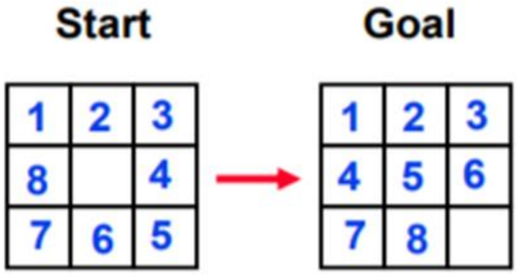
Problems, Problem Spaces And Search

Problem Solving in AI

- To build a system/agent to solve a particular problem in AI, we need to do following four things:

1) Define the problem precisely. This definition must include precise specification of what the initial state (s) will be as well as what final situations constitute acceptable solutions to the problem.



Problem Solving in AI

2. Analyze the problem i.e. aspects that have important impact on the appropriateness of various possible techniques for solving the problem.

3. Isolate and represent the task knowledge to solve the problem.

4. Choose the best problem-solving technique(s) and apply it (them) to get a solution.

Problem Formulation in AI

Problem can be defined using five components:

- Initial state
- Total Possible Actions: A description of possible actions available to the agent.
- State Space: Set of all states reachable from the initial state by any sequence of actions
- Path Cost: Cost of each path
- Goal state

Defining the problem as state space

- State spaces (means all possible states) are used extensively in solving Artificial Intelligence Problems.

- A **state space** consists of
  - **A (possibly infinite) set of states**
    - The **start state** represents the initial problem
    - Each **intermediate state** represents some configuration reachable from the start state
    - Some states may be **goal states** (solutions)

Defining the problem as state space- Contd…

- **A set of rules/operators/actions**
- Applying an rule to a state transforms it to another state in the state space
  - Not all rules are applicable to all states.
- **Path Cost**
  - Cost incurred in reaching from one state to another.

Example1: Water Jug Problem

Problem:

- We have given two jugs with different capacities, one is an X-liter jug and the other is a Y-liter jug.
- Neither has any marking on it.
- There is a pump that can be used to fill the jugs.
- How can we get exactly L liters of water in a particular jug?

For example:

Suppose we have two jugs with 4L and 3L capacity with no markings. How can we get exactly 2L water in 4L jug.

State Space Representation of Water Jug Problem

- Each state of the problem as a tuple (x, y) where x represents the amount of water in the first-jug and y represents the amount of water in the second-jug. y).
- **Initial State:** Usually (0,0)
- **Intermediate State:** (x , y) is obtained after applying possible rule on the previous state. Note $0 \leq x \leq max(x\_capacity)$, and $0 \leq y \leq max(y\_capacity)$
- **Final State:** L liters water in any one jug i.e. (L,y) or (x , L).
- **Rules / Operators:** Rules for the problem can be defined as:
  - water can be filled in any jug from pump,
  - we can pour water out of the jug onto the ground,
  - water can be poured from one jug to another.

State Space Representation of Water Jug Problem Contd…..

Firstly we need to define a state space:

Define an ordered pair (x, y) such that

x=0,1,2,3,4 and

y = 0,1,2

State Space Representation of Water Jug Problem Contd.....

- For example, for the 4L and 3L jugs with initial state (0,0) and goal state (2,y) following rules can be defined:

Rule	Description
1.	$(x, y) \rightarrow (4, y)$ If $x < 4$, we can fill 4-L jug completely.
2.	$(x, y) \rightarrow (x, 3)$ If $y < 3$, we can fill 3-L jug completely.
3.	$(x, y) \rightarrow (0, y)$ If $x > 0$ we can empty 4-L jug on ground
4.	$(x, y) \rightarrow (x, 0)$ If $y > 0$, Empty 3-L jug on ground
5.	$(x, y) \rightarrow (4, y - (4 - x))$ $0 < x+y \geq 4$ and $y > 0$ Pour some water from 3-L jug to fill 4-L jug
6.	$(x, y) \rightarrow (x - (3-y), 3)$ $0 < x+y \geq 3$ and $x > 0$; Pour some water from 4-L jug to fill 3-L jug

State Space Representation of Water Jug Problem Contd…..

Rule	Description
7.	(x, y) → (x+y, 0) 0 < x+y < =4 and y ≥ 0 ; Pour all of water from 3-L jug into 4-L jug
8.	(x, y) → (0, x+y) 0 < x+y <= 3 and x ≥ 0 ; Pour all of water from 4-L jug into 3-L jug
9.	(x, y) → (x-d, y) If x >0, pour some water out of 4-L jug completely.
10.	(x, y) → (x, y-d) If y >0, pour some water out of 4-L jug completely.

Example

Jug X	Jug Y	Rule No.
0	0	
0	3	2
3	0	7
3	3	2
4	2	5
0	2	3
2	0	7

Note: There may be many solutions and we try to search the best solution

8-Puzzle Problem

- The 8-Puzzle problem includes a 3x3 square tray in which 8 numbered tiles are placed and the ninth one is unnumbered (blank/ black).
- This problem involves moving the tiles on the tray into a particular configuration.
- The blank square on the board represents a space.
- The player can move a tile into the space, freeing that position for another tile to be moved into and so on.
- **For Example,**

Initial State

1	2	3
	4	6
7	5	8

Goal State:

1	2	3
4	5	6
7		8

State Space Representation of 8- Puzzle Problem

- Each state of 8-puzzle problem is represented as a 3X3 matrix with one of the tile blank and remaining 8 tiles numbered.
- **Initial State:** Any random arrangement of 8 numbered tiles and one blank tiles given in the problem.
- **Intermediate States:** Any random arrangement of 8 numbered tiles and one blank tiles obtained from after applying valid operators or rules on the current state.
- **Final State:** Any random arrangement of 8 numbered tiles and one blank tiles given in the problem.

State Space Representation of 8- Puzzle Problem

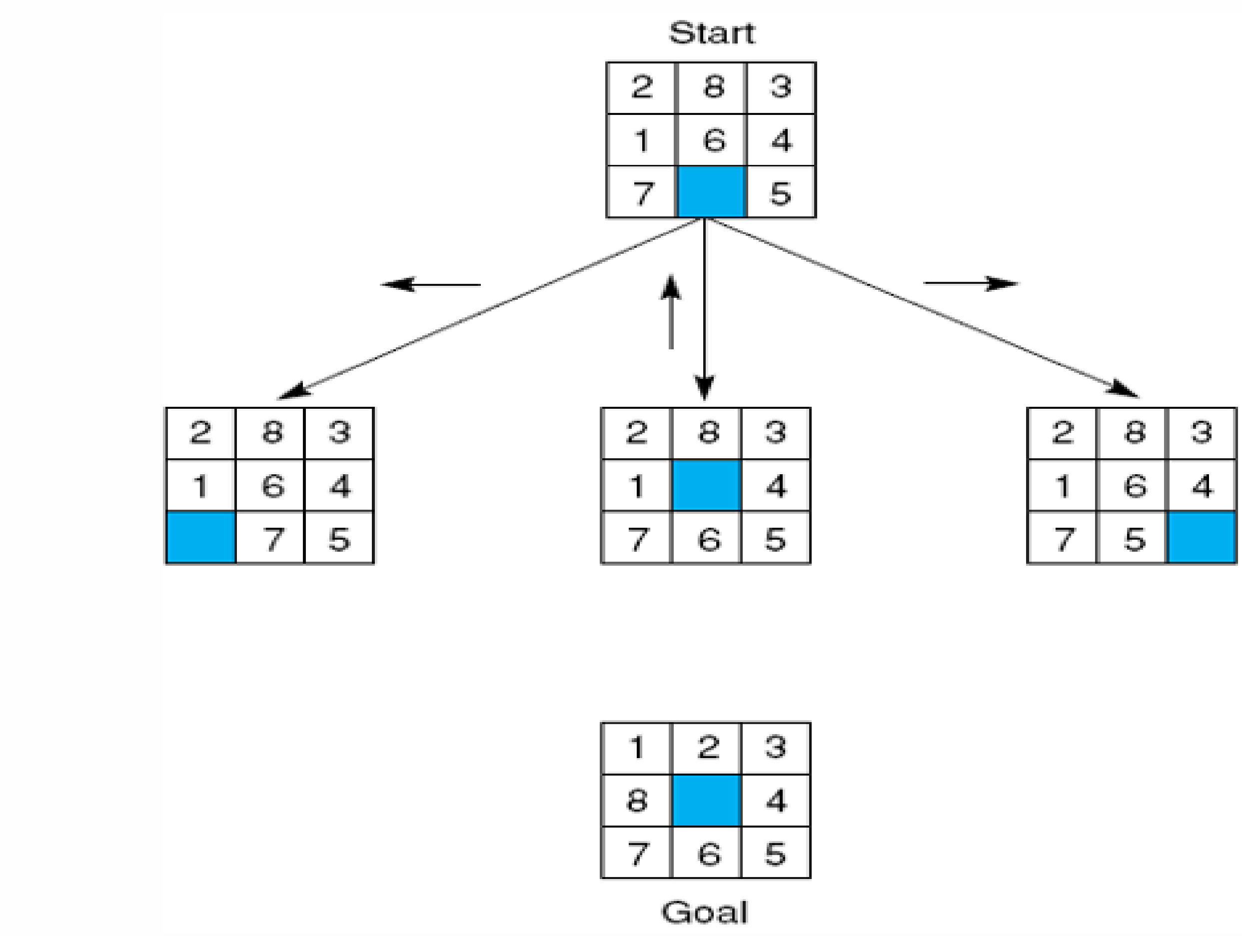
- **Operators / Rules:**

The empty space can only move in four directions (Movement of empty space)

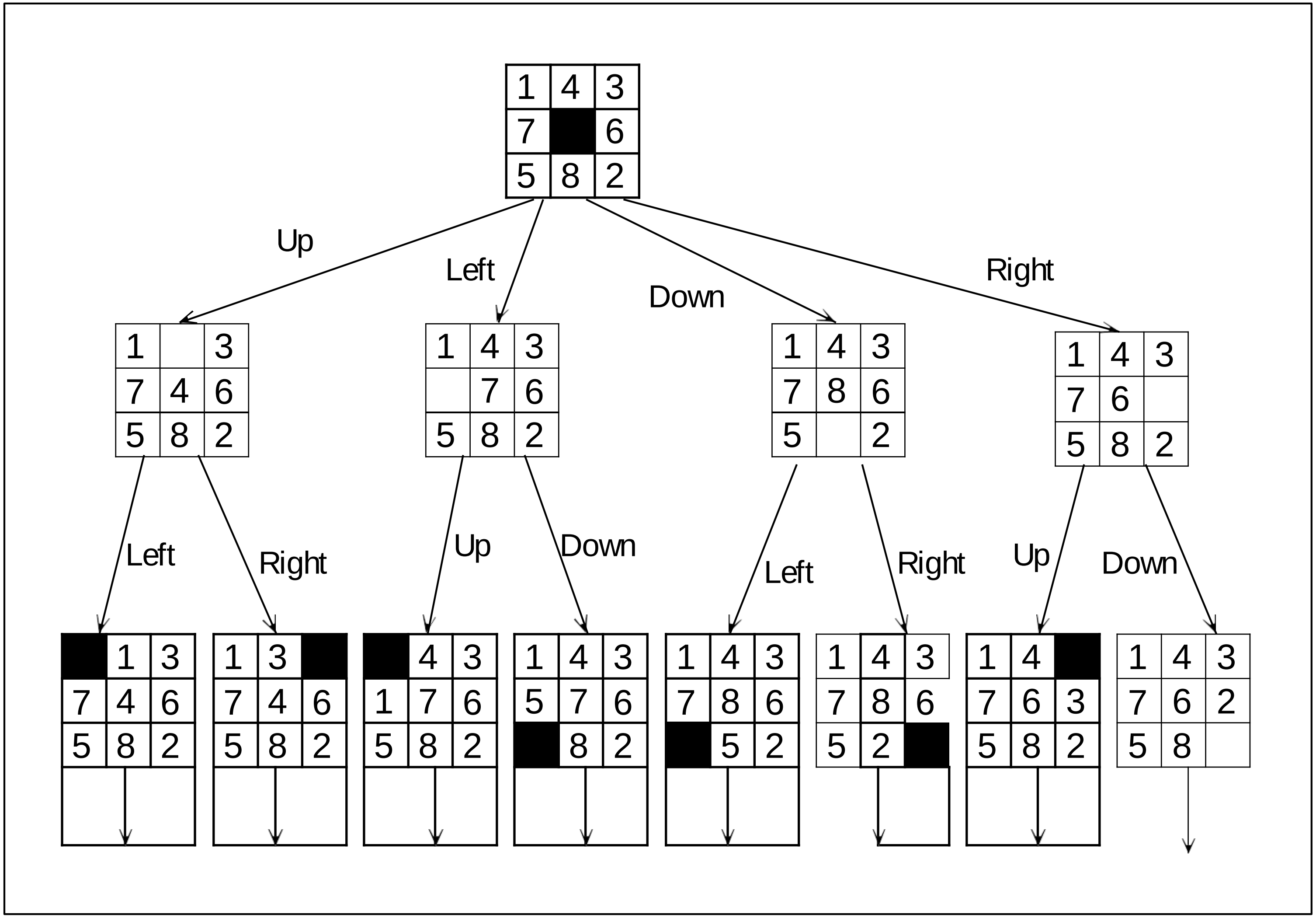
i. Up
ii. Down
iii. Right or
iv. Left

The empty space cannot move diagonally and can take only one step at a time.

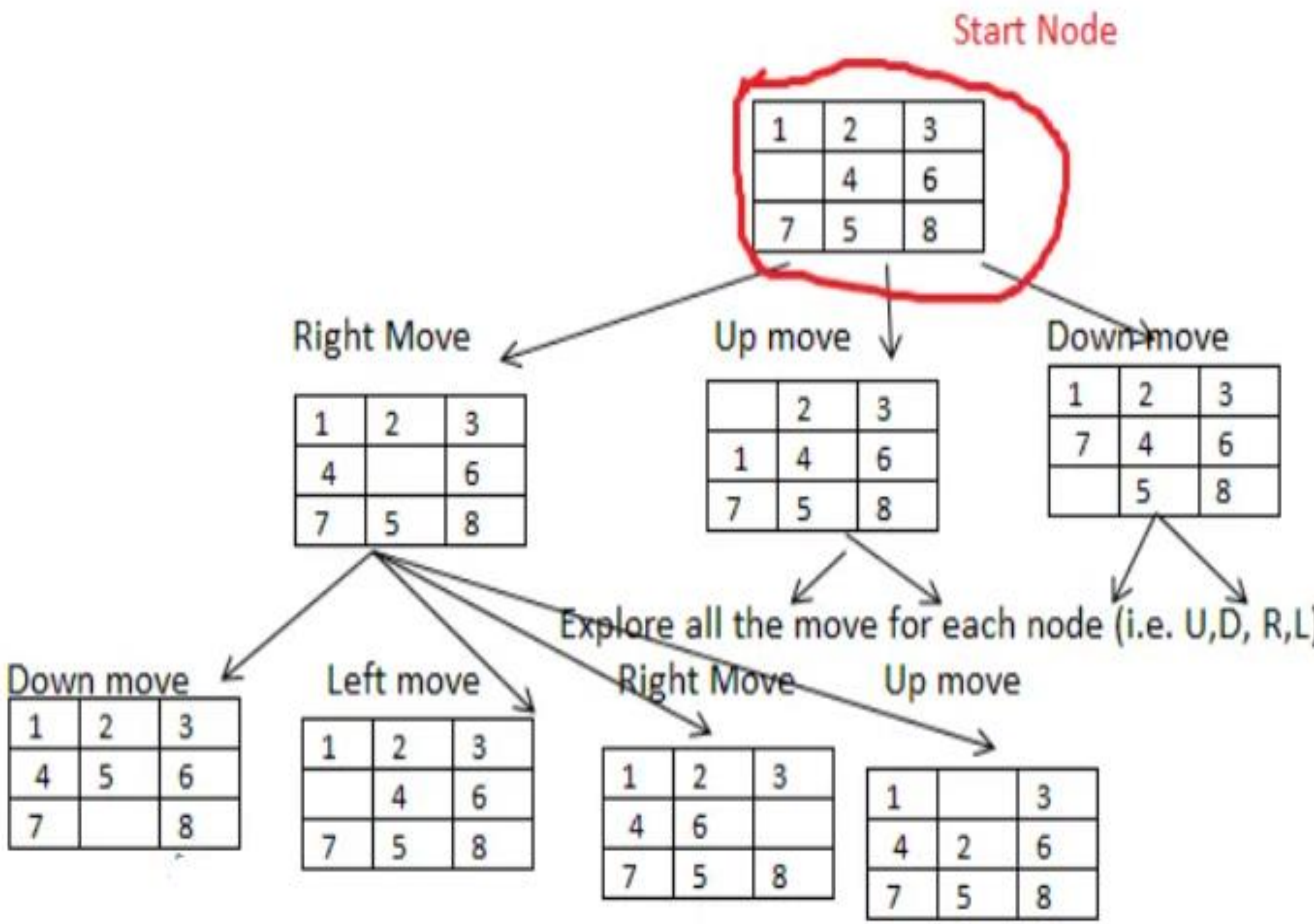
State Space Representation of 8- Puzzle Problem



State Space Representation of 8-Puzzle Problem



State Space Representation of 8- Puzzle Problem



All Possible moves of empty tile

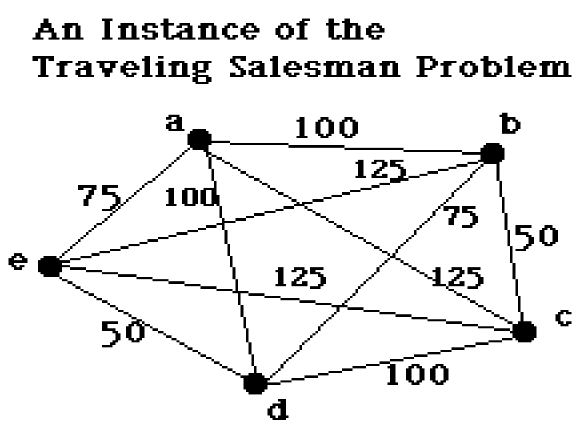
O	X	O
X	#	X
O	X	O

Travelling Salesman Problem (TSP)

Problem:

- A salesman has a list of cites, each of which he must visit exactly once.
- There are direct roads between each pair of cities on the list.
- Find the route that the salesman should follow for the shortest trip that both starts and finishes at any one of the cities.

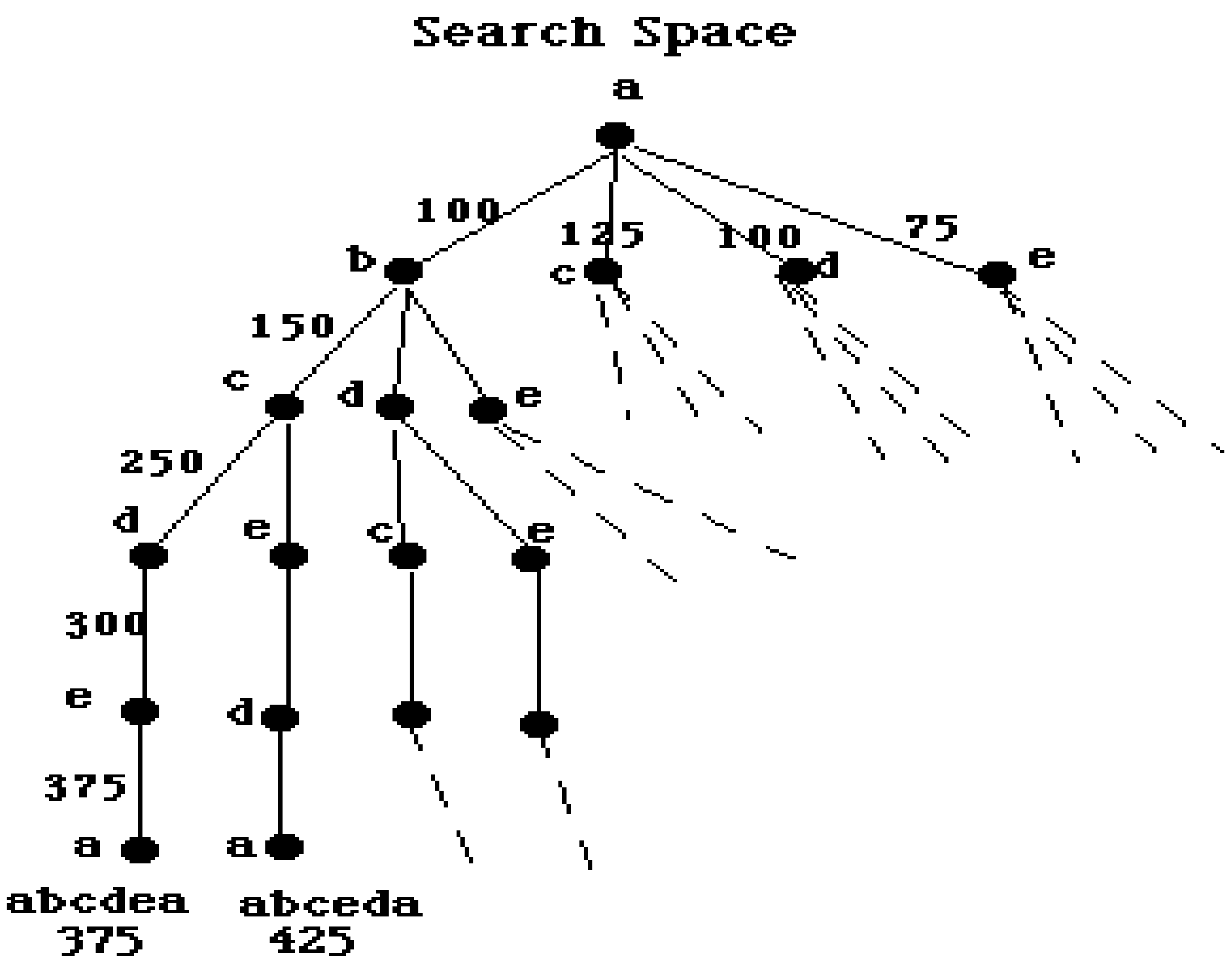
For Example:



State Space representation Of TSP

- **Initial State**: A starting city.
- **Intermediate States**: Each intermediate state is represented as the pair of any two cities and the distance between them.
- **Goal State:** A path from starting city and all other cities travelled exactly once and back to the starting city such that the total distance travelled is minimum.
- **Rules / Operators for Legal Moves:** The city once traversed should not be visited again.

State Space representation Of TSP

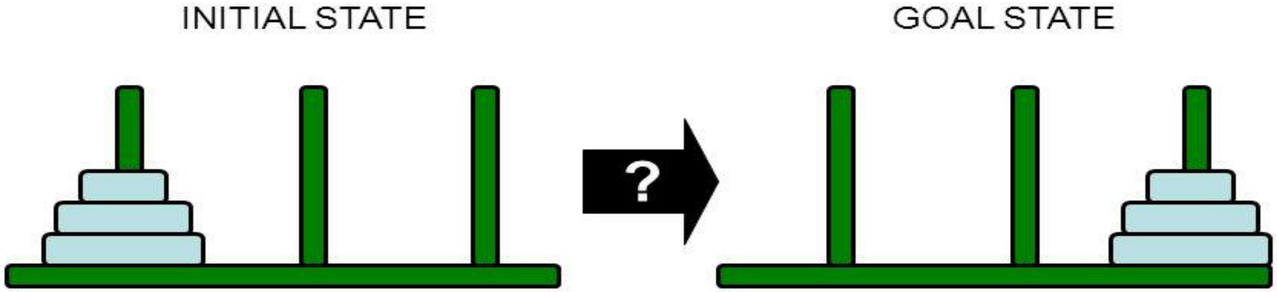


Towers of Hanoi

Problem:

- In Hanoi, there is a monastery whose monk devoted their lives to a very important task.
- In their courtyard there are three towers (poles or posts)
- On these posts, there is a set of **n** disks, each with a hole in the center and of different radii.
- When the monastery was established, all of the disks were on one posts, each disk resting on the one larger than it.
- The monk's task is to move all the disks to one of the other posts.

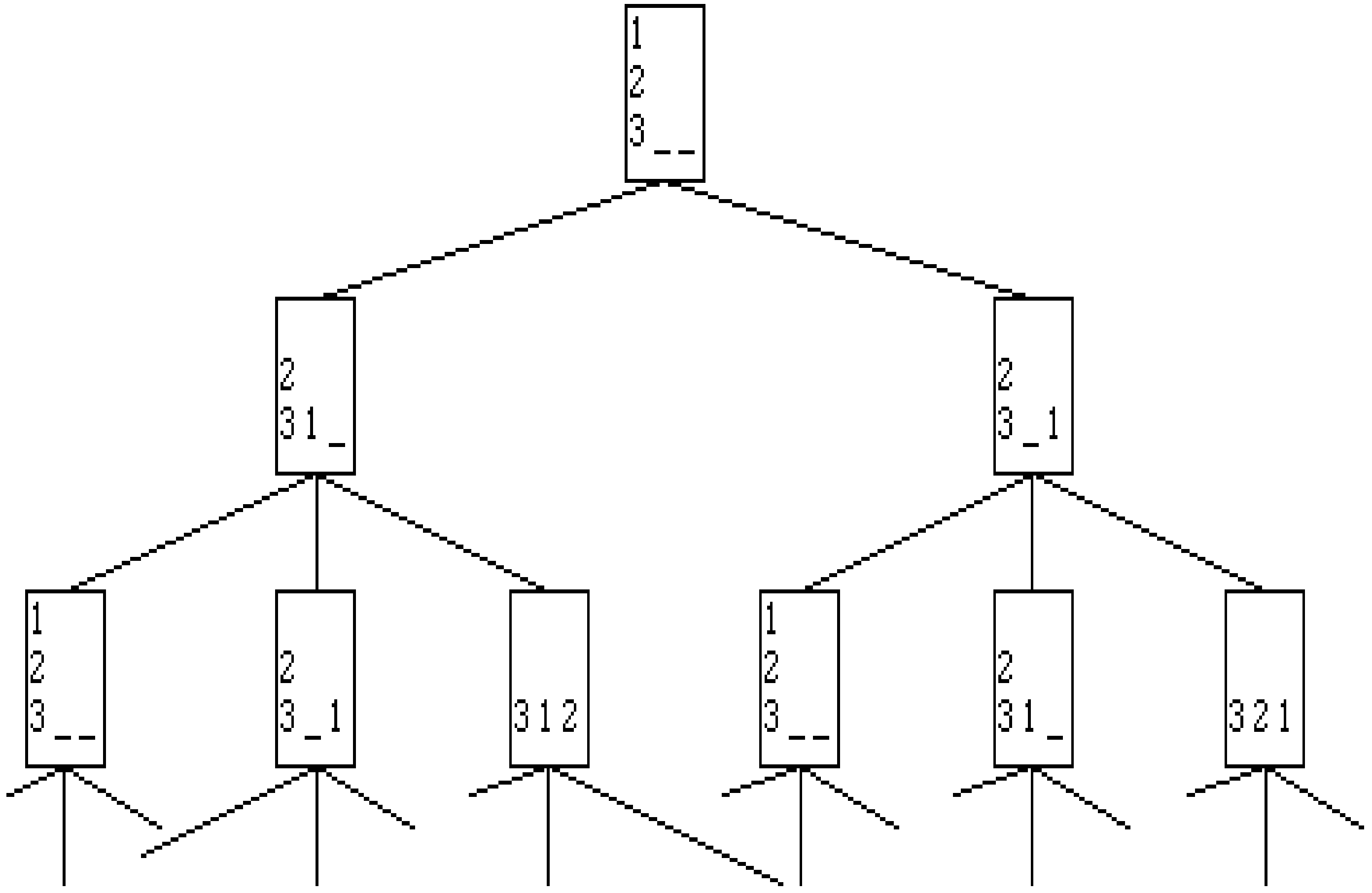
Towers of Hanoi



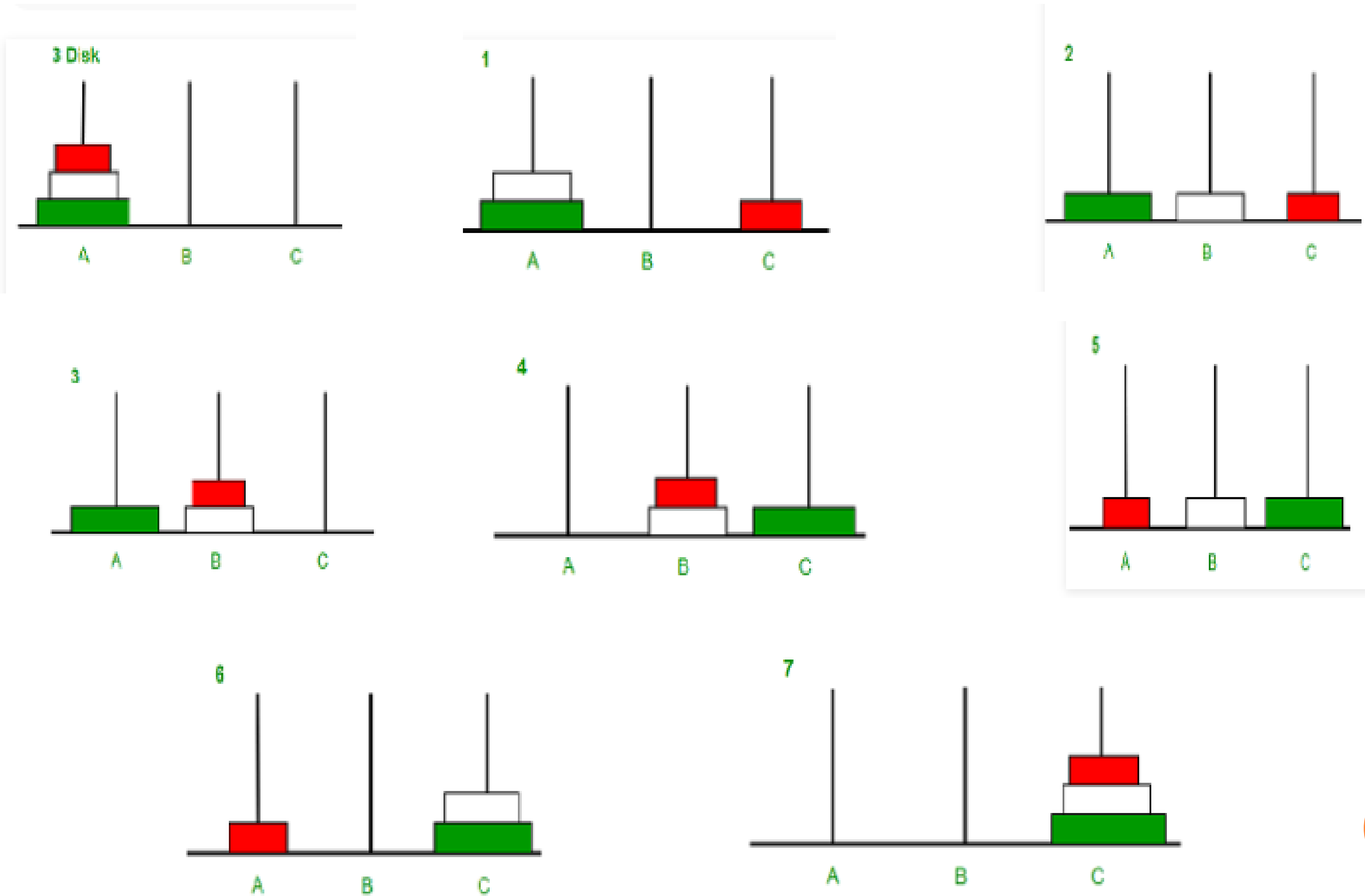
State Space Representation of Tower of Hanoi

- Each state is represented as the tuple (post number, sequence of disks) such as ((1,2,3), (2,1), (3, NIL)).
- **Initial State:** ((1,1,2,3), (2, NIL), (3, NIL))
- **Intermediate State:** Any tuple obtained after applying operators or rules on the current state.
- **Final State:** ((1, NIL), (2, NIL), (3, 1, 2, 3))
- **Rules / Operators:**
  - Only one disk can be moved at one time and all other disks should be on one of the pegs.
  - A larger disk cannot be placed on the smaller disk.
  - The third pole can be used as temporary resting place for the disks.

State Space Representation of Tower of Hanoi Contd……



State Space Representation of Tower of Hanoi Contd……

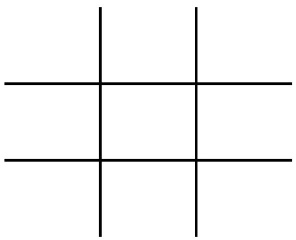


Tic - Tac - Toe Problem

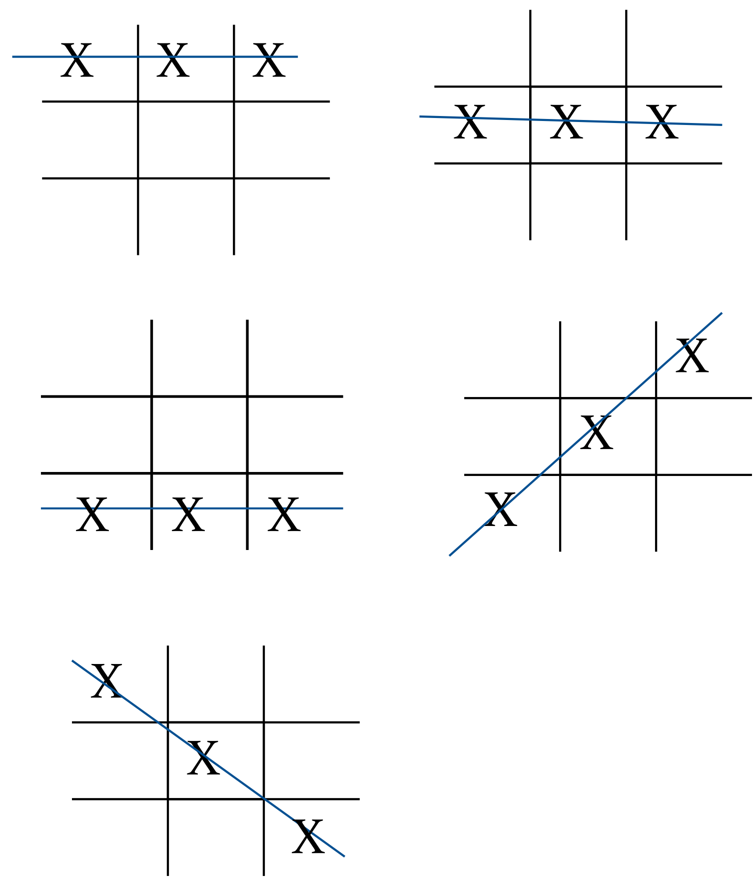
Problem:

- It is a two-player game where one player (say Player 1) marks a letter X and the opponent (Player 2) marks a letter 0.
- There is a 3x3 grid where the players put their letters.
- The player who is able to first mark his three letters in a complete row or column or diagonal wins the game.

TIC-TAC-TOE PROBLEM



Initial state

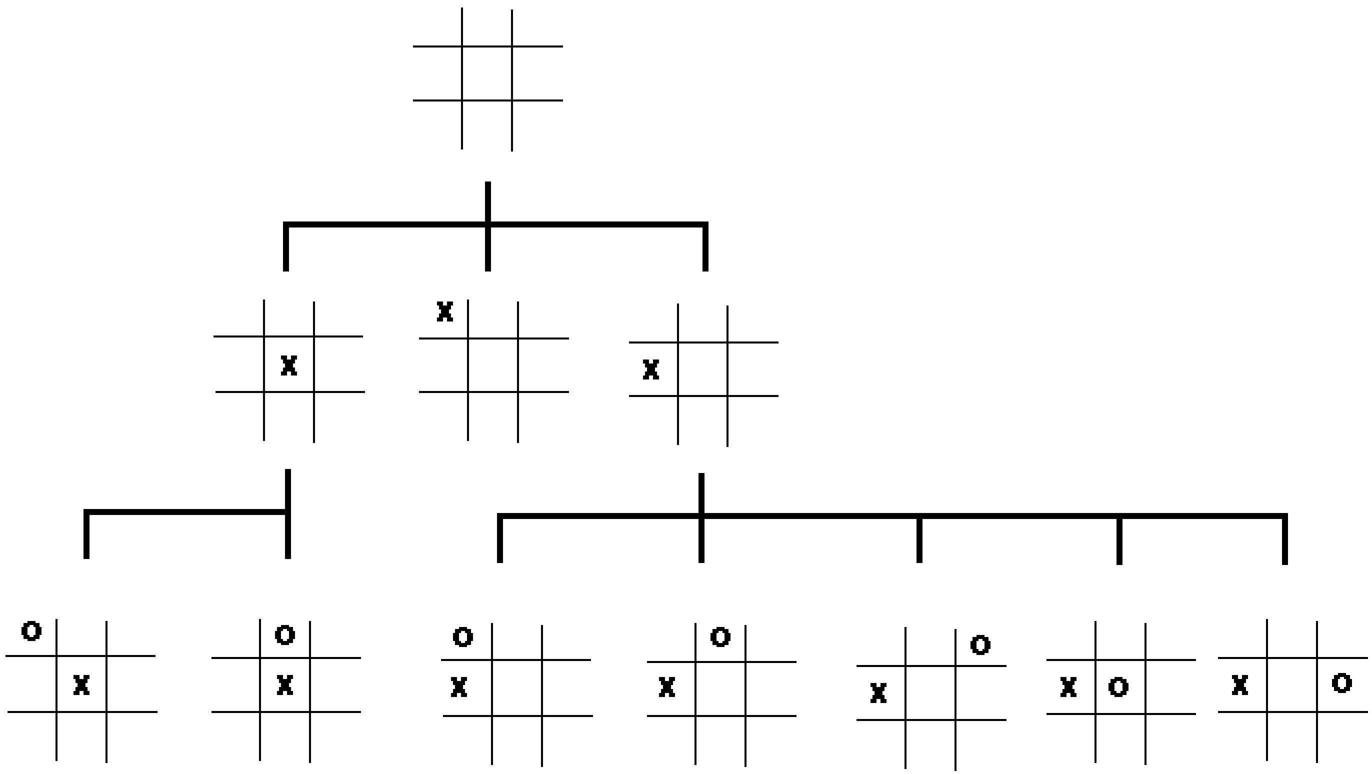


Player 1 will mark X
Player 2 will mark O

State Space Representation Of TIC-TAC-TOE Problem

- Each state is represented by the positions occupied by the letters of the two players in 3x3 matrix along with other empty places.
- **Initial State:** Empty 3x3 matrix.
- **Intermediate State:** Any arrangement of a 3x3 matrix obtained after applying valid rules / operators on the current state.
- **Final State:** Same letters in a complete row or column or diagonal.
- **Rules / Operators:**
  - The players will get turns one after the other
  - The player can mark their own letter on the empty cell.

State Space Representation Of TIC-TAC-TOE Problem Contd…

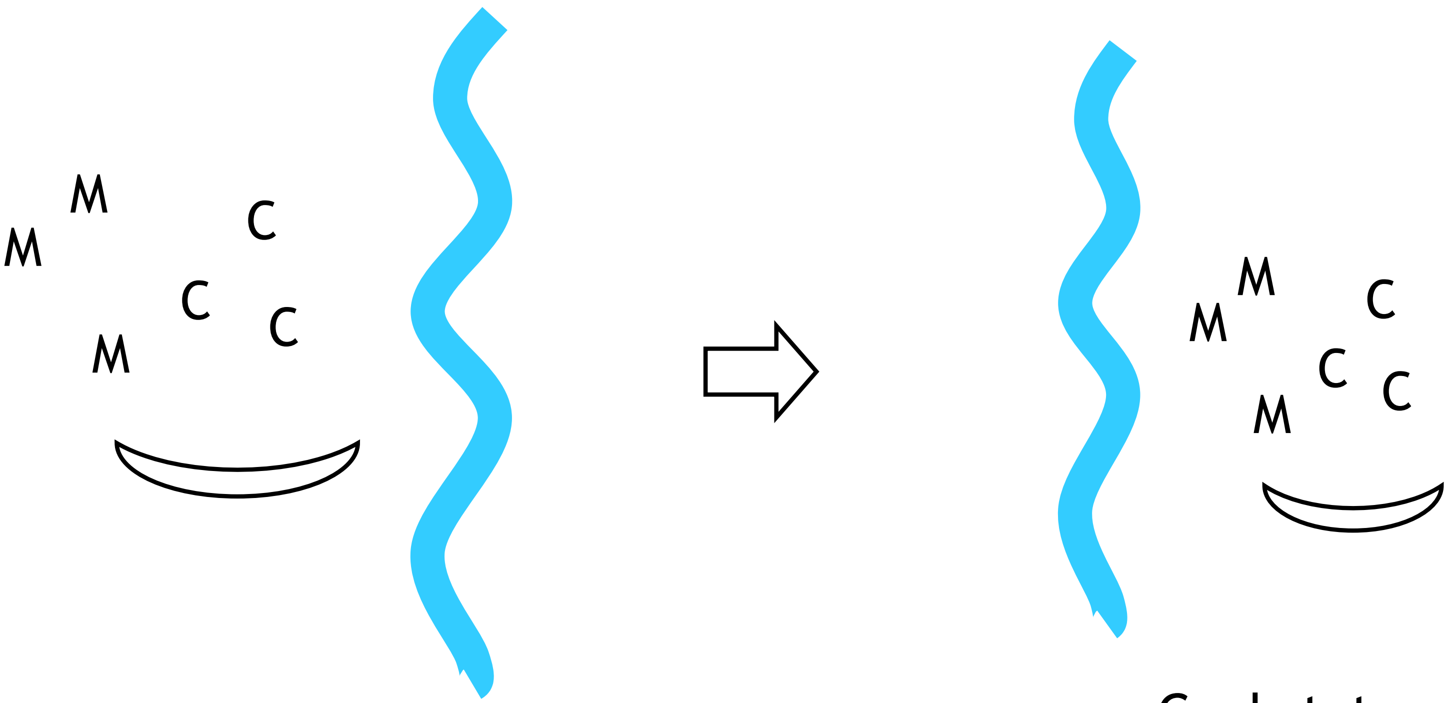


Missionaries and Cannibals Problem

Problem:

- Three missionaries and three cannibals find themselves on one side of the river.
- They have agreed that they would all like to get to the other side of the river.
- But the missionaries are not sure whether they can trust cannibals or not. So, the missionaries want to manage the trip across the river so that the number of missionaries on the either side is never less than the number of cannibals who are on the same side.
- The boat can take only two people at a time.
- How can everyone get to the other side without missionaries being eaten?

Missionaries and Cannibals Problem



Initial state

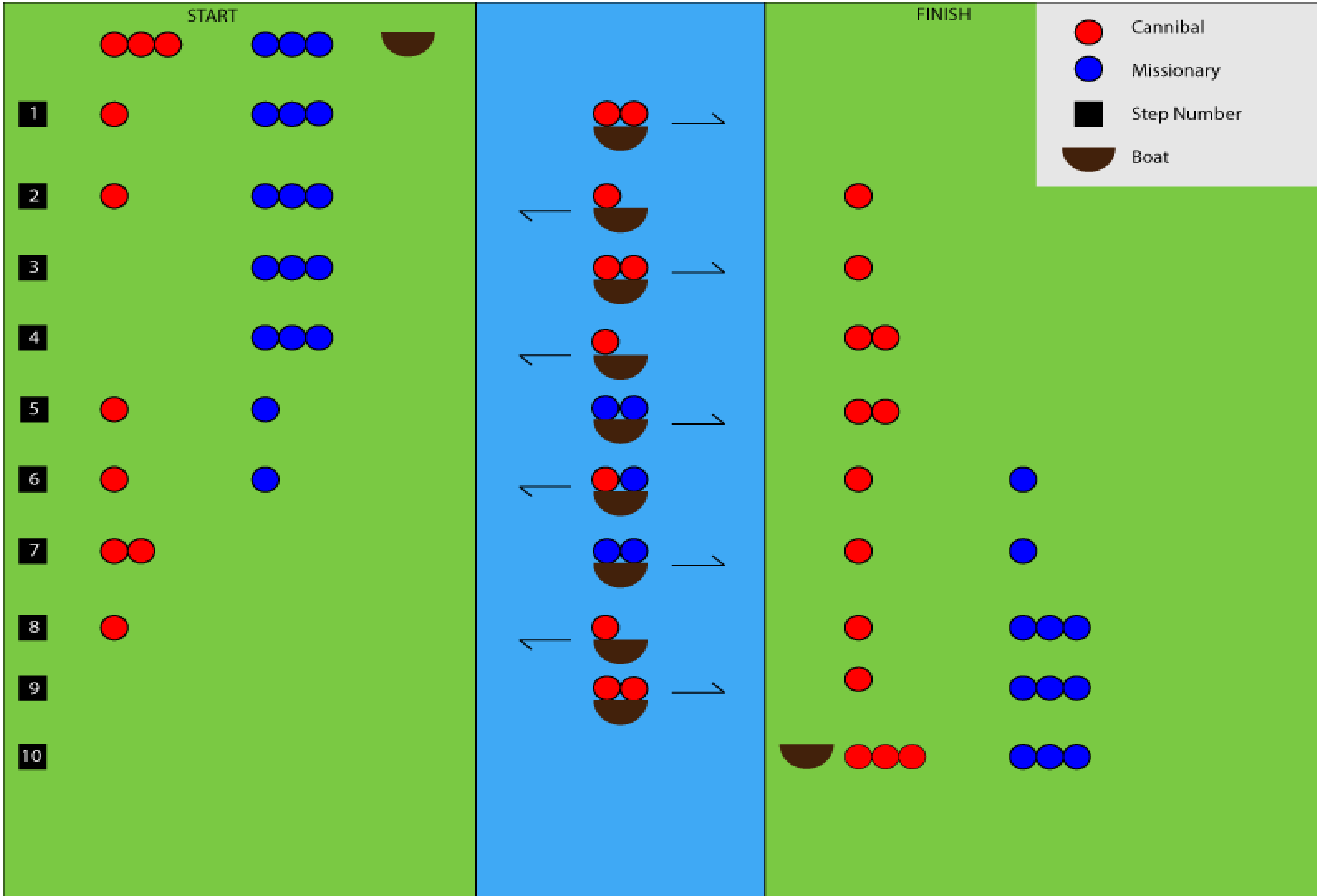
Goal state

State Space Representation of Missionaries and Cannibals Problem

- Each state is represented as 3 tuple < rs, m, c > where rs represents either state which can be either 1 (starting side of river) or 2 (destination side of river). m is the number of missionaries on river side and c is the number of cannibals on the river side.
- **Initial State:** ((1,3,3), (2,0,0))
- **Intermediate States:** ((1,m,c), (2,m’,c’)) where $0 \le m, m', c, c' \le 3$ obtained after applying valid rules.
- **Goal States:** ((1, 0, 0), (2, 3, 3))
- **Rules / Operators:** The number of missionaries on the either side is never less than the number of cannibals who are on the same side.

  The boat can take only two people at a time and cannot cross with zero people as at least one person is required to row the boat.

State Space Representation of Missionaries and Cannibals Problem Contd….



Q. What is the difference between a performance measure and a utility function?

A performance measure is used by an outside observer to evaluate how much successful an agent is. It is a function that returns some number.

An utility function is used by an agent itself to evaluate how desirable states or histories are.
Further, not all agents have utility function but each agent have performance measure.

Q. PEAS for Internet book shopping agent

Environment: Internet

Sensors: User Request, web pages

Actuators: Follow web links, submit data, display records to the users

Performance Measure: Obtain requested books, Minimize expenditure

Monkeys and Banana Problem

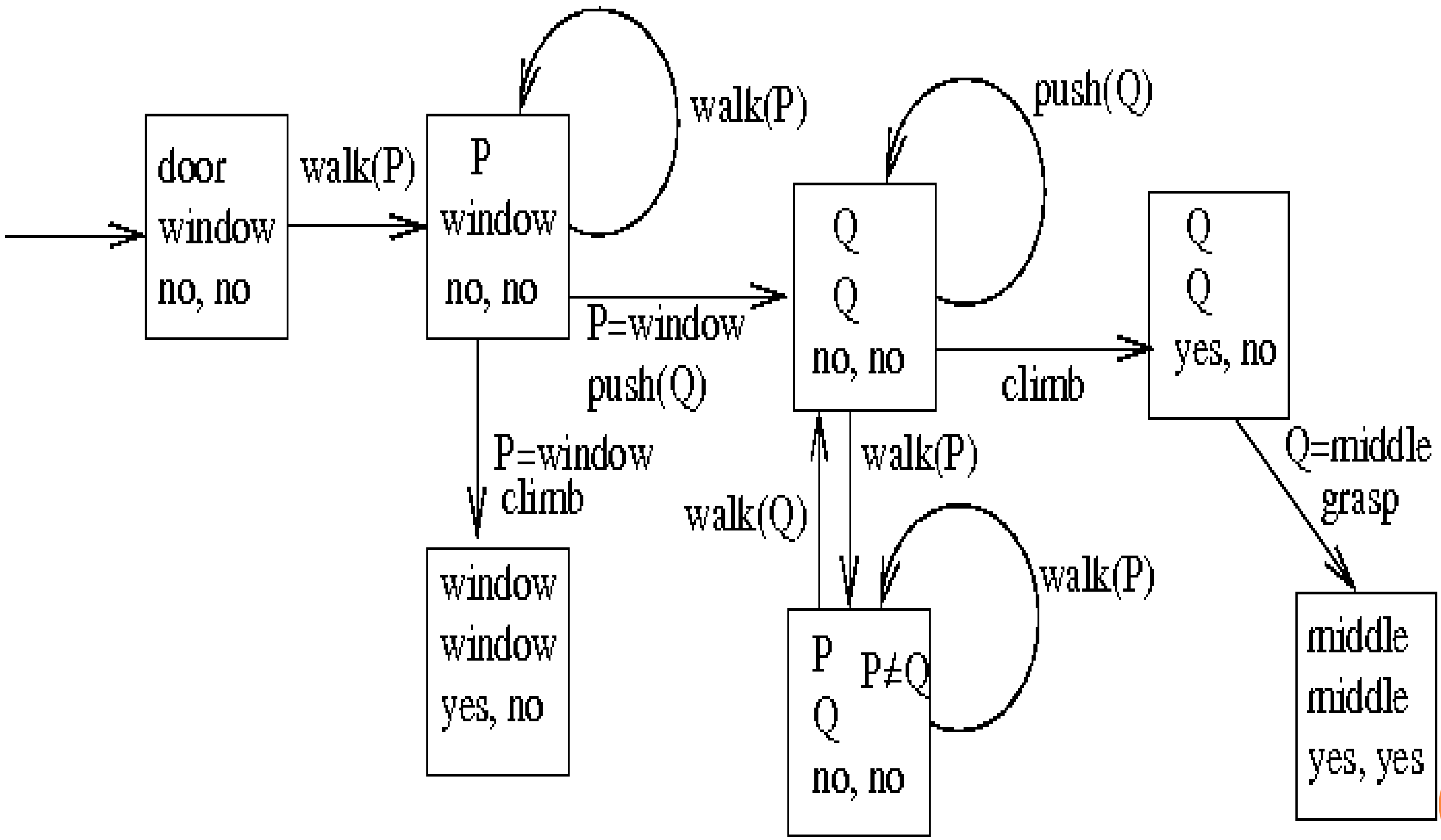
Problem:

- A monkey enters a room via the door.
- In the room, there is a box near the window.
- In the middle of the room, hangs a banana from the ceiling.
- The monkey wants to grasp the banana, and can do so after climbing on the box in the middle of the room

State Space Representation of Monkey and banana Problem

- Each state, is recorded with four variables:
  - the position of the monkey (door, window, middle, any other position P)
  - the position of the box (door, window, middle, any other position P)
  - if the monkey is on the box (yes, no)
  - if the monkey has the banana (yes, no)
- **Initial State:** (door, window, no, no).
- **Final State:** (*, *, *, yes).
- **Rule / Operators:**
  - walk(P): from (M, B, no, H) to (P, B, no, H). // H- banana hanging
  - push(P): from (M, M, no, H) to (P, P, no, H).
  - climb: from (M, M, no, H) to (M, M, yes, H).
  - grasp: from (middle, middle, yes, no) to (middle, middle, yes, yes).

State Space Representation of Monkey and banana Problem Contd...



Theorem Proving Problem

- A number of axioms are given along with the axiom to be proved.
- The user needs to relate the axioms in any particular sequence to find the answer to the solution.
- The state is represented as the number of axiom matched started from the goal state and the new axiom generated to be proved.
- The solution steps continue in the fashion till no further axiom is left to be proved.

Theorem Proving Example

- Consider the given set of axioms:

1. Marcus was a man.
2. Marcus was a Pompeian
3. Marcus was born in 40 AD.
4. All men are mortal.
5. All Pompeians died when the volcano erupted in 79 AD.
6. No Mortal lives longer than 150 years.
7. It is now 2020 AD.

To prove: Is Marcus alive.

Theorem Proving Example Solution

- Marcus was a man. (given)
- All men are mortal. (given)
- Marcus is mortal (using 1,4)
- Marcus age= 2020-40 = 1980 (using 3,7)
- No mortal lives longer than 150 years. (given)

Marcus is dead now.

PROBLEM CHARACTERISTICS

- **In order to choose an appropriate method, it is necessary to analyze the problem with respect to the following considerations.**

1) Is the problem decomposable ?

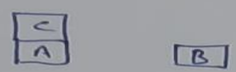
- A very large and composite problem can be easily solved if it can be broken into smaller problems and recursion could be used. Suppose we want to solve.
- Ex:- $\int x^2 + 3x+\sin^2 x \cos^2 x \, dx$

Problem Characteristics

Non Decomposable Problem

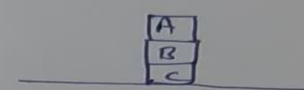
Non decomposable problem :– blocks world

Start



On (C, A)

Goal



On (B, C) and (A, B)

Operators :

(1) CLEAR (x) → on (x, table)

block x has nothing on it

pick up x and put it on the table

(2) CLEAR (x) and clear (y) → on (x, y)

Goal

ON (B, C) and on (A, B)

I ON (B, C)

ON (A, B) II

dependency of I on II

Analyze the problem- Problem Characteristics

2) **Can solution steps be ignored or undone?**

- AI Problems generally fall under three classes ignorable , recoverable and irrecoverable. This classification is with reference to the steps to be used for the solution of a problem.

- Ignorable problems- In which, solution steps can be ignored.

  Ex:- theorem proving.

- Irrecoverable problems- In which solution steps can't be undone.

  Ex- chess problem.

Analyze the problem- Problem Characteristics

- **Recoverable problems-** The problem where steps can be undone. Ex:- 8 puzzle problem

Initial State

1	2	3
8	6	4
7		5

1	2	3
8	6	4
7	5	

1	2	3
8	6	4
7		5

Final State

1	2	3
8		4
7	6	5

Problem Characteristics Contd……

3) Is the Universe Predictable?

- Problems can be classified into those with certain outcome (eight puzzle and water jug problems) and those with uncertain outcome (playing cards) .
- In certain – outcome problems, planning could be done to generate a sequence of operators that guarantees to a lead to a solution. Planning helps to avoid unwanted solution steps.
- For uncertain outcome problems, planning can at best generate a sequence of operators that has a good probability of leading to a solution. The uncertain outcome problems do not guarantee a solution and it is often very expensive.

Problem Characteristics Contd……

4) Is good solution absolute or relative ?

- There are two categories of problems.
- In one (absolute), like the water jug and 8 puzzle problems, we are satisfied with the solution, unmindful of the solution path taken.
- In the other category (relative) not just any solution is acceptable. We want the best, like that of traveling sales man problem, where it is the shortest path.
- In any – path problems, by searching methods we obtain a solution and we do not explore alternatives. For the best-path problems all possible paths are explored using an exhaustive search until the best path is obtained.

PROBLEM CHARACTERISTICS CONTD……

5) Role of Knowledge

- Though one could have unlimited computing power, the size of the knowledge base available for solving problem does matter in arriving at a good the solution.
- For example, the game of playing chess, just the rules for determining legal moves and some simple control mechanism is sufficient to arrive at a solution.
- But additional knowledge about good strategy and tactics could help to constrain the search and speed up the execution of the program. The solution would then be realistic.
- Consider the case of predicting the political trend or News paper understanding. This would require an enormous amount of knowledge even to be able to recognize a solution , leave alone the best.

Problem Characteristics Contd……

6) Is the solution a state or a path?

The problems such as water jug or 8-puzzle or TSP have final solution as a path.

It is easy to look for state as solution where as looking for path requires more effort.

State as a solution:

The bank president ate a dish of pasta salad with the fork.

Problem Characteristics Contd……

7) Does the task requires interaction with the human ?

i) Solitary in which the computer will be given a problem description and will produce an answer.

ii) Conversational, in which there will be intermediate communication between a person and the computer, wither to provide additional assistance to the computer or to provide additional informed information to the user.

PROBLEM CHARACTERISTICS

Problem characteristics	TSP	Water Jug	8- puzzle	Tower of Hanoi	Missionaries and Cannibals
Is the problem decomposable?	**No**	No	No	No	No
Can solution steps be ignored or undone?	Yes	Yes	Yes	Yes	Yes
Is the problem universe predictable?	Yes	Yes	Yes	Yes	Yes
Is a good solution absolute or relative?	Relative	Absolute	Absolute	Absolute	Absolute
Is the solution a state or a path?	Path	Path	Path	Path	Path
What is the role of knowledge?	Lot*	Les	Less	Less	Less
Does the task require human-interaction?	No	Yes	No	No	Yes

Thanks