Problem Solving: The Concept

- Problem solving is a process of generating solutions from observed data.
- A problem is characterized by:
  - a set of **goals**,
  - a set of **objects**
  - a set of **operations.**
- These could be ill-defined and may evolve during problem solving.
- Problem solving is a core aspect of artificial intelligence (AI) that mimics human cognitive processes. It involves identifying challenges, analyzing situations, and applying strategies to find effective solutions.

Problem Solving: The Concept

- Problem solving is fundamental to many AI-based applications.
- There are two types of problems:
- **COMPUTATION:**
  - The Problems like, **computation** of the sine of an angle or the square root of a value.
  - These can be solved through the use of deterministic procedure and the success is guaranteed.
  - In the real world, very few problems lend themselves to straightforward solutions.
- **SEARCHING:**
  - Most real world problems can be solved only by **searching** for a solution.
  - AI is concerned with these type of problems solving.

Problem Space: The Concept

- Problem space is an *abstract space*.
- A problem space encompasses all valid states that can be generated by the application of any combination of operators on any combination of objects.
- The problem space may contain one or more solutions.
- Solution is a combination of operations and objects that achieve the goals.

Defining Problem as State Space Search

- In solving problems, each possible ways of solving problem must be considered.
- Also, each possible combinations of factors influencing problem solving should be considered.
- Many problems can be formalised in a general way as search problems.

Analyzing Problem

- AI Problem can be is classified as:
  - Mundane/ General Tasks
  - Formal Tasks
  - Expert Tasks

Analyzing Problem

- **Mundane Tasks:**
  - These are the ones that the humans do on regular basis without any special training such as computer vision, speech recognition, Natural language processing, generation and translation etc.
  - Common sense, reasoning and planning are the common characteristics of these tasks.

- **Formal Tasks:**
  - These are the ones where there is an application of formal logic, some learning etc.
  - Verifications, Theorom proving etc are the common characteristics.
  - Games such as Chess Checkers, Go etc are classified in these task.

- **Expert Tasks:**
  - These are which comes under functional expert domain such as engineering, fault finding, manufacturing planning, medical diagnosis etc.
  - These Tasks which requires high analytical and thinking skills, a job only a professionals can do.

Isolating Task Oriented Knowledge

- Solving the problem, knowledge related to the field is required.
- As per the nature of AI applications, knowledge and there representation varies and so does the representation techniques.
- Various knowledge representation (KR) techniques are designed in this phase.

Finding the Solution

- After representation of problem and knowledge in suitable format, the appropriate methodology is chosen which uses knowledge and transforms the start goal to end goal.
- The techniques of finding the solution are called search techniques.

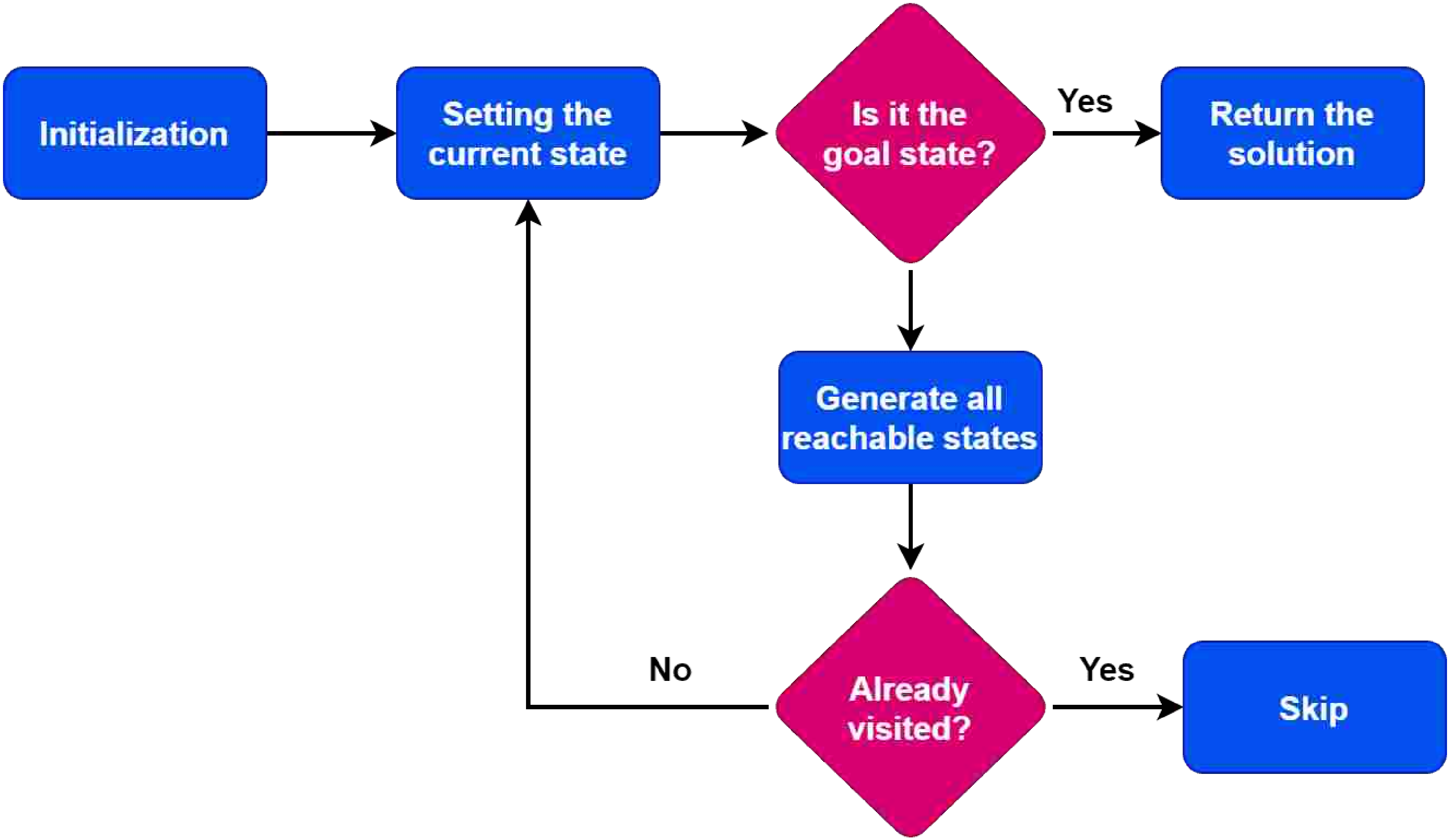
Representation of AI problem as State Space Search

- **State Space Search** is a problem-solving technique used in Artificial Intelligence (AI) to find the solution path from the initial state to the goal state by exploring the various states.
- The state space search approach searches through all possible states of a problem to find a solution.
- It is an essential part of Artificial Intelligence and is used in various applications, from game-playing algorithms to natural language processing.
- By modeling problems as states and transitions, AI systems can solve complex tasks like puzzle-solving, robotics, and planning through systematic search strategies that efficiently navigate toward the goal state.

Representation of AI problem as State Space Search

- A state space is a way to **mathematically represent** a problem by defining all the possible states in which the problem can be.
- This is used in search algorithms to represent the
  - **Initial State**
  - **Goal State**
  - **Current State** of the problem.
- Each state in the state space is represented using a set of variables.
- The efficiency of the search algorithm greatly depends on the size of the state space, and it is important to choose an appropriate representation and search strategy to search the state space efficiently.

State Space Search Algorithm



Initialization

Setting the current state

Is it the goal state?

Yes

Return the solution

Generate all reachable states

Already visited?

No

Yes

Skip

State Space Search Algorithm

- **Step 1: Define the State Space**
  - The first step in state space search is to define the state space clearly.
  - This involves identifying all possible states the system can be in and the transitions between these states.
  - For example, in the 8-puzzle problem, each configuration of the puzzle represents a state, and moving a tile constitutes a transition.
  - Defining the state space accurately is critical for setting up the search process.

State Space Search Algorithm

- **Step 2: Pick a Search Strategy**
  - The next step is to choose an appropriate search strategy.
  - The choice of strategy depends on the problem and the computational resources available.
  - There are two main categories:
    - **Uninformed Search**: are strategies like breadth-first search (BFS) and depth-first search (DFS) explore the state space without prior knowledge
    - **Informed Search**: are strategies like A* use heuristics to guide the search toward the goal state.

State Space Search Algorithm

- **Step 3: Start the Search**
  - Once the state space and search strategy are defined, the search begins from the initial state.
  - The system starts exploring the possible states based on the selected search strategy.
  - The search can either proceed by expanding all nodes at each level (in BFS) or exploring a path to its depth (in DFS).

- **Step 4: Extend the Nodes**
  - During the search, each node is expanded by generating its successor states.
  - These successor states represent the next possible configurations after an action is taken.
  - The search process continues by adding these nodes to the frontier or open list, which tracks all unexplored nodes.

State Space Search Algorithm

- **Step 5: Address State Repetition**
  - In large state spaces, it is common to encounter repeated states.
  - To avoid exploring the same state multiple times, it is crucial to maintain a closed list of visited nodes.
  - This prevents the search from wasting resources on redundant paths and ensures a more efficient search process.

- **Step 6: End the Search**
  - The search concludes when the goal state is reached or when all possible states have been explored.
  - If the search successfully finds the goal state, it returns the path that led to the solution.
  - Otherwise, the system may determine that no solution exists within the defined state space.

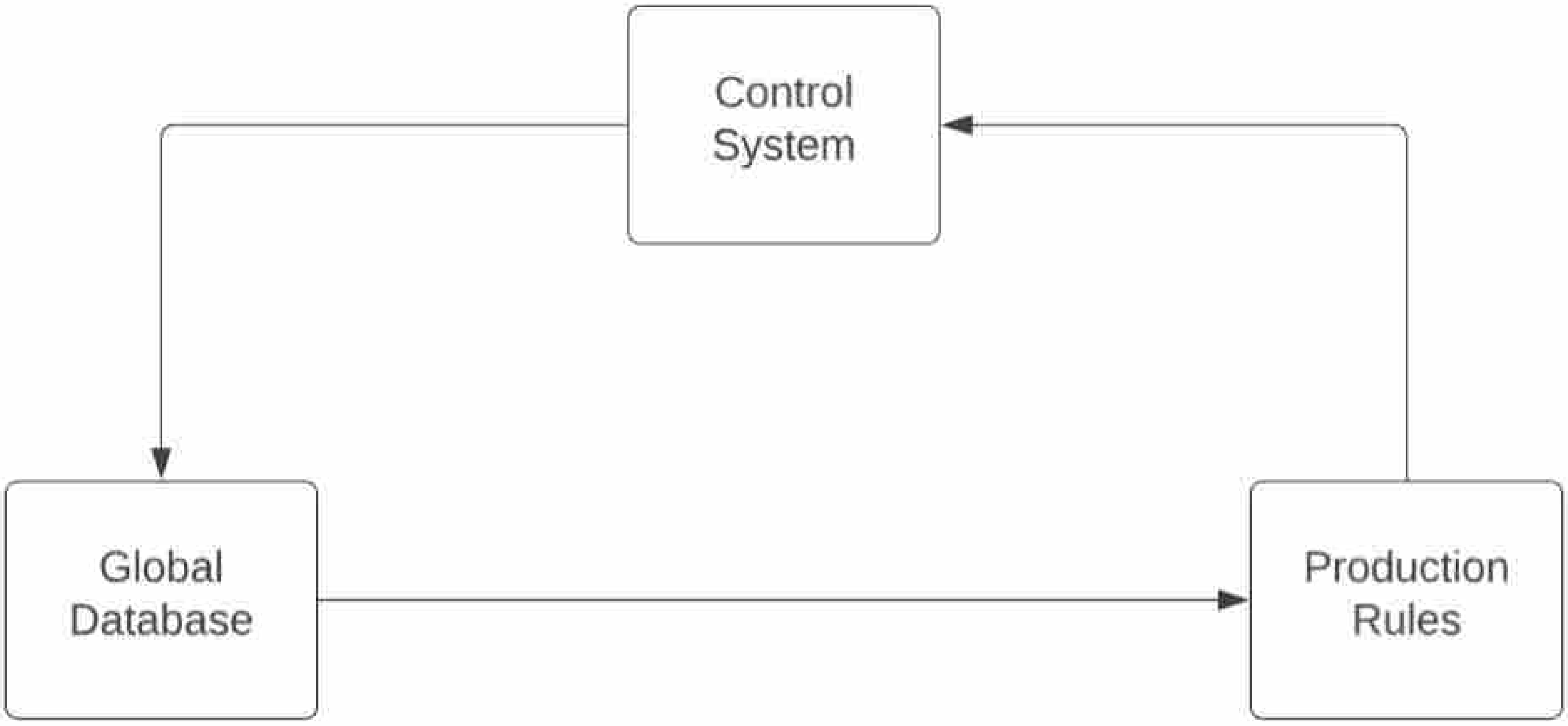
State Space Search Algorithm

- To define a problem as a state space search, you need to identify the following components:
- **State Space**: The set of all possible states of the problem. This includes the initial state, goal state(s), and all the intermediate states.
- **Initial State**: The starting state of the problem.
- **Goal State(s)**: One or more states that represent the desired outcome of the problem.
- **Successor Function**: A function that generates all the possible states that can be reached from the current state by applying an action.
- **Action(s)**: The set of all possible actions that can be taken from a given state.
- **Cost Function**: A function that assigns a cost to each action

Production System

- A production system is based on a set of rules about behavior.
- The production systems in artificial intelligence are rules applied to different behaviors and environment.
- A production system refers to a type of rule-based system that is designed to provide a structured approach to problem solving and decision-making.
- This framework is particularly influential in the realm of expert systems, where it simulates human decision-making processes using a set of predefined rules and facts.
- **Production system or production rule system is a computer program typically used to provide some form of artificial intelligence, which consists primarily of a set of rules about behavior but it also includes the mechanism necessary to follow those rules as the system responds to states of the world.**

Components of Production System



Control System

Global Database

Production Rules

Components of Production System

- The major components of the Production System in Artificial Intelligence are:

- **Global Database**: The global database is the central data structure used by the production system in Artificial Intelligence.

- **Knowledge Base/ Set of Production Rules:** The production rules operate on the global database. Each rule usually has a precondition that is either satisfied or not by the global database. If the precondition is satisfied, the rule is usually be applied. The application of the rule changes the database.

- **A Control System:** The control system then chooses which applicable rule should be applied and ceases computation when a termination condition on the database is satisfied. If multiple rules are to fire at the same time, the control system resolves the conflicts.

Features of Production System

- **Simplicity**: The structure of each sentence in a production system is unique and uniform as they use the "IF-THEN" structure. This structure provides simplicity in knowledge representation. This feature of the production system improves the readability of production rules.

- **Modularity**: This means the production rule code the knowledge available in discrete pieces. Information can be treated as a collection of independent facts which may be added or deleted from the system with essentially no deleterious side effects.

- **Modifiability**: This means the facility for modifying rules. It allows the development of production rules in a skeletal form first and then it is accurate to suit a specific application.

- **Knowledge-intensive**: The knowledge base of the production system stores pure knowledge. This part does not contain any type of control or programming information. Each production rule is normally written as an English sentence; the problem of semantics is solved by the very structure of the representation.

Examples of AI Problems: Tic – Tac – Toe

- Tic-Tac-Toe is a classic game often used to illustrate concepts in Artificial Intelligence, especially state space search.

- The goal of the game is to get three marks in a row (either "X" or "O") on a 3x3 grid, either horizontally, vertically, or diagonally.

- Let's break down how Tic-Tac-Toe can be represented as a state space search problem.
- **Game Representation**
  - A Tic-Tac-Toe game can be represented as a state in the form of a **3 x 3 grid**.
  - Each cell in the grid can be:
    - Empty (denoted by "_")
    - X (player 1's mark)
    - O (player 2's mark)
- For example, the initial state of the game (empty board) looks like this:

```
_ _ _

_ _ _

_ _ _
```

Examples of AI Problems

- **State Space**

- The state space of Tic-Tac-Toe is the set of **all possible board configurations** that can be achieved during the course of the game.

- The game involves alternating moves between the two players, "X" and "O", until one player wins or the game ends in a draw.

  - Initial State: The game starts with an empty board.

  - Goal States: The goal is to find a configuration where either "X" or "O" has three marks in a row (horizontally, vertically, or diagonally).

- **Terminal States**: The game ends in either a win for one player or a draw (if no more moves are possible and no one has won).

- **Moves**: Players take turns placing their mark ("X" or "O") in an empty cell.

Examples of AI Problems

- **State Representation**
- Each state in the state space represents a specific configuration of the board. For example:
  - Initial State:

    _ _ _

    _ _ _

    _ _ _
  - State after Player 1 (X) makes a move:

    X _ _

    _ _ _

    _ _ _
- State after Player 2 (O) makes a move:

  X _ _

  _ O _

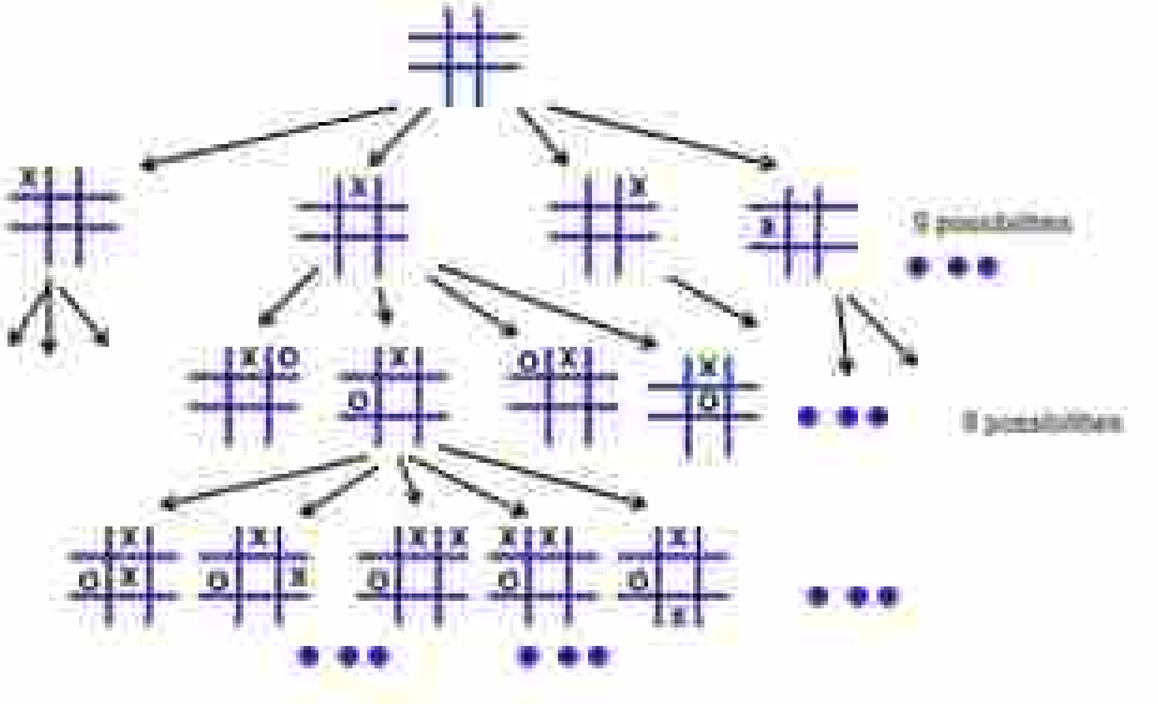
  _ _ _

Examples of AI Problems

- **Actions**
  - At any point in the game, the action is the move made by the current player.
  - The actions are the placement of "X" or "O" in one of the empty cells.
  - For instance, if it's "X"'s turn, the possible actions are the set of empty cells where "X" can be placed.

Examples of AI Problems

- **Actions**
  - At any point in the game, the action is the move made by the current player.
  - The actions are the placement of "X" or "O" in one of the empty cells.
  - For instance, if it’s "X"'s turn, the possible actions are the set of empty cells where "X" can be placed.



Examples of AI Problems

- Tic – Tac – Toe
- Water – Jug Problem
- Tower of Hanoi
- Traveling Salesman Problem