IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Defining The Problem as a State Space Search

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 01
Presentation Date – 05/09/2024

Course Outcome

At the end of the course, students should be able to:

CO2: Relate appropriate problem-solving methods to optimize the search results.

Topic Learning Outcome

Develop the ability to explore a space of possible states or configurations to find a solution to a problem.

Defining the Problem As a State Space Search

Problem Solving in AI

Artificial Intelligence as a problem solver

- According to Computer science most ***important part of Artificial intelligence is problem solving*** which can be done by using various techniques and algorithms of AI.
- **Ex**: STATE SPACE SEARCH
- The ***aim of Artificial Intelligence*** is to develop a system which can solve the various problems on its own.

Problem Solving in AI : State Space Search

- ***SSS*** is the most commonly used technique in AI ***for problem solving.***
- In, general ***searching*** refers to as finding information one needs.

Problem Solving in AI : State Space Search

- ***State space search :*** is a fundamental technique used ***to solve problems by navigating through a series of states and transitions.***
- **Objective:** To find a sequence of actions that leads from an initial state to a goal state.
- By applying state space search, AI systems ***can effectively tackle a diverse array of problems,*** ranging from robotics and game-playing to natural language processing and scheduling.
- It serves as a crucial tool for ***enabling machines to make intelligent decisions and find optimal solutions in complex, dynamic environments.***

Problem Solving in AI : State Space Search

Problem Solving Agent

- The ***problem-solving agent*** performs precisely by defining problems and find its several solutions.
- ***According to psychology***, "*a problem-solving refers to a state where we wish to reach to a definite goal from a present state or condition.*"
- ***According to computer science***, *a problem-solving is a part of artificial intelligence which encompasses a number of techniques such as algorithms, heuristics to solve a problem.*
- Therefore, ***a problem-solving agent is a goal-driven agent*** and ***focuses on satisfying the goal***.

Problem Solving in AI : State Space Search

Steps Performed by Problem Solving Agent:

- **Goal Formulation**: It is the first and simplest step in problem-solving. It organizes the steps/sequence required to ***formulate one goal out of multiple goals*** as well as ***actions to achieve that goal***. Goal formulation is ***based on the current situation*** and the agent's performance measure (discussed below).

- **Problem Formulation**: It is the most important step of problem-solving which decides ***what actions should be taken to achieve the formulated goal***. There are following **five components** involved in problem formulation:

Problem Solving in AI : State Space Search

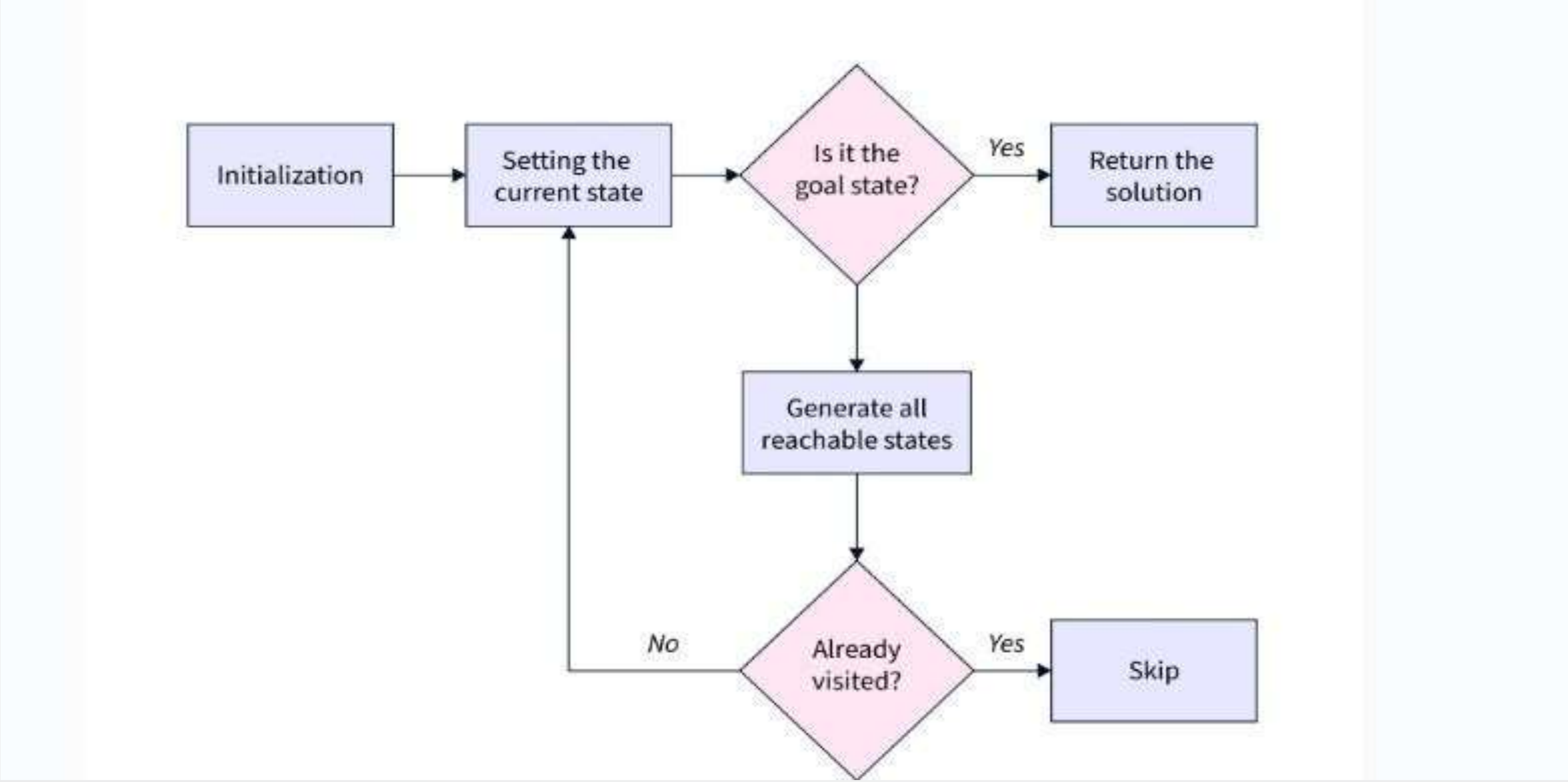
Steps for formulating Problems

- **Initial State**: It is the starting state or initial step of the agent towards its goal.
- **Actions**: It is the description of the possible actions available to the agent.
- **Transition Model:** It describes what each action does.
- **Goal Test**: It determines if the given state is a goal state.
- **Path cost**: It assigns a numeric cost to each path that follows the goal. The problem solving agent selects a cost function, which reflects its performance measure. Remember, *an optimal solution has the lowest path cost among all the solutions.*

State Space Search Representation

- **Tuple T**: **{S, A, Action (s), Result (s, a), Cost(s, a)}**
  - **S**- Total number of states (Start, Goal, Intermediate)
  - **A**- Set of all possible actions (legal and illegal moves) / (All rules).
  - **Action(s)-** selected action i.e. which action is possible for current state.
  - **Result(s, a)** - Resultant state i.e. it determines which state is reached by performing action ‘a’ on state ‘s’.
  - **Cost(s, a)** - it determines how much cost is required to perform the action ‘a’ on state ‘s’. e.g., 1 for every move. Searching should be with Minimum cost (best solution)

Steps in State Space Search



Features of State Space Search

The efficiency and effectiveness of state space search are heavily dependent on several principles and characteristics.

✓ **Expansiveness**: The number of successors that each state can generate. This impacts how many new states are explored from a given state.

✓ **Branching Factor**: The average number of successors in each state. It *influences the width of the search tree and the overall complexity of the search.*

✓ **Depth**: The *length from the initial state to the goal state* in the search tree. Deeper search trees can increase the time required to find a solution.

Features of State Space Search / Search Algorithm (Performance of search algorithm)

1. **Completeness**: A search is said to be complete if it guarantee to return a at least one solution for any random input.
2. **Optimality**: A search strategy is optimal if it guarantees finding the best solution(lowest path cost).
3. **Time Complexity**: It is the measure of the time for an algorithm to complete its task.
4. **Space Complexity**: The amount of memory required to carry out the search. This depends on the number of states that need to be stored in memory simultaneously.

Toy Problems

Most of the problems are toy problems, like:

- Puzzle-solving (8, 15-puzzle)
- 8-queens
- Water Jug Problem
- Crypt arithmetic

Real-world problems

But in fact, these techniques are used to solve Real problems too:

- Route-finding in airline travel planners
- Travelling Salesperson Problem (planning drill movements for automatic circuit board drills)
- VLSI layout (cell layout and channel routing) g Automatic manufacturing (assembly sequencing)

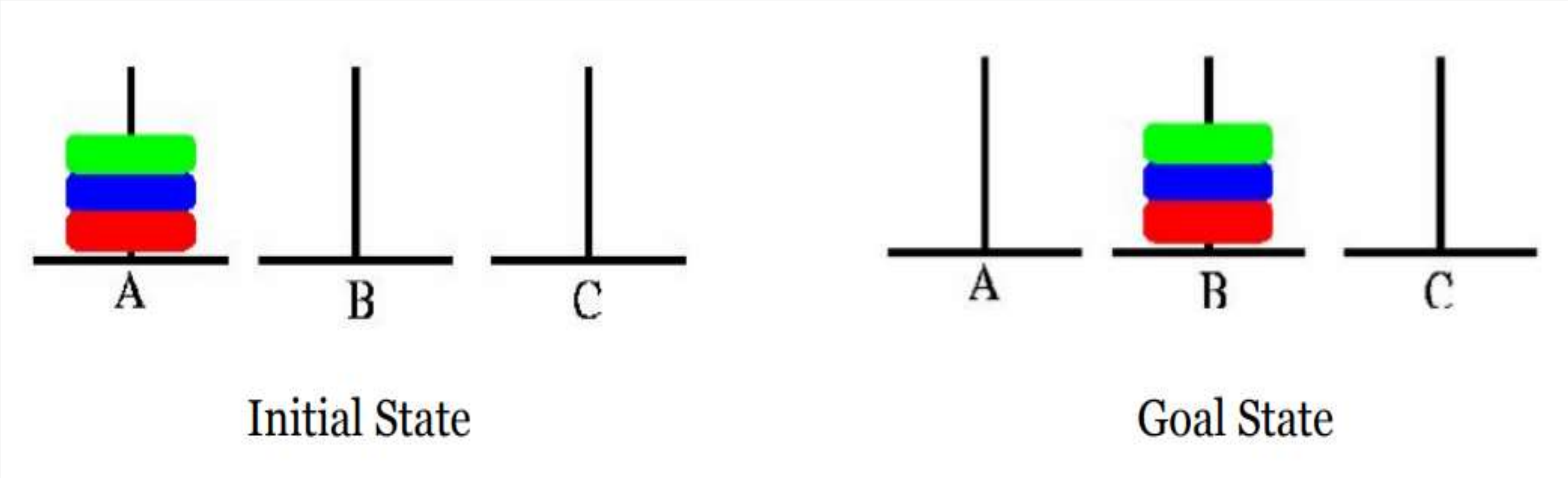
Example-1(Toy Problem)
Pegs and Disks Problem

State Space Representation of Pegs and Disks Problem

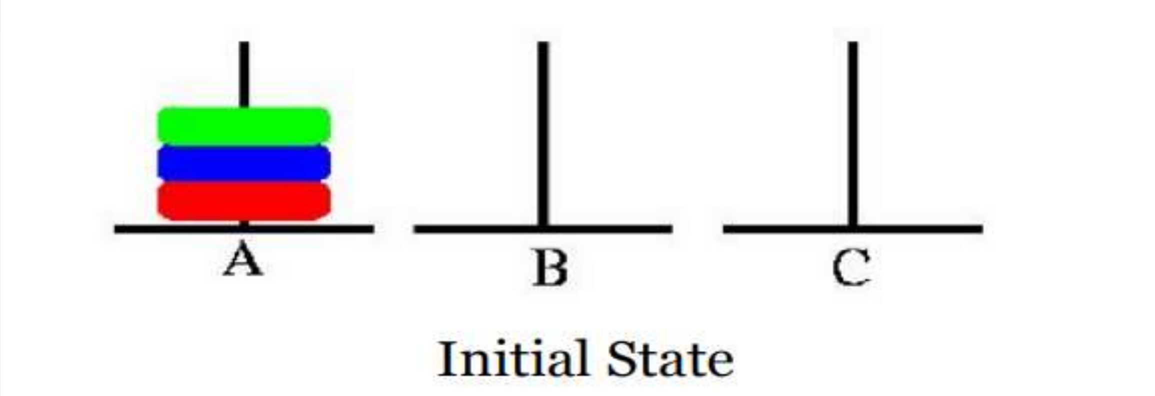
Scenario / The Problem Formulation:

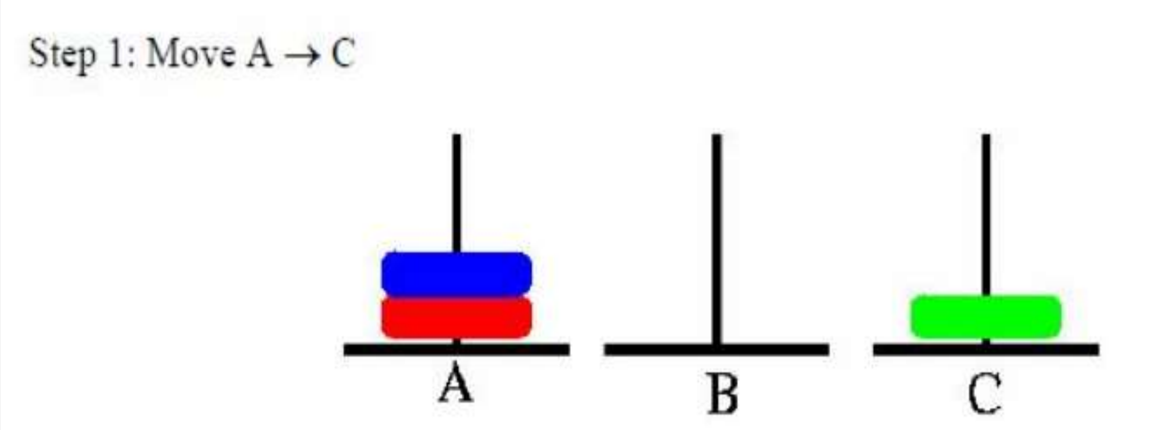
- **States**: We have 3 pegs and 3 disks.
- **Initial State**: all the pegs are in the needle A.
- **Goal State**: all the pegs are in the needle B.
- **Actions**: one may move the topmost disk on any needle to the topmost position to any other needle.
- **Transition Model**: Describes the state that arises after carrying out a certain action.
- **Path Cost**: The uniform cost of every motion is one.

Pegs and Disks Problem



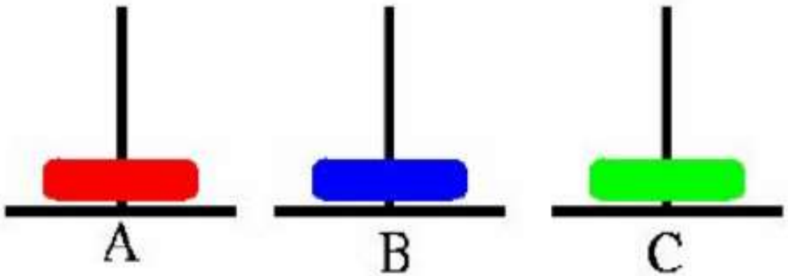
Pegs and Disks Problem (Contd..)



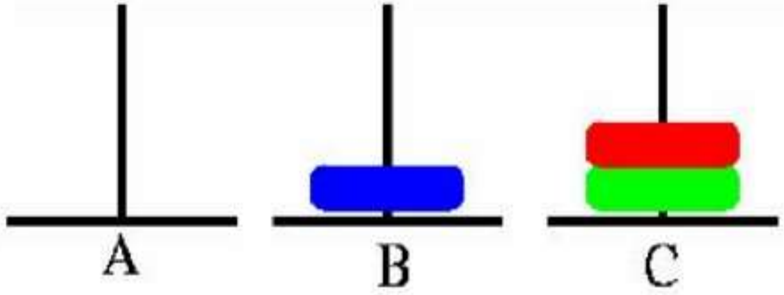


Pegs and Disks Problem (Contd..)

Step 2: Move A → B

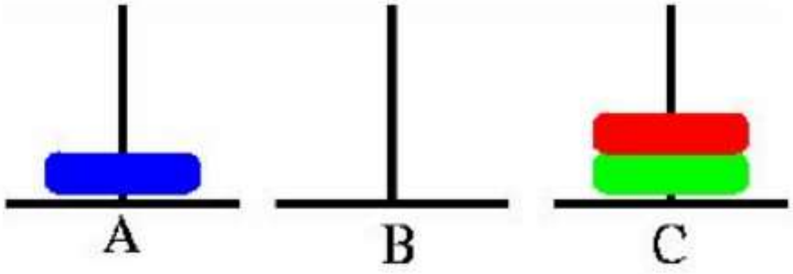


Step 3: Move A → C

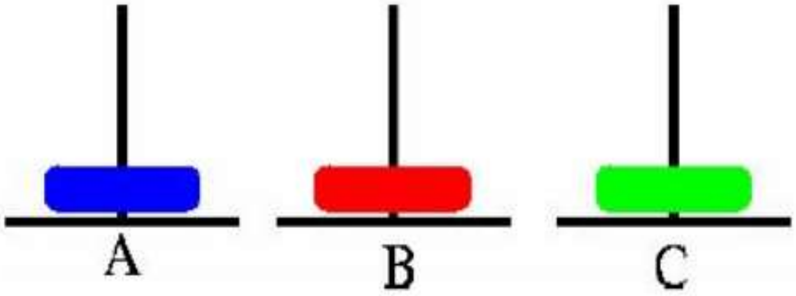


Pegs and Disks Problem (Contd..)

Step 4: Move B→ A

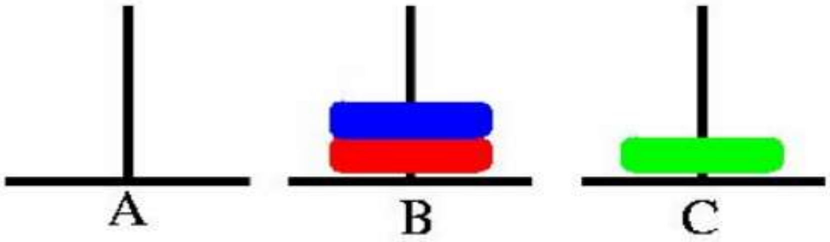


Step 5: Move C → B

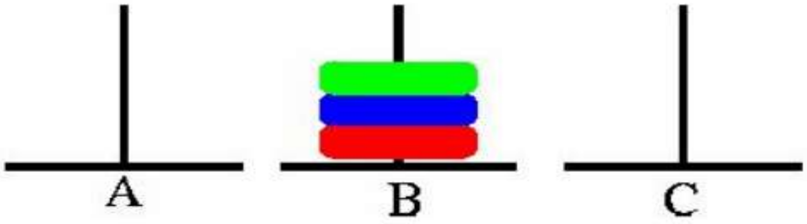


Pegs and Disks Problem (Contd..)

Step 6: Move A → B



Step 7: Move C→ B



Goal State

Path Cost = 7

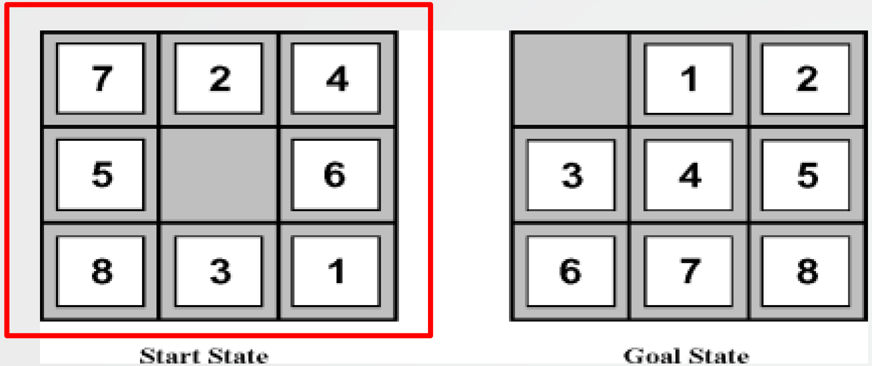
Example-2 (Toy Problem)
The 8-Puzzle Problem

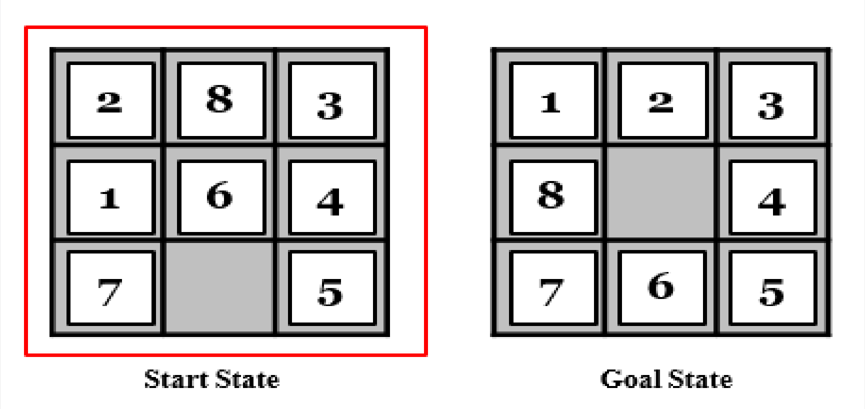
State Space Representation of The 8-Puzzle Problem

Scenario / The Problem Formulation:

- **State**: Specification of each of the eight tiles in the nine squares (the blank is in the remaining square).
- **Initial state**: Any state.
- **Successor function (actions)**: Blank moves Left, Right, Up, or Down.
- **Transition Model**: It returns the resulting state as per the given state and actions.
- **Goal test**: Check whether the goal state has been reached.
- **Path cost**: Each move costs 1. The path cost = the number of moves.

The 8-Puzzle Problem (Contd..)





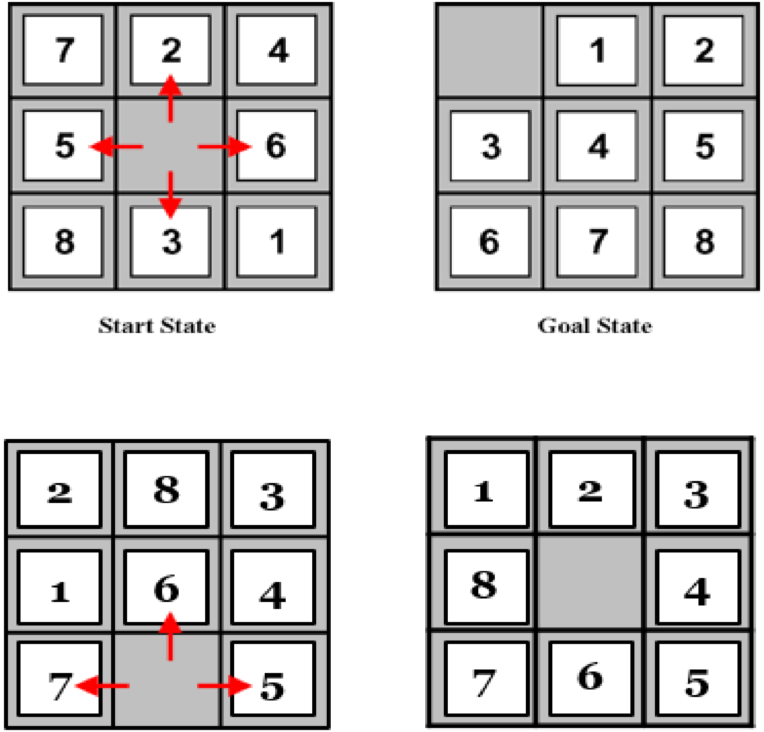
- **State:** Specification of each of the eight tiles in the nine squares (the blank is in the remaining square).

<u>Examples:</u>

θ = {7, 2, 4, 5, 0, 6, 8, 3, 1}

θ = {2, 8, 3, 1, 6, 4, 7, 0, 5}

The 8-Puzzle Problem (Contd..)

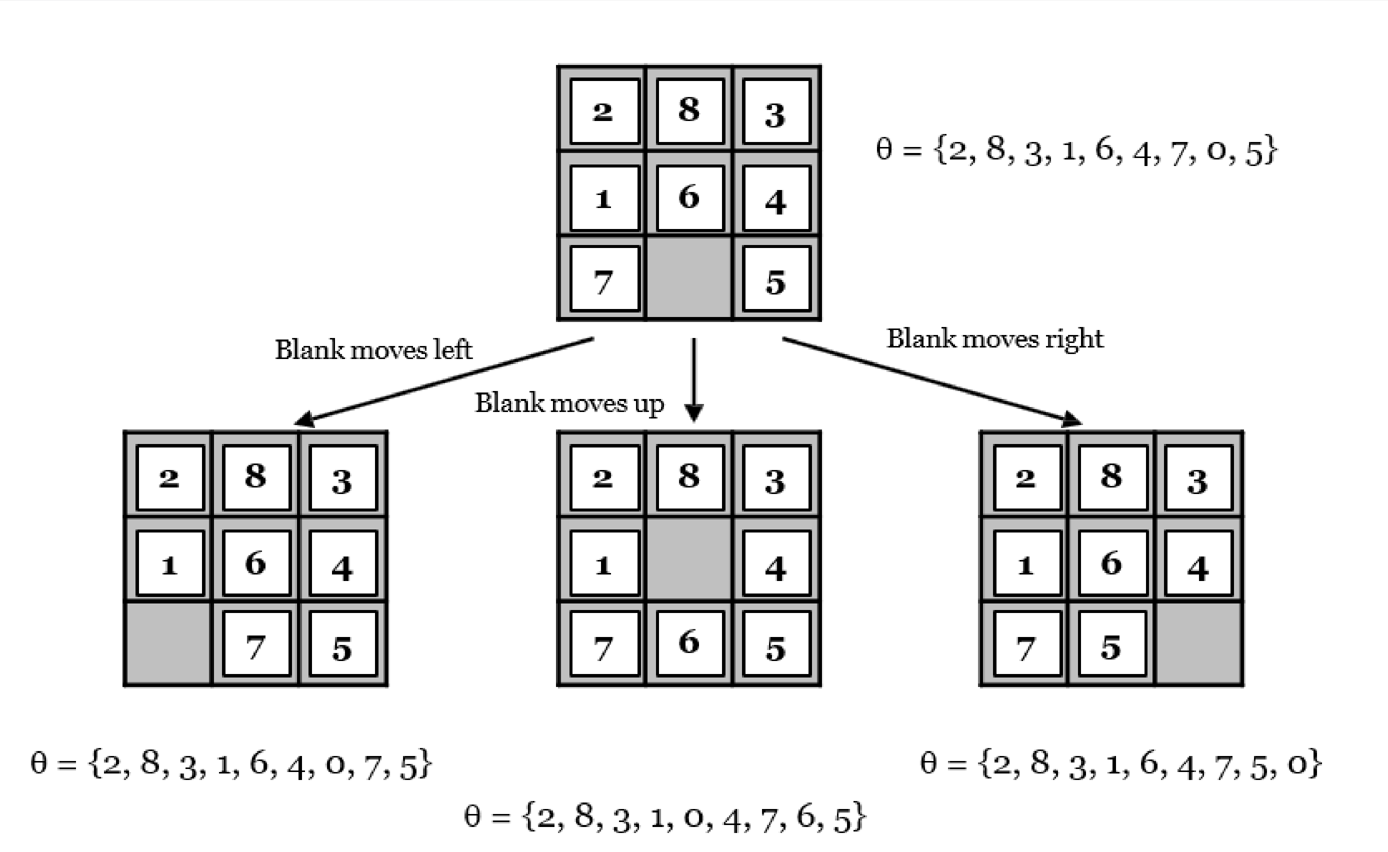


Start State Goal State

Start State Goal State

- **Successor function (actions):** Blank moves Left, Right, Up, or Down

The 8-Puzzle Problem (Contd..)



Continue to the next step and so on…

Example-3 (Toy Problem) Vacuum World Problem

State Space Representation of Vacuum World Problem

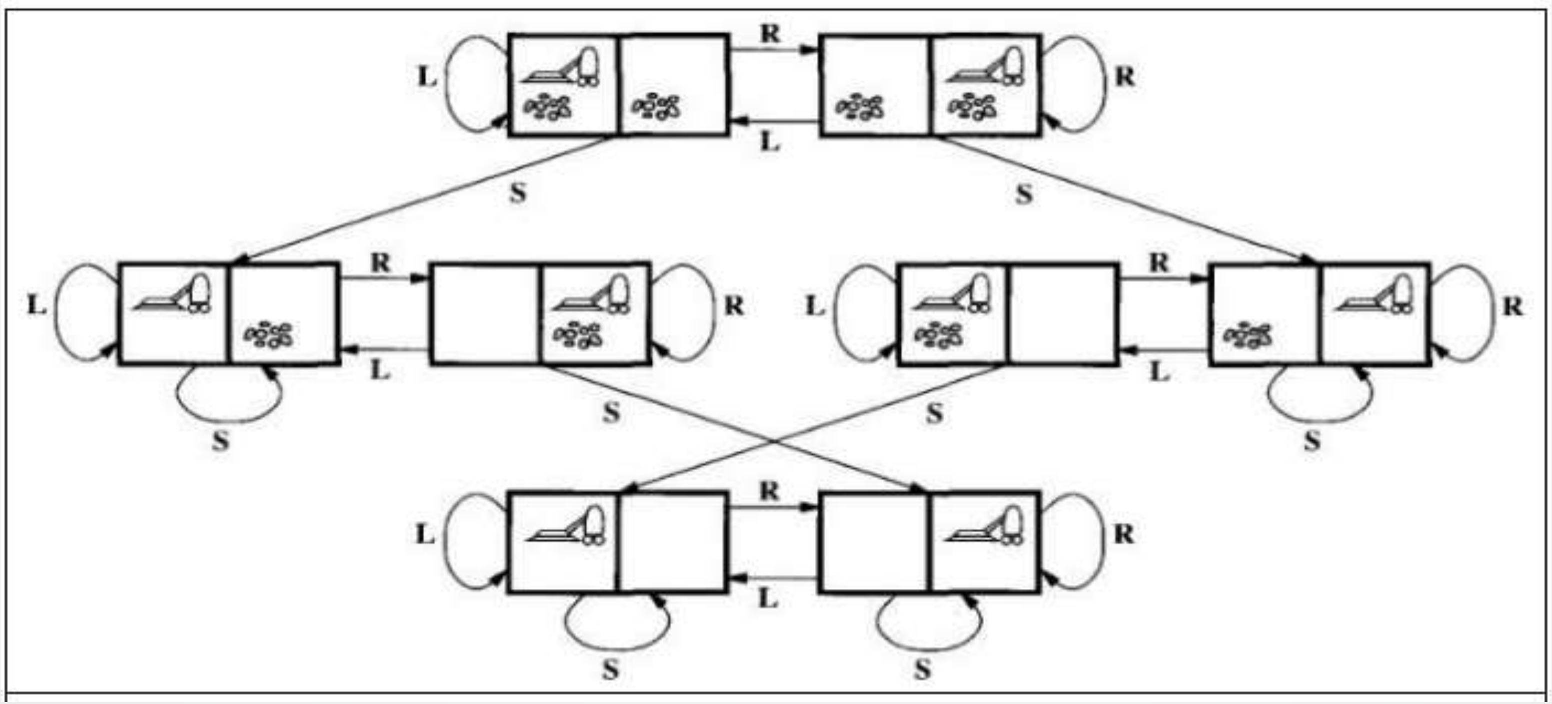
Scenario / The Problem Formulation

- **State**: The agent is in one of two locations, each of which might or might not contain dirt. Thus there are $2 \times 2^2 = 8$ possible world states. (*dirt or robot location*)
- **Initial state**: *Any state* can be designated as initial state.

State Space Representation of Vacuum World Problem

- **Successor function (actions)**: This generates the legal states that results from trying the three actions (*left, right, suck*). The complete state space is shown in the next slide.
- **Goal test**: This tests whether all the squares are *clean*.
- **Path cost**: Each step costs *one*, so that the path cost is the number of steps in the path.

Vacuum World Problem



(The state space for the vacuum world)

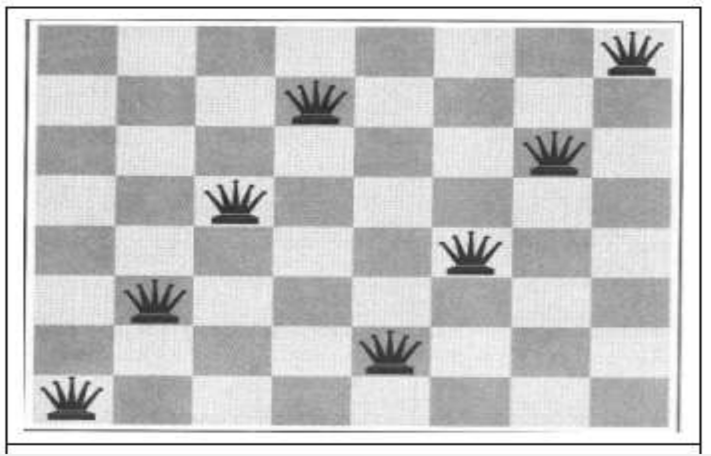
Arcs denote actions:
L = Left,
R = Right

Example-4 (Toy Problem)
8-Queens Problem

State Space Representation of 8-Queens Problem

Scenario / The Problem Formulation

The aim of this problem is to place eight queens on a chessboard in an order where no queen may attack another. A queen can attack other queens either diagonally or in same row and column.



State Space Representation of 8-Queens Problem

- **States**: Arrangement of any 0 to 8 queens on the chessboard.
- **Initial State**: An empty chessboard
- **Actions**: Add a queen to any empty box.
- **Transition model**: Returns the chessboard with the queen added in a box.
- **Goal test**: Checks whether 8-queens are placed on the chessboard without any attack.
- **Path cost**: There is no need for path cost because only final states are counted.

Example-5 (Real World Problem)
Airline Travelling Problem

Airline Travelling Problem

The Problem Formulation

- **States**: Each is represented by a location and the current time.
- **Initial State**: This is specified by the problem.
- **Actions**: This returns the states resulting from taking any scheduled flight, leaving later than the current time plus the within airport transit time, from the current airport to another.
- **Goal test**: Are we at the destination by some pre-specified time?
- **Path cost**: This depends on the monetary cost, waiting time, flight time, customs and immigration procedures, seat quality, time of day, type of air place, frequent-flyer mileage awards and so on.

Applications of State Space Search and it’s Challenges

Applications of State Space Search

State space search is extensively employed in many different fields, such as:

- **Pathfinding**: Finding the best pathways using algorithms such as A* in robotics and GPS.
- **Puzzle solving**: resolving puzzles like Rubik's Cube, Sudoku, and the 8-puzzle.
- **AI for gaming**: To assess potential moves in board games like chess, checkers, and others.

Applications of State Space Search (Contd..)

- **Planning:** The automated scheduling of tasks in logistics and robotics to achieve a specific objective.
- **Natural language processing:** involves computer translation and sentence parsing by examining many interpretations.
- **Theorem Proving**: Examining logical proofs by looking for potential logical inference sequences.

Challenges in State Space Search

- **Complexity**: High branching factors can cause an exponential growth in the number of states to be explored.
- **Resource Limitations**: Memory and processing power limit the size of the state space that can be practically searched.
- **Quality of Heuristics**: The effectiveness of the search is often limited by the quality of the heuristic function.

Thank You

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Production System and It’s Characteristics

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 02
Presentation Date – 10/09/2024

Course Outcome

At the end of the course, students should be able to:

CO2: Relate appropriate problem-solving methods to optimize the search results.

Topic Learning Outcome

Utilize knowledge-based approaches to simulate human reasoning and decision-making processes.

Production System

Production System

- AI Production Systems are the ***backbone of decision-making***. These systems ***automate complex tasks*** through ***production rules***, efficiently processing data and generating insights.
- ***Production systems*** are a procedural approach to describe and execute a search process.
- It is one of the formalisms ***that helps in structuring AI programs in a way that facilitates describing and performing search process***.

Production System (Contd..)

- A production system, also known as a ***rule-based system***.
- It consists of a knowledge base of rules and a inference engine that applies those rules to solve problems or make decisions within a specific domain.
- It consists of two components: **rule and action**.

Rules *recognize the* ***condition,*** *and*

the ***actions*** *part has the knowledge of* ***how to deal with the condition.***

Production System (Contd..)

- ***In simpler words***, the production system in AI contains a set of rules ***which are defined by the left side and right side of the system.***
- The left side contains a set of things to watch for (condition), and the right side contains the things to do (action).

Elements / Components of Production System

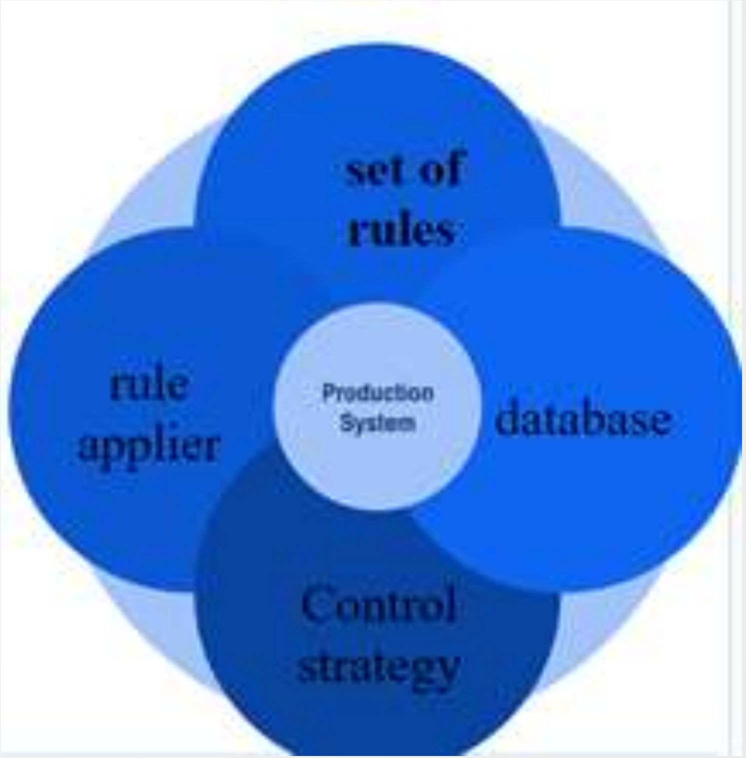
An AI production system has four main elements which are as follows:

1) ***<u>A set of rules</u>*** *(procedures)*

2) ***<u>Knowledge base</u>*** *(store the knowledge in a database)*

3) ***<u>Control strategy</u> (order of action so that conflict can be resolved in a minimum time)***

4) ***<u>Rule applier / User</u>*** *(Rules applied based on control strategy)*



Elements / Components of Production System (Contd..)

- ***A set of rules***, "each consisting of a left side (a pattern) that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied."

For example:

In the <u>Water Jug Problem</u>,

Rule (x, y) is x<4 ->(4, Y) indicates Fill the 4-gallon jug

Rule (x, y) if y<3 -> (x, 3) indicates Fill the 3-gallon jug.

Elements / Components of Production System (Contd..)

- ***One or more knowledge/databases*** "that contain whatever information is appropriate for the particular task. Some plans of the database may be permanent, while other plans of it may pertain only to the solution of the current problem."

- ***A control strategy*** "that specifies the order in which the rules will be compared to the database and a way of resolving the conflicts that arise when several rules match at once."

- ***A rule applier.***

Steps to Solve the Problem Using Production System

- **S1**: First ***reduce the problem*** so that it can be shown in a precise format (i.e, initial state and goal state)
- **S2**: Problem can be solved ***by searching a path through space.***

  [start→ goal]
- **S3**: Solving process can be modelled through a production system.

Advantages of Production System

- Production systems provide an ***excellent tool for structuring AI programs.***
- Production Systems are ***highly modular / flexible,*** because the individual rules can be added, removed or modified independently.
- The production ***rules are expressed in a natural form***, so the statements contained in the knowledge base should the a recording of an expert thinking out loud.

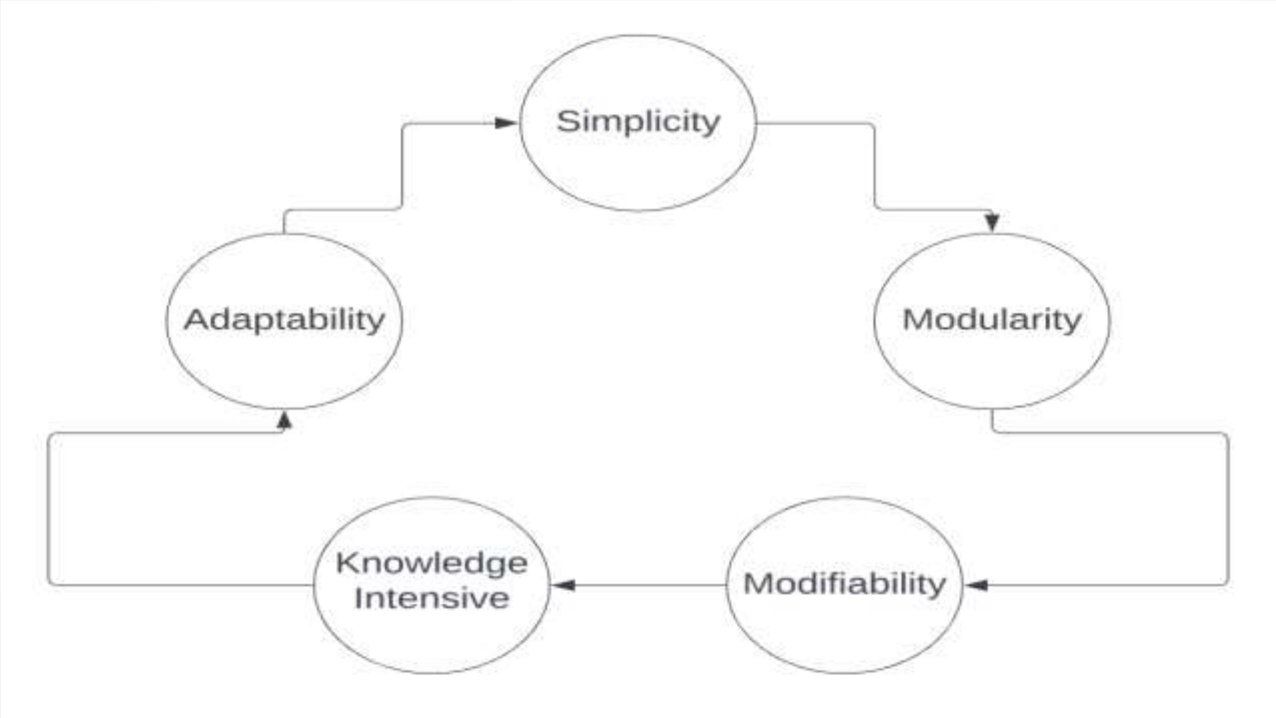
Disadvantages of Production System

- One important disadvantage is the fact that ***it may be very difficult to analyze the flow of control within a production system*** because the individual rules don’t call each other.

Key Features / Characteristics
of
Production System

Characteristics of Production System

- A production system has the following features:



Characteristics of Production System (Contd..)

- **Simplicity**: Production Systems offer a straightforward way to encode and execute rules, making them accessible for developers and domain experts.

- **Modularity**:

  - These systems are composed of modular components, *allowing for the addition, removal, or modification of rules* without disrupting the entire system.

  - This modularity *enhances flexibility and ease of maintenance.*

Characteristics of Production System (Contd..)

- **Modifiability**:
  - AI Production *Systems are highly adaptable*.
  - *Rules can be updated or replaced without extensive reengineering*, ensuring the system remains up-to-date and aligned with evolving requirements.
- **Knowledge-intensive**: They excel in handling knowledge-rich tasks, relying on a comprehensive global database.
- **Adaptability**: AI Production Systems *can dynamically adapt to new data and scenarios.* This adaptability allows them to continuously improve.

Classification of Production System

Classification of Production System

A production system is ***classified into four main classes*** which are:

- **<u>Monotonic Production System</u>**: ***(Rules are independent in nature)***:
  - In a monotonic production system, the use of one rule never prevents the involvement of another rule when both the rules are selected at the same time. Hence, it enables the system to apply rules simultaneously.

Classification of Production System (Contd..)

- **Non-monotonic Production System: (Which is not true):**
  - non-monotonic production systems are ***more dynamic and adaptive***.
  - During execution, the ***system can add, modify, or retract rules***. Therefore, this flexibility makes them excellent for situations where the knowledge base needs to change in response to shifting circumstances, offering high adaptability and responsiveness.

Classification of Production System (Contd..)

- **Partially Commutative Production System**:
  - In this production system if a set of rules is used to change state A to state B then any allowable combination of these rules will also produce the same results (convert state A to state B).
- **Commutative Production System**: ***(It is both monotonic and partially commutative):***
  - These type of production systems is *used when the order of operation is not important*, and *the changes are reversible.*

Problems Solved Using Production System

1. Water Jug Problem

Water Jug Problem

Problem Definition of water Jug Problem:

“You are given two jugs, a 4-liter one and a 3-liter one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 litters of water into a 4-liter jug.”

Water Jug Problem

State-space search representation of water Jug Problem:

- **State**: (x, y)
- where x represents the quantity of water in a **4-liter jug** and y represents the quantity of water in a **3-liter jug**.
- That is, x = 0, 1, 2, 3, or 4 y = 0, 1, 2, 3
- **Start state: (0, 0).**
- **Goal state: (2, n) for any n.**
- Here need to start from the current state and end up in a goal state.

Water Jug Problem (Contd..)

Production Rules for Water Jug Problem

Rule No	Left – Right Rule	Description
1	(x, y) if X<4 ->(4, Y)	Fill the 4-liter jug
2	(x, y) if Y<3 -> (x, 3)	Fill the 3-liter jug
3	(x, y) if x>0 -> (x-d, y)	Pour some water out of the 4-liter jug.
4	(x, y) if Y>0 -> (x, y-d)	Pour some water out of the 3-liter jug.
5	(x, y) if x>0 -> (0, y)	Empty the 4-liter jug on the ground
6	(x, y) if y>0 -> (x,0)	Empty the 3-liter jug on the ground

Water Jug Problem (Contd..)

Production Rules for Water Jug Problem

Rule No	Left – Right Rule	Description
7	(x, y) if X+Y >= 4 and y>0 -> (4, y-(4-x))	Pour water from the 3-liter jug into the 4-liter jug until the 4-liter jug is full
8	(x, y) if X+Y>=3 and x>0 -> (x-(3-y), 3))	Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.
9	(x, y) if X+Y <=4 and y>0 -> (x+y, 0)	Pour all the water from the 3-liter jug into the 4-liter jug.
10	(x, y) if X+Y<=3 and x>0 -> (0, x+y)	Pour all the water from the 4-liter jug into the 3-liter jug.
11	(0, 2) -> (2, 0)	Pour the 2-liter water from the 3-liter jug into the 4-liter jug.
12	(2, Y) -> (0, y)	Empty the 2-liter in the 4-liter jug on the ground.

Water Jug Problem (Contd..)

The solution to Water Jug Problem (General Approach)

1. Current state = (0, 0)

2. Loop until the goal state (2, 0) reached

   – Apply a rule whose left side matches the current state

   – Set the new current state to be the resulting state

Water Jug Problem (Contd..)

The solution -1 to Water Jug Problem

(0, 0) – Start State
(0, 3) – Rule 2, Fill the 3-liter jug
(3, 0) – Rule 9, Pour all the water from the 3-liter jug into the 4-liter jug.
(3, 3) – Rule 2, Fill the 3-liter jug
(4, 2) – Rule 7, Pour water from the 3-liter jug into the 4-liter jug until the 4-liter jug is full.
(0, 2) – Rule 5, Empty the 4-liter jug on the ground
(2, 0) – Rule 9, Pour all the water from the 3-liter jug into the 4-liter jug.

Goal State reached.

Water Jug Problem (Contd..)

The solution -2 to Water Jug Problem

(0, 0) – Start State
(4, 0) – Rule 1, Fill the 4-liter jug
(1, 3) – Rule 8, Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.
(1, 0) – Rule 6, Empty the 3-liter jug on the ground
(0, 1) – Rule 10, Pour all the water from the 4-liter jug into the 3-liter jug.
(4, 1) – Rule 1, Fill the 4-liter jug
(2, 3) – Rule 8, Pour water from the 4-liter jug into the 3-liter jug until the 3-liter jug is full.

Goal State reached.

Try

Q1:

"You are given two jugs, a 2-liter one and a 3-liter one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 1 litter of water into a 2-liters jug."

i.e, Start state is (0,0) , goal state is (1,n) for any value of n <=3. Write the production rules and find the solution.

Try

Q2:

We have 2 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.

i.e, Start state is (0,0) , goal state is (4,n) for any value of n <=3.

Write the production rules and find the solution.

2. Missionaries and cannibals problem

Missionaries and cannibals problem

Problem Definition : Missionaries and cannibals problem

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 person. How should they use this boat to cross the river in such a way that cannibals never outnumber missionaries on either of the river? If the cannibals ever outnumber the missionaries (on either bank) then the missionaries will be eaten. How can they all cross over without anyone being eaten?

Missionaries and cannibals problem

Problem Definition of Missionaries and cannibals problem

That means 3 missionaries and 3 cannibals were on one side of the river.

- All wants to cross the river.
- On same side of river missionaries count can’t be less than cannibals.
- Only one boat is available which can hold only one / two people at a time.

Missionaries and cannibals problem

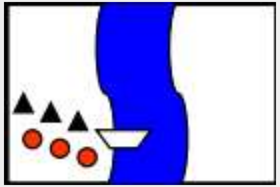
Here,

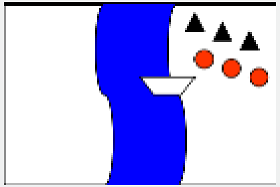
Missionaries → Human being

Cannibals → Human eater (who eat human / missionaries)

Number of boat = 1 (B)

#**Goal**: MMMCCC----- B ---MMMCCC





Missionaries and cannibals problem

#Constraints:

1. At most 2 person / 1 person can travel either side.

2. Boat can’t cross the river without any people.

3. No. of Cannibals should not be more than the no. of missionaries at either end of the river, otherwise cannibals will eat missionaries. (both side of the bank of river)

i.e. CCCM , CCM (Not possible)

CM, MMMC, MMC (Possible)

Missionaries and cannibals problem

State Space Representation:

States: A state consists of an ordered sequence of three numbers representing the number of missionaries, cannibals, and boats on the bank of the river from which they started. Thus, the start state is **(3,3,1)**

Operators: From each state the possible operators are to take either *one missionary, one cannibal, two missionaries, two cannibals, or one of each across* in the boat. Thus, there are at most *five operators*, although most states have fewer because it is necessary to avoid illegal states.

Missionaries and cannibals problem

Goal test: reached state **(3,3,2)**

Path cost: number of crossings. (step cost: 1 for each crossing , number of crossings = length of path)

Missionaries and cannibals problem

Solution:

		LEFT BANK		RIGHT BANK
0	Initial setup:	MMMCCC	B	-
1	Two cannibals cross over:	MMMC	B	CC
2	One comes back:	MMMCC	B	C
3	Two cannibals go over again:	MMM	B	CCC
4	One comes back:	MMMC	B	CC
5	Two missionaries cross:	MC	B	MMCC
6	A missionary & 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

cost: 11

Thank You

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Problem Characteristics

Presenter’s Name – Ms. Bidyutlata Sahoo
Presenter’s ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 03
Presentation Date – 11/09/2024

Course Outcome

At the end of the course, students should be able to:

CO2: Relate appropriate problem-solving methods to optimize the search results.

Topic Learning Outcome

Identify the strategies will be most efficient and effective in solving the problem.

Problem
and
it’s Characteristics

What is a Problem?

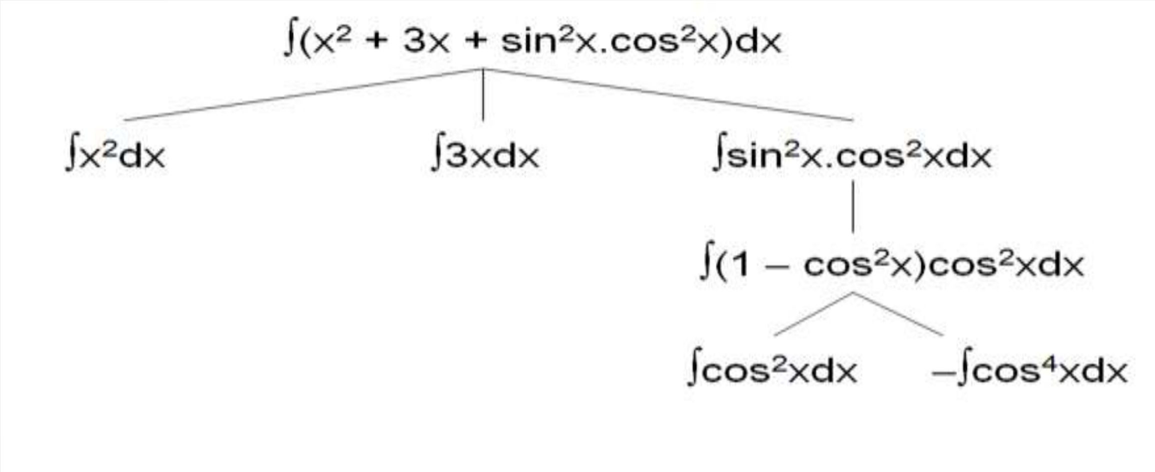
- A ***problem*** is a specific task or challenge that requires finding a solution or making a decision.
- A problem may have different aspects of representation and explanation.
- In order ***to choose an appropriate method for a particular problem*** first we need to ***categorize the problem*** based on the following characteristics.

Problem Characteristics

1. Is the problem decomposable into small sub-problems which are easy to solve?
2. Can solution steps be ignored or undone?
3. Is the universe of the problem is predictable?
4. Is a good solution to the problem is absolute or relative?
5. Is the solution to the problem a state or a path?
6. What is the role of knowledge in solving a problem using artificial intelligence?
7. Does the task of solving a problem require human interaction?

1. Is the problem decomposable into small sub-problems which are easy to solve?

- The ***decomposable problem can be solved easily.***
- **<u>Example</u>**: In the below case, the problem is divided into smaller problems. The smaller problems are solved independently. Finally, the result is merged to get the final result.



<u>Note</u>: But Water jug problem is non decomposable.

2. Can solution steps be ignored or undone?

Solution steps can ***be ignorable, recoverable and irrecoverable***.

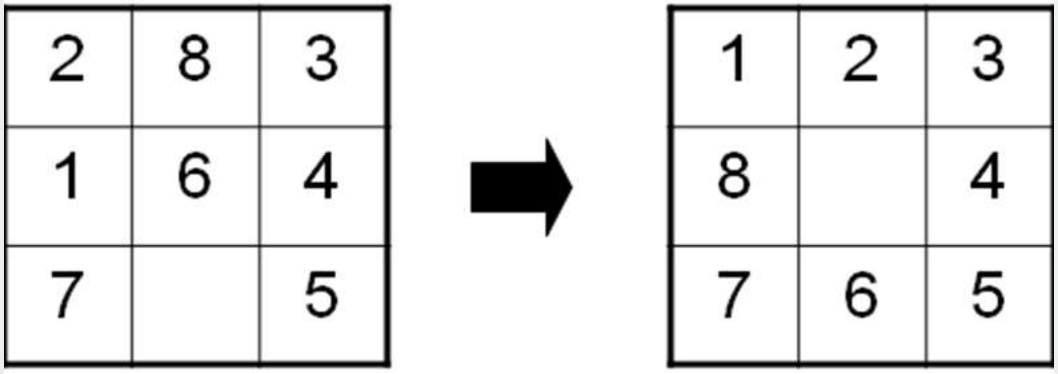
Ignorable: (can be solved by simple control strategy)

- Here we can ignore the solution step.
- If we don’t find any solution, every time we start with initial step.
- It is used in ***theorem proving problem***.
- Can be solved using a simple control structure that never backtracks.

2. Can solution steps be ignored or undone? (Contd..)

Recoverable: (can be solved by backtracking)

- Problems where solution steps can be undone.
- **Ex**: Used in ***water-jug problem, 8-puzzle problem***, Moves can be undone and backtracked.



2. Can solution steps be ignored or undone? (Contd..)

Irrecoverable: (can be solved by recoverable style methods via planning.

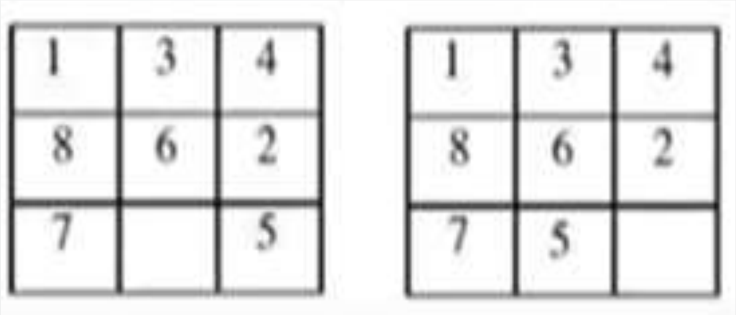
- Problems where solution steps can't be undone.
- i.e., with the current existing solution will try to find better out of it.
- Ex: Used in ***Snake and Ladder, Chess, Tic-tac-toe*** problem.

3. Is the universe of the problem is predictable?

- There are ***2 types of problem in this universe.***

1. For Certain Outcome Problem

- ***Planning*** can be used to generate a sequence of operators that is guaranteed to lead to a solution.
- Outcome is fixed.
- Destination is known.
- Predictable.



- Ex: ***8-puzzle problem, water-jug problem****….*etc.

3. Is the universe of the problem is predictable? (Contd..)

2. For Uncertain Outcome Probelm

- A sequence of generated operators can only have a good probability of leading to a solution. Plan revision is made as the plan is carried out and the necessary feedback is provided.
- ***Outcome is not fixed.***
- ***Not predictable***.
- ***Dealing with environment.***
- Ex: ***Chess game, Tic-tac-toe game, Card game***…..etc.

4. Is a good solution to the problem is absolute or relative?

There are 2 types of solutions to a problem:

1. Absolute Solution

- Focusing on getting the ***optimal solution.***
- Best path problem.
- Ex: CGPA system

2. Relative Solution

- Focusing on ***getting solution.***
- Any path problem.
- Ex: Percentage System.

5. Is the solution to the problem a state or a path?

Path:

- In which ***we know our destination / goal.***
- **Ex**: ***Water-jug problem*** (the path that leads to the goal must be reported.)
- **Ex**: In the ***missionaries and cannibals problem***, if we organise the various states in the form of a tree, it can be seen that the solution to the problem is a path connecting the various states.

5. Is the solution to the problem a state or a path? (Contd..)

State:

- In which ***we don't know our destination.***
- Looking at each state to proceed.
- Ex: ***Chess, Tic-Tac-Toe game*** etc.

6. What is the role of knowledge in solving a problem using artificial intelligence?

Knowledge could be in the ***form of rules and facts*** which helps to generate search space for finding the solution. There are 2 types of knowledge.

1. Fixed Knowledge:

- Here knowledge is fixed.
- Ex: Chess game

2. Updated Knowledge

- Here every time knowledge gets updated.
- Ex: Elections, AI media systems in terms of AI.

7. Does the task of solving a problem require human interaction?

- Sometimes it is useful to program computers to solve problems in ways that the majority of people would not be able to understand.
- This is fine if the level of the interaction between the computer and its human users is ***problem-in solution-out***.
- But increasingly we are building programs that require intermediate interaction with people, both to provide additional input to the program and to provide additional reassurance to the user.

7. Does the task of solving a problem require human interaction?

There are two types of tasks:

1. Solitary problem / task

- There is no interaction between machine and the user.
- Used in theorem proving.

***2.* Conversational problem / task**

- There is an intermediate interaction between machine and the user.
- Used in medical diagnosis.

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You

IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

Course Title - Logic Programming for Artificial Intelligence

Topic Title – Issues in the Design of Search Program

Presenter's Name – Ms. Bidyutlata Sahoo
Presenter's ID – IARE11028
Department Name – CSE (AI & ML)
Lecture Number - 04
Presentation Date -

Course Outcome

At the end of the course, students should be able to:

CO2: Relate appropriate problem-solving methods to optimize the search results.

Topic Learning Outcome

Understand different search algorithms and their roles for problem-solving.

Issues
In the Design of
Search Programs

Issues in the Design of Search Programs

- ***The direction in which to conduct the search*** (forward versus backward reasoning). If the search proceeds from start state towards a goal state, it is a forward search or we can also search from the goal.

- ***How to select applicable rules (Matching).*** Production systems typically spend most of their time looking for rules to apply. So, it is critical to have efficient procedures for matching rules against states.

Issues in the Design of Search Programs (Contd..)

- ***How to represent each node of the search process (knowledge representation problem).*** In games, an array suffices; in other problems, more complex data structures are needed. In water jug problem we can’t use a graph or tree.

IARE
INSTITUTE OF
AERONAUTICAL ENGINEERING

Thank You