Artificial Intelligence and Expert System

MCA IV Semester

Problem Solving

- Is a method of deriving solution steps beginning from initial description to desired solution.
- Problem is solved by series of action that minimizes the gap between the initial and final state
- In AI, problems are modeled as *state space representation* where state space is set as all possible states from start to goal state.
- Set of states form a *graph or tree* where two states are linked is there is operation that transforms the given state to the new state
- While solving the problem state space is generated in process of searching the solution

Problem Solving

- Typical state space is too large to be stored in the memory
- Two types of problem solving are
  - general –purpose method : generate the solution and test it
  - Special-purpose method : tailored for specific type of problems
- For generating a new state an *action/operator/rule* is applied
- If the state is not the goal state procedure is repeated
- Order of applications of rule to current state is called *control strategy*

Problems , Problem Spaces and Search

- To build a system to solve a particular problem , we need four things
  - **Define the problem precisely**: must include definition of initial and final situation-*state space representation*
  - **Analyze the problem** : Few parameters can have immense impact on appropriateness of various possible techniques.
  - **Isolate and represent the task knowledge**
  - **Choose the best** problem solving technique and apply it.

State space Representation

- Informal description of a problem is presented in a formal way.
- Given situation is converted into desirable one by implementing sequence of operators
- Solution to a problem can be defined by the moving in *state space*
- Each state correspond to legal position on the board; values of a variables or matrix describing the problem status.
- Change in state takes place by applying an operator in current state and generating the new state
- These movement trigger the solution

Advantages of state space representation

- These representations are useful for less structured problems
- Advantages
  - It allows formal definition of the problem
  - It permits us to define the solution as a combination of knowledge techniques and search strategy
- Examples
  - Water Jug problem
  - Missionary Cannibals Problem
  - Tic-Tac-Toe

Formal description of the problem

- Define a state space containing all possible configuration
- Specify one or more state as *Initial State*
- Specify one or more state as *goal state*
- Specify a set of rules that describe actions(operators) available.

- If formal description of problem is done from an informal description automatically – it is called *operationalisation*

Defining problem as state space

- Problem can be solved **using rules** in combination with proper control strategy
- Control strategy will cause **movement** in problem space till path from initial to goal state is found
- Thus process of **search** is fundamental to problem solving
- Search forms the core of Intelligent system
- If we do not have a direct solution, search methods can be applied
- **Production system** is one of the formalism that helps AI programs to do search process more conveniently in state-space problem

State Space Representation(SSR)

State Space representation

- Water Jug Problem
- Missionary Cannibles
- Tic-Tac-Toe

Water Jug problem-SSR1

Informal description of problem

Given two jugs a 4 gallon and a 3 gallon with no measuring marks on it.Pump is used to fill jug with water. How to get exactly 2 gallon of water in 4 gallon jug?

- *Formal Description* State space representation
- Data representation:
  - any state is described by 2 variables (x,y)
  - x= 0,1,2,3,4 y=0,1,2,3
  - x=number of gallons in 4 gallon jug
  - y=number of gallon in 3 gallon jug
- Initial state (0,0)
- Final State=(2,n)

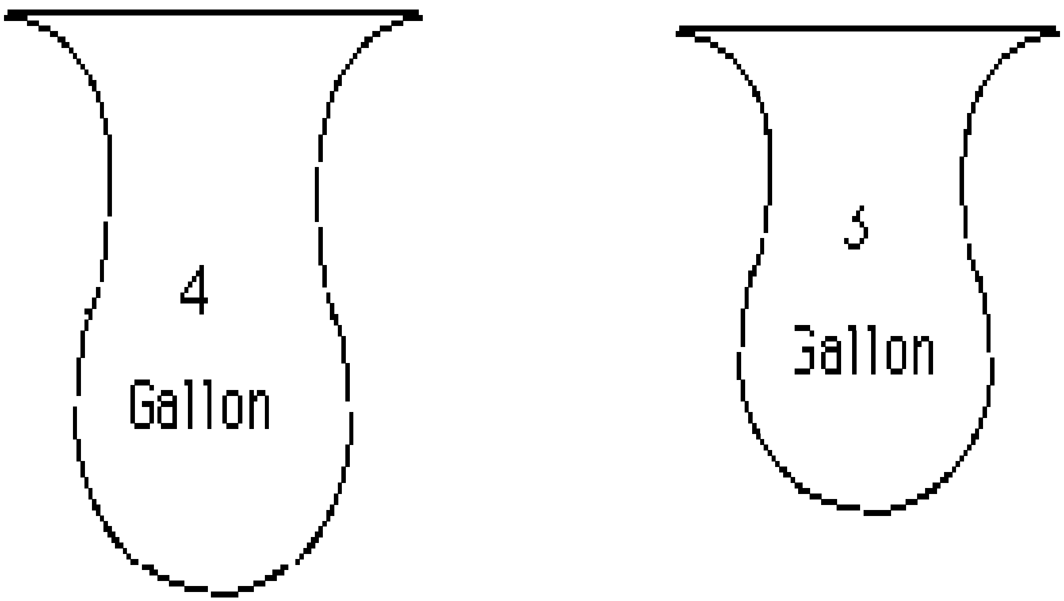
Water Jug problem

- Operators(in form of rules) are as follows
  1. Fill the 4 gallon jug if $(x<4) \rightarrow (4,y)$
  2. Fill the 3 gallon jug if $y<3\} \rightarrow (x,3)$
  3. Pour some water(d) out of 4 gallon jug if $(x>0) \rightarrow (x-d,y)$
  4. Pour some water out of 3gallon jug if $(y>0) \rightarrow (x,y-d)$
  5. Empty 4 gallon jug on ground if $(x>0) \rightarrow (0,y)$
  6. Empty 3 gallon jug on ground if $(y>0) \rightarrow (x,0)$
  7. Pour water from 3G into 4G till 4G jug is full

     if $(x+y >=4$ and $y>0) \rightarrow (4,y-(4-x))$
  8. Pour water from 4G into 3G till 3G jug is full

     if $(x+y >=3$ and $x>0) \rightarrow (x-(3-y),3)$
  9. Pour all water from 3G jug into 4G jug if $(x+y <4$ and $y>0) \rightarrow (x+y,0)$
  10. Pour all water from 4G jug into 3G jug if $(x+y <3$ and $x>0) \rightarrow (0,x+y)$
  11. Pour 2G water from 3G jug to 4G jug $(0,2) \rightarrow (2,0)$

Water Jug Problem



x = 0,1,2,3,4

y = 0,1,2,3

State Representation - S(x y)

x,y =
integer gallons of water in 4 and 3 gal. containers, respectively.

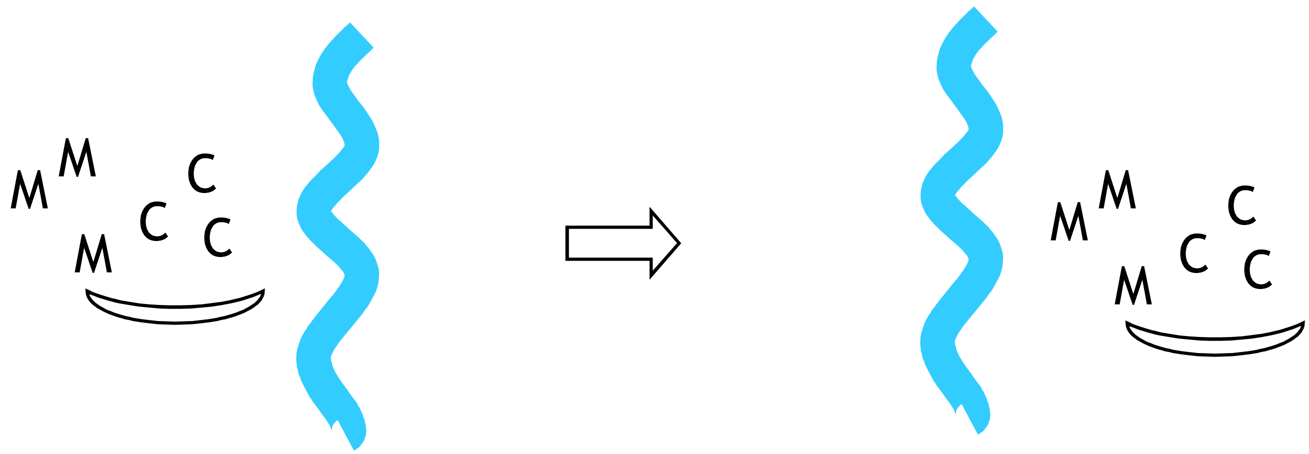
Start State: (S 0 0)

Goal State: (S 2 y)

Solution

Rule /operator applied	Change in state	
	x	Y
2	0	0 Initial state
9	0	3
2	3	0
7	3	3
5	4	2
9	0	2
	2	0 goal state

Missionary Cannibals



Initial state Goal state

Missionary Cannibals problem

- *Informal Representation*
- 3 missionaries and 3 cannibals on left side of river. One boat can carry 2 entities at a time .Number of cannibals should not exceed number of missionaries. All should reach right side the river
- *Formal Representation-SSR1*
- *Data representation*
  - (#m,#c,1/0)
  - #m= number missionary on the left side of the river
  - #c = number of cannibals on left side of the river
  - Boat on left = 0 boat on right = 1

Missionary Cannibals problem

- *Initial State*
  - (3,3,1) three missionaries and three cannibals are on right side of the river
- *Goal State*
  - (0,0,0)
  - None of the missionary and cannibals are on the left side of the river and all are shifted to right side successfully
- *Operators*
  - How these entities cross the river?
  - Number of missionaries and cannibals on boat represent the operations of the problem

Missionary Cannibals problem

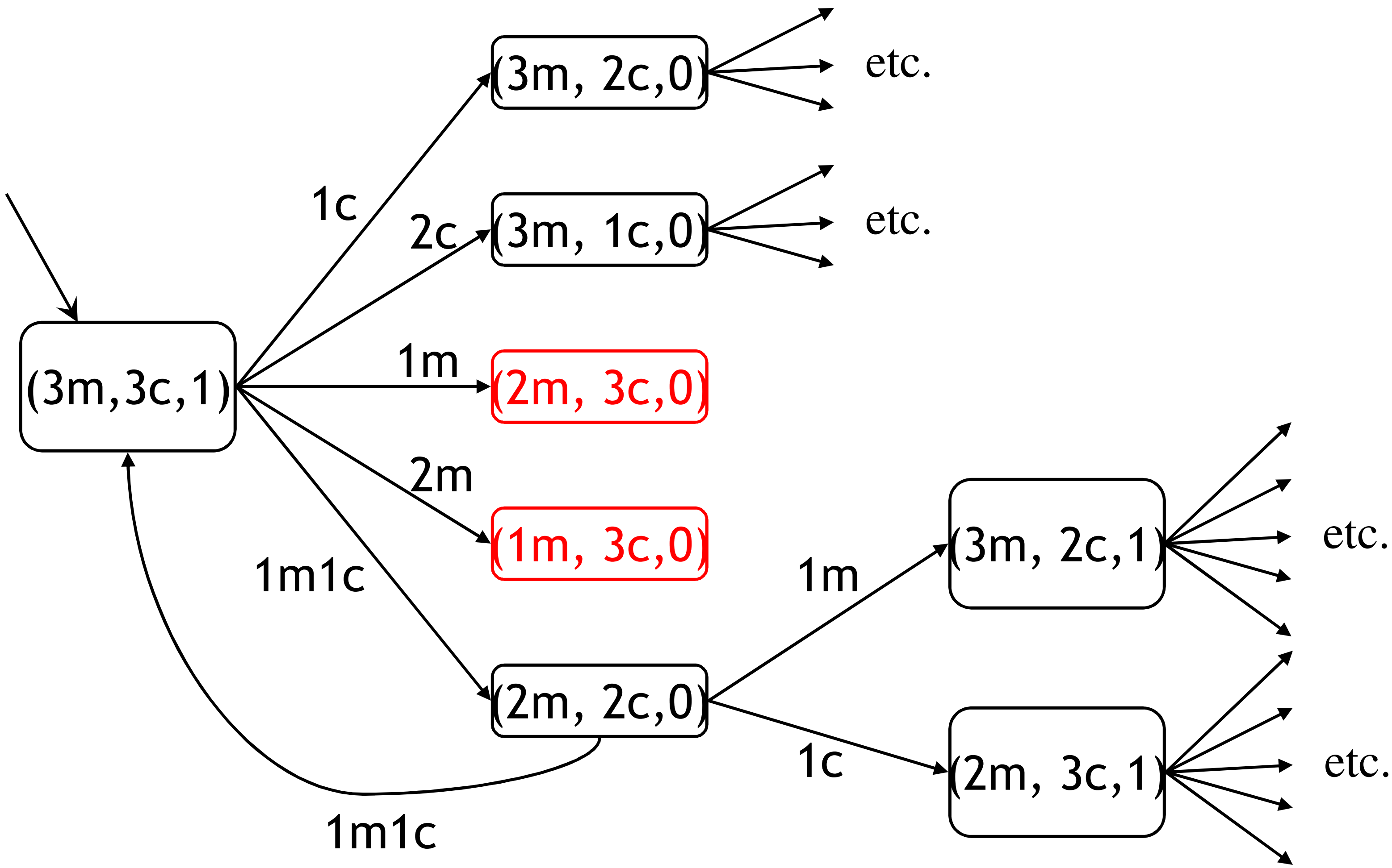
- (x,y) where x+y <=2 and x is number of missionary on boat and y is number of cannibals on boat'
  - So allowed values on boat is
  - (2,0) : boat can carry two missionary and 0 cannibals
  - (0,2): boat can carry 0 missionary and 2 cannibals
  - (1,1): boat can carry 1 missionary and 1 cannibals
  - (1,0): boat can carry 1 missionary and 0 cannibals
  - (0,1): boat can carry 0 missionary and 1 cannibals

Missionary cannibals

One of the solution as sequence of operators in state space

State (M,C,boat) on right	operator
(3m,3c,1)	2c from Left to right
(3m,1c,0)	1c from right to left
(3m,2c,1)	2c from left to right
(3m,0c,0)	1c from right to left
(3m,1c,1)	2m from left to right
(1m,1c,0)	1m,1c from right to left
(2m,2c,1)	2m from left to right
(0m,2c,0)	1c from right to left
(0m,3c,1)	2c from left to right
0m,1c,0)	1c from right to left
0m,2c,1)	2c from left to right
(0m,0c,0)	

The state space



Missionary Cannibals SSR2

-
- Any state captures information on both the sides of river bank
  - ([#M,#C,1/0] [#M,#C,1/0])
- Represents number of M and C on both sides of river
- Start State
  - ([3M,3C,1],[0M,0C,0])
- Goal State
  - ([0M,0C,0] [3M,3C,1])
- Operators
  - (x,y) where x+y <=2 and x is number of missionary on boat and y is number of cannibals on boat'

Tic-Tac-Toe

X	O	

e(p) = 6 - 5 = 1

- **Initial State**: Board position of 3x3 matrix with 0 and X.
- **Goal State**: Any row or column or diagonal with all O's or X's
- **Operators:** Putting 0's or X's in vacant positions alternatively
- **Terminal test:** Which determines game is over
- **Heuristic function**:

  e(p) = (No. of complete rows, columns or diagonals are still open for player) – (No. of complete rows, columns or diagonals are still open for opponent)

Tic-Tac-Toe