carry up to two people. Everyone can navigate the boat. If at any time the Cannibals outnumber the Missionaries on either bank of the river, they will eat the Missionaries. Find the smallest number of crossings that will allow everyone to cross the river safely.

Goal: Move all the missionaries and cannibals across the river.

Constraint: Missionaries can never be outnumbered by cannibals on either side of river or else missionaries will be killed.

State: Configuration of missionaries and cannibals and boat on each side of river

Initial State: 3 missionaries and 3 cannibals and the boat are on the near side of the bank

Operators: Move containing some set of occupants across the river to the other side.

Solution:

B represents the position of the boat on the near and far side of the river

M represents the Missionaries

C represents the Cannibals

Step no		Near side	Far side
0	Initial Setup:	MMMCCC B	-
1	Tow cannibals cross over:	MMMC	B CC
2	One comes back:	MMMCC B	C
3	Tow cannibals go over again:	MMM	B CCC
4	One comes back:	MMMC B	CC
5	Two missionaries cross:	MC	B MMCC
6	A missionary and a cannibal return:	MMCC B	MC
7	Two missionaries cross again:	CC	B MMMC
8	A cannibal Returns:	CCC B	MMM
9	Two cannibals cross:	C	B MMMCC
10	One returns:	CC B	MMMC
11	And brings over the third:	-	B MMMCCC

GOAL ACHIEVED

Conclusion: Hence we successfully analysed and formalized the problems as a state space graph, designed heretic and selected amongst different search or games-based techniques to solve them.