Date.............

General Problem Solving

* Production System : Production System is one of the formalisms that helps AI Programs to do search process more conveniently in State - Space Problems
  two states
  (1) Start (initial) State
  (2) Goal (final) State
* Production System Consist no of Production Roles
  (1) Left Side :-> applicability of role (current state)
  (2) Right Side :-> action to be performed if role is applied (new state)
  (3) with database, where new input can be added which changes behaviour of the System

(1) Water Jug Problem :-

* Problem Statement : We have two jugs, a 5-gallon (5-g) and the other 3-gallon (3-g) with no measuring marker on them. There is endless supply of water through tap. Our task is to get 4 gallon of water in the 5-g Jug.

  1 gallon = ~~4~~ 3.785 Liters (we have considered 5 liters)

* Solution :

(1) Start State (0,0)
(2) Goal State (4, N) for any Value of N ≤ 3

Vidya 254

Date..................

* Possible Operations on water jug problem

(1) Fill 5-g jug from the tap and empty the 5-g jug by throwing water down the drain

(2) Fill 3-g jug from the tap and empty the 3-g jug by throwing water down the drain

(3) Pour some or 3-g water from 5-g jug into 3-g jug to make it full

(4) Pour some or full 3-g jug water into the 5-g jug.

* Production Rules for Water Jug Problem

Rule No	Left of rule	Right of rule	Description
1	$(x, y \mid x < 5)$	$(5, y)$	Fill 5-g jug
2	$(x, y \mid x > 0)$	$(0, y)$	Empty 5-g jug
3.	$(x, y \mid y < 3)$	$(x, 3)$	Fill 3-g jug
4	$(x, y \mid y > 0)$	$(x, 0)$	Empty 3-g jug
5	$(x, y \mid x + y \leq 5 \wedge y > 0)$	$(x + y, 0)$	Empty 3-g into 5-g jug
6	$(x, y \mid x + y \leq 3 \wedge x > 0)$	$(0, x + y)$	Empty 5-g into 3-g jug
7	$(x, y \mid x + y \geq 5 \wedge y > 0)$	$(5, y - (5 - x))$ until 5-g is full	Pour water from 3-g into 5-g jug
8	$(x, y \mid x + y \geq 3 \wedge x > 0)$	$(x - (3 - y), 3)$	Pour water from 5-g jug into 3-g jug until 3-g jug is full

* Solution Path

Rule applied	5-g jug	3-g jug	Step No
Start State	0	0	–
1	5	0	1
8	2	3	2
4	2	0	3
6	0	2	4
1	5	2	5
8	4	3	6
Goal State	4	–	

For Genarating new State from Current state rules are applied, In case the state is not goal State, Procedure is repeated until goal State

Date..................

(2) Missionaries and Cannibals Problem

Problem Statement : Three missionaries and three cannibals want to cross a river. There is a boat on their side of the river that can be used by either one or two persons. How should they use this boat to cross the river in such a way that cannibals never outnumber missionaries on either side the river?
If the cannibals ever outnumber the missinaries (on either bank) then the missionaries will be eaten. How can they all cross over without anyone being eaten?

Solution : State space for this problem can be described as the set of orders pairs of left and right banks of the river as (L, R) where each bank is represented [n m, m C, B].

$n \rightarrow$ no of missonaries (3)
$m \rightarrow$ no of cannibals (3)
$B \rightarrow$ Boat (1)

(1) Start state : ([3m, 3C, 1B], [0m, 0C, 0B])
(2) Any state : ($[n_1 M, m_1 C, \_]$, $[n_2 M, m_2 C, \_]$)

- $n_1 (\neq 0) \geq m_1$; $n_2 (\neq 0) \geq m_2$; $n_1 + n_2 = 3$
  ; $m_1 + m_2 = 3$
- $[[n_1 M, m_1 C, \underbrace{\_]}$ $\rightarrow$ means boat may be present or absent

Date......................

Solution path

Rule number	([3m,3C,1B],[0m,0C,0B)] ← Start
L2 :	([2m,2C,0B],[1m,1C,1B])
R4 :	([3m,2C,1B],[0m,1C,0B])
L3 :	([3m,0C,0B],[0m,3C,1B])
R4 :	([3m,1C,1B],[0m,2C,0B])
L1 :	([1m,1C,0B],[2m,2C,1B])
R2 :	([2m,2C,1B],[1m,1C,0B])
L1 :	([0m,2C,0B],[3m,1C,1B])
R5 :	([0m,3C,1B],[3m,0C,0B])
L3 :	([0M,1C,0B],[3m,2C,1B])
R5 :	([0m,2C,1B],[3m,1C,0B])
L3 :	([0m,0C,0B],[3m,3C,1B])

↑ Goal State