PROBLEM-SOLVING TECHNIQUES

Chapter 2

PROBLEM STATE SPACES

To Build a System to Solve a Problem

- Define the problem precisely
  - Definition must include precise specifications of what the initial situation (s) will be as well as what final situations constitute acceptable solutions to the problem.

To Build a System to Solve a Problem

- Define the problem precisely
  - Definition must include precise specifications of what the initial situation (s) will be as well as what final situations constitute acceptable solutions to the problem.
- Analyze the problem
  - A very few important features can have an immense effect on the appropriateness of various possible techniques to solve the problem.

To Build a System to Solve a Problem

- Define the problem precisely
  - Definition must include precise specifications of what the initial situation (s) will be as well as what final situations constitute acceptable solutions to the problem.
- Analyse the problem
  - A very few important features can have an immense effect on the appropriateness of various possible techniques to solve the problem.
- Isolate and represent task knowledge that is necessary to solve the problem.

To Build a System to Solve a Problem

- Define the problem precisely
  - Definition must include precise specifications of what the initial situation (s) will be as well as what final situations constitute acceptable solutions to the problem.
- Analyze the problem
  - A very few important features can have an immense effect on the appropriateness of various possible techniques to solve the problem.
- Isolate and represent task knowledge that is necessary to solve the problem.
- Choose the best problem-solving technique (s) and apply it (them) to the particular problem.

State Space Search

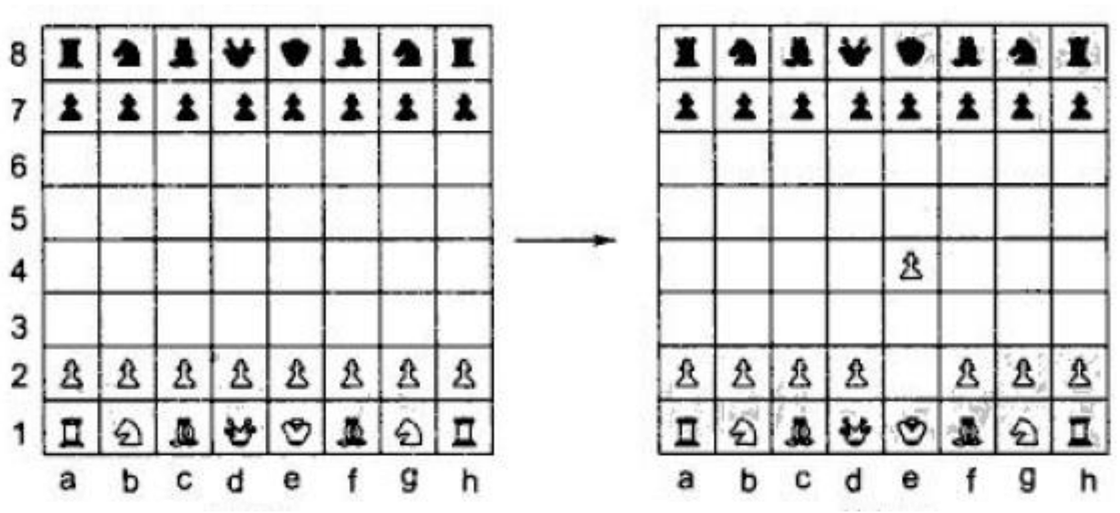
- Problem solving = Searching for a goal state

State Space Search: Playing Chess

- Each state can be described by an 8-by-8 array.
- Initial state is the game opening position.
- Goal state is any position in which the opponent does not have a legal move and his or her king is under attack.
- Legal moves can be described by a set of rules:
  - Left sides are matched against the current state.
  - Right sides describe the new resulting state.

State Space Search: Playing Chess

- A legal chess move



State Space Search: Playing Chess

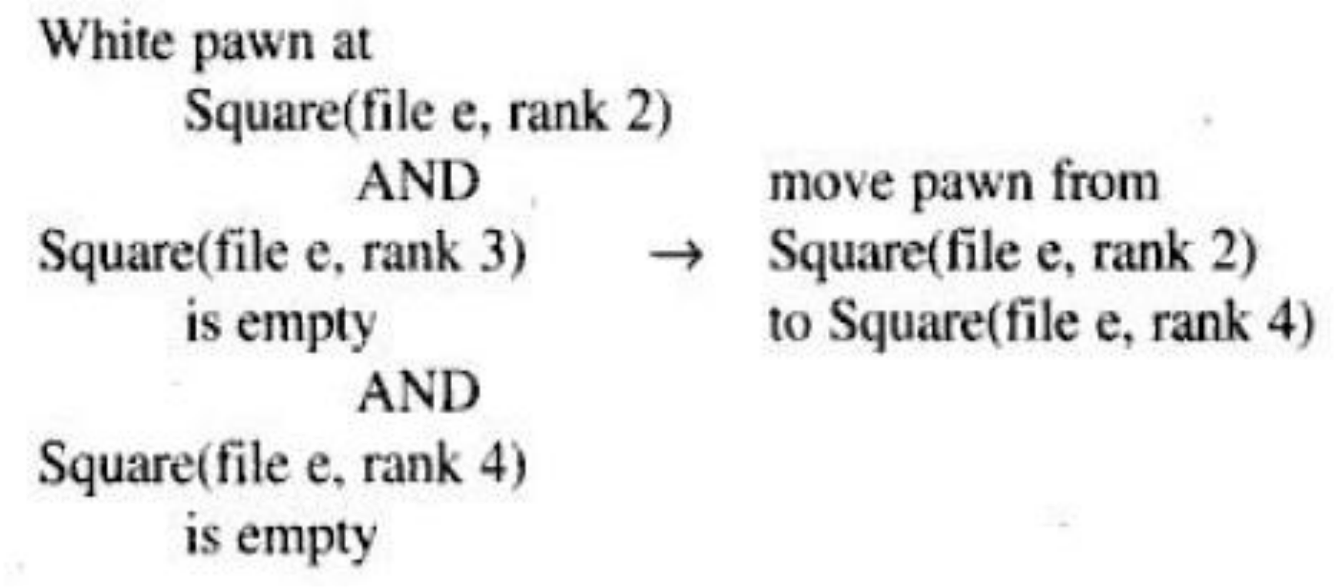
- State space is a set of legal positions.
- Starting at the initial state.
- Using the set of rules to move from one state to another.
- Attempting to end up in a goal state.

State Space Search: Playing Chess

- If we write the rules like the one above, we have to write a very large number of them since there has to be a separate rule for each of the roughly 10^{120} possible board positions.
  - No person could ever supply a complete set of such rules. It would take too long and could certainly not be done without mistakes.
  - No program could easily handle all those rules.

State Space Search: Playing Chess

- In order to minimize such problems, we should look for a way to write rules describing the legal moves in as general way as possible.



```
White pawn at
      Square(file e, rank 2)
              AND                    move pawn from
Square(file e, rank 3)        →     Square(file e, rank 2)
      is empty                       to Square(file e, rank 4)
              AND
Square(file e, rank 4)
      is empty
```

State Space Search: Water Jug Problem

- “You are given two jugs, a 4-litre one and a 3- litre 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 litres of water into 4-litre jug.”

State Space Search: Water Jug Problem

- Problem state space: (x, y)
  - x = 0, 1, 2, 3, or 4
  - y = 0, 1, 2, 3
- Start state: (0, 0).
- Goal state: (2, n) for any n.
- Attempting to end up in a goal state.

State Space Search: Water Jug Problem

1. $(x, y) \rightarrow (4, y)$

   if $x < 4$

2. $(x, y) \rightarrow (x, 3)$

   if $y < 3$

3. $(x, y) \rightarrow (x - d, y)$

   if $x > 0$

4. $(x, y) \rightarrow (x, y - d)$

   if $y > 0$

5. $(x, y) \rightarrow (0, y)$

   if $x > 0$

6. $(x, y) \rightarrow (x, 0)$

   if $y > 0$

7. $(x, y) \rightarrow (4, y - (4 - x))$

   if $x + y \geq 4, y > 0$

8. $(x, y) \rightarrow (x - (3 - y), 3)$

   if $x + y \geq 3, x > 0$

State Space Search: Water Jug Problem

9. $(x, y) \rightarrow (x + y, 0)$

    if $x + y \leq 4, y > 0$

10. $(x, y) \rightarrow (0, x + y)$

    if $x + y \leq 3, x > 0$

11. $(0, 2) \rightarrow (2, 0)$
12. $(2, y) \rightarrow (0, y)$

State Space Search: Water Jug Problem

1. Current state = (0, 0)

2. Loop until reaching the goal state (2, 0)
   - Apply a rule whose left side matches the current state
   - Set the new current state to be the resulting state

$(0, 0) \rightarrow (0, 3) \rightarrow (3, 0) \rightarrow (3, 3) \rightarrow (4, 2) \rightarrow (0, 2) \rightarrow (2, 0)$

State Space Search: Water Jug Problem

- The role of the condition in the left side of a rule
  - $\Rightarrow$ restrict the application of the rule
  - $\Rightarrow$ more efficient

1. $(x, y) \rightarrow (4, y)$

   if $x < 4$

2. $(x, y) \rightarrow (x, 3)$

   if $y < 3$

State Space Search: Water Jug Problem

- Special-purpose rules to capture special-case knowledge that can be used at some stage in solving a problem

11. $(0, 2) \rightarrow (2, 0)$

12. $(2, y) \rightarrow (0, y)$

State Space Search: Summary

- Define a state space that contains all the possible configurations of the relevant objects.
- Specify the initial states.
- Specify the goal states.
- Specify a set of rules:
  - What are unstated assumptions?
  - How general should the rules be?
  - How much knowledge for solutions should be in the rules?

PROBLEM CHARACTERISTICS

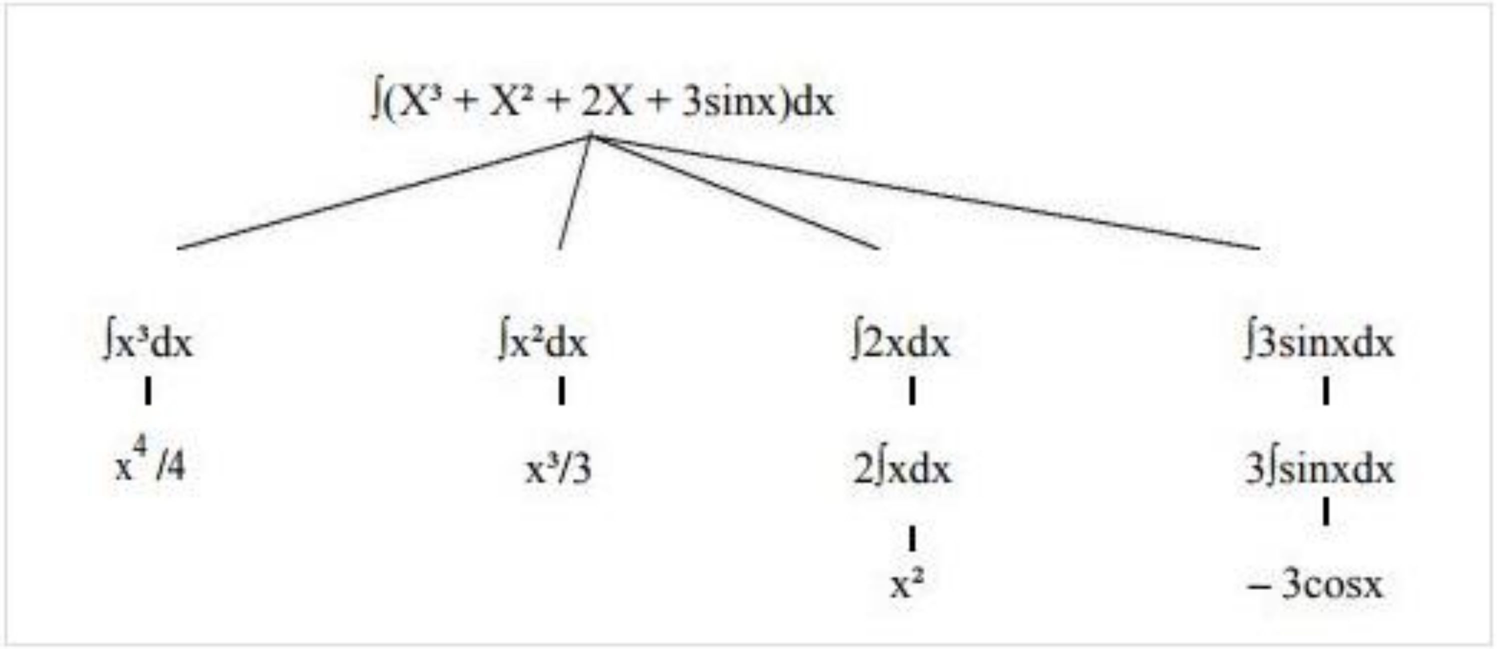
Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).

Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?

Is the problem decomposable?



Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?

Can the solution step be ignored or undone?

- **Ignorable :** In which solution steps can be ignored (e.g., Theorem Proving).
- Suppose we are trying to prove a mathematical theorem.
- First we proceed with proving a lemma that we think will be useful.
- Later we realize that it is not useful.
- So, Are we in serious trouble?
- No, still we can prove the theorem, only we need to ignore the first approach and start with another one to prove the theorem.

Can the solution step be ignored or undone?

- **Recoverable :** In which solution steps can be undone. (e.g., 8-Puzzle Problem).
- While attempting to solve the 8-puzzle problem, mistakenly we make a wrong move and realize that mistake. Now to correct our mistake we need to undo incorrect steps.
- To undo incorrect steps the control mechanism for an 8-puzzle solver must keep track of the order in which steps are performed, so that we can backtrack to the initial state and start with some correct move.

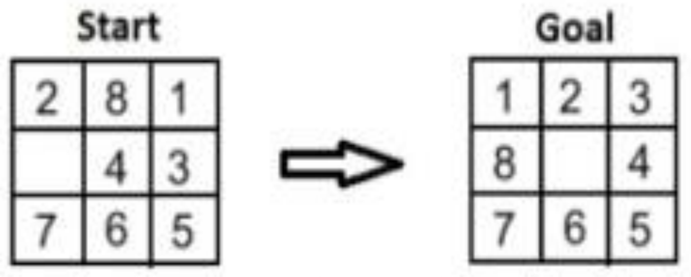


Figure : 8 Puzzle Problem

Can the solution step be ignored or undone?

- **Irrecoverable :** In which solution steps cannot be undone. (e.g., Chess).
- Suppose playing chess program makes a wrong move and realize it after couple of moves.
- It cannot simply ignore the mistake.
- Neither it can be undone its move and start the game again.
- Here, once we make a move we never recover from that step.
- Only we can try to give the best of the current situation.

Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?
  - Is the problem universally predictable?

Is the problem universally predictable?

- Consider that we are playing with the 8-Puzzle problem.
  - Every time we make a move, we know exactly what will happen.
  - This means that it is possible to plan an entire sequence of moves and be confident what the resulting state will be.
  - We can backtrack to previous moves if they prove wrong.
- Suppose we want to play Bridge.
  - We need to make some decisions like which card to play on the first trick.
  - Then we need to make a plan for the entire hand before the first play, but we cannot plan and play with certainty since we cannot know where all the cards are or what the other players will do on their turns.
  - So we have to investigate several plans and use probability of various outcomes to choose best plan for a good score.
  - So, the outcome of this game is very uncertain.

Is the problem universally predictable?

- These two games shows us difference between **certain outcome** (e.g. 8-Puzzle) and **uncertain outcome** (e.g. play bridge) .
- In case of **certain outcome,** we can make a plan to generate a sequence of operation that guaranteed to lead to a solution. So, that the outcome is very certain.
- In case of **uncertain outcome** problems, we follow the process of plan revision as the plan is carried out and the necessary feedback is provided. The disadvantage is that the planning in this case is often very expensive.

Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?
  - Is the problem universally predictable?
  - Is a good solution to the problem obvious without comparison to all the possible solutions?

Is a good solution absolute or relative?

- Suppose we have the problem of answering questions based on the simple facts, such as following :
  1. James was a man.
  2. James was a Pompeian.
  3. James was born in 50 A.D
  4. All men are mortal.
  5. All Pompeians died when volcano erupted in 79 A.D.
  6. No mortals lives longer than 150 years.
  7. It is now 2018 A.D.
- Consider we ask the question “Is James alive?”.
- By going through each of these facts, we can easily find the answer to the question.
- Suppose, two paths leads to the answer but we are interested in is the answer to the question, it does not matter which path we follow.
- But now consider again the traveling salesman problem. In that, we need to find the best solution (shortest path) among all of other solution.

Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?
  - Is the problem universally predictable?
  - Is a good solution to the problem obvious without comparison to all the possible solutions?
  - Is the desired solution a state of the world or a path to a state?

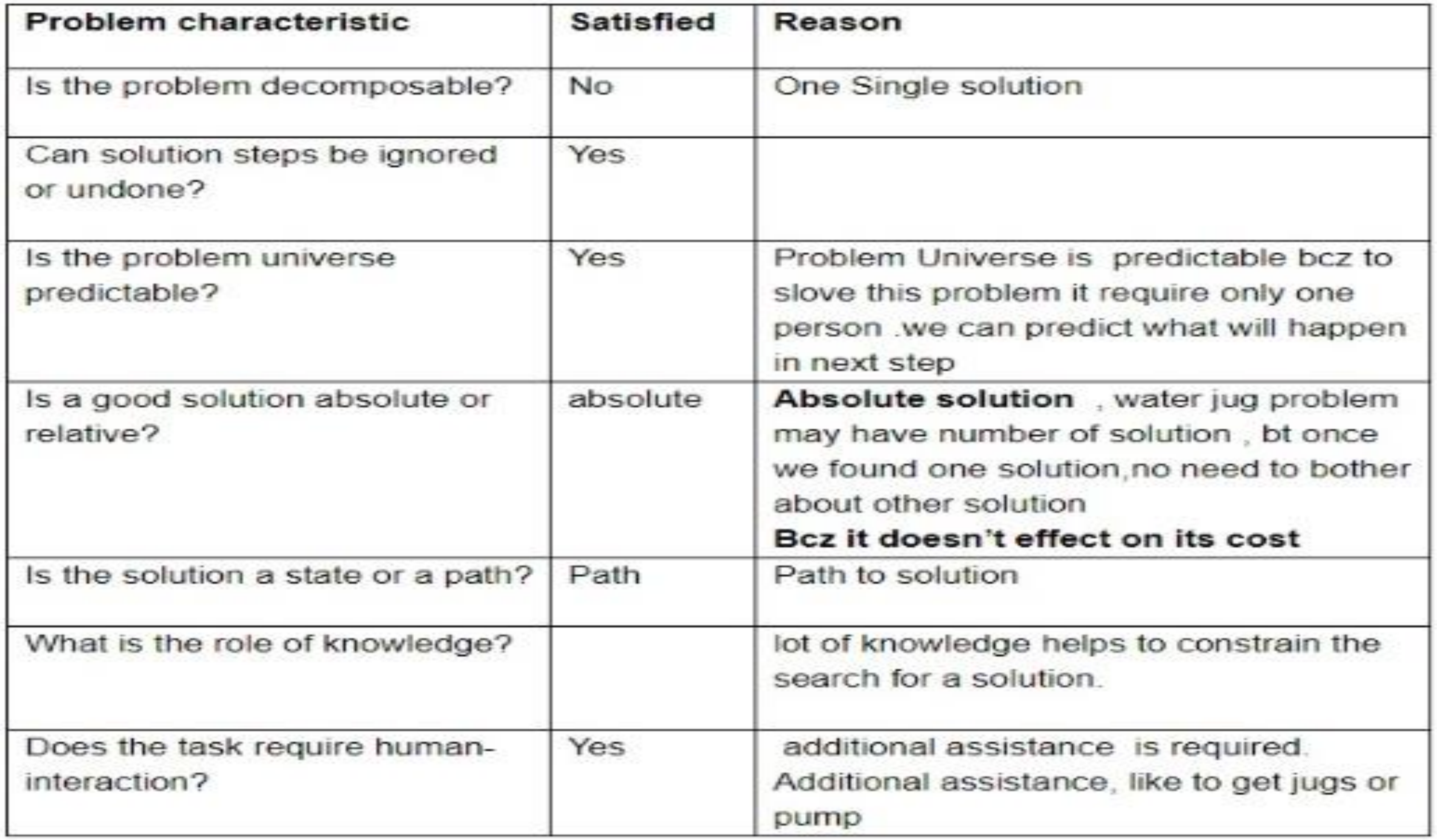
Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?
  - Is the problem universally predictable?
  - Is a good solution to the problem obvious without comparison to all the possible solutions?
  - Is the desired solution a state of world or a path to a state?
  - Is a large amount of knowledge absolutely required to solve the problem?

Problem Characteristics

- A problem may have different aspects of representation and explanation. In order to choose the most appropriate method for a particular problem, it is necessary to analyze the problem along several key dimensions (7 AI problem characteristics).
  - Is the problem decomposable into set of sub problems?
  - Can the solution step be ignored or undone?
  - Is the problem universally predictable?
  - Is a good solution to the problem obvious without comparison to all the possible solutions?
  - Is the desire solution a state of world or a path to a state?
  - Is a large amount of knowledge absolutely required to solve the problem?
  - Will the solution of the problem require interaction between the computer and the person?

Problem Characteristics – Water Jug Problem



Problem characteristic	**Satisfied**	**Reason**
Is the problem decomposable?	No	One Single solution
Can solution steps be ignored or undone?	Yes	
Is the problem universe predictable?	Yes	Problem Universe is predictable bcz to slove this problem it require only one person .we can predict what will happen in next step
Is a good solution absolute or relative?	absolute	**Absolute solution** , water jug problem may have number of solution , bt once we found one solution,no need to bother about other solution **Bcz it doesn't effect on its cost**
Is the solution a state or a path?	Path	Path to solution
What is the role of knowledge?		lot of knowledge helps to constrain the search for a solution.
Does the task require human-interaction?	Yes	additional assistance is required. Additional assistance, like to get jugs or pump

PRODUCTION SYSTEM

Production System

- The production system is a model of computation that can be applied to implement search algorithms and model human problem solving.
- Such problem solving knowledge can be packed up in the form of little quanta called productions.
- A production is a rule consisting of a situation recognition part and an action part.
  - A production is a situation-action pair in which the left side is a list of things to watch for and the right side is a list of things to do so.
- Production systems may be called premise-conclusion pairs rather than situation action pair.

Components of a Production System

- A set of production rules, which are of the form A → B.
  - Each rule consists of left hand side constituent that represent the current problem state and a right hand side that represent an output state.
  - A rule is applicable if its left hand side matches with the current problem state.

Components of a Production System

- A set of production rules, which are of the form A → B.
  - Each rule consists of left hand side constituent that represent the current problem state and a right hand side that represent an output state.
  - A rule is applicable if its left hand side matches with the current problem state.
- A knowledge database, which contains all the appropriate information for the particular task.
  - Some part of the database may be permanent while some part of this may pertain only to the solution of the current problem.

Components of a Production System

- A set of production rules, which are of the form A → B.
  - Each rule consists of left hand side constituent that represent the current problem state and a right hand side that represent an output state.
  - A rule is applicable if its left hand side matches with the current problem state.
- A knowledge database, which contains all the appropriate information for the particular task.
  - Some part of the database may be permanent while some part of this may pertain only to the solution of the current problem.
- A control strategy that specifies order in which the rules will be compared to the database of rules and a way of resolving the conflicts that arise when several rules match simultaneously.

Components of a Production System

- A set of production rules, which are of the form A → B.
  - Each rule consists of left hand side constituent that represent the current problem state and a right hand side that represent an output state.
  - A rule is applicable if its left hand side matches with the current problem state.
- A knowledge database, which contains all the appropriate information for the particular task.
  - Some part of the database may be permanent while some part of this may pertain only to the solution of the current problem.
- A control strategy that specifies order in which the rules will be compared to the database of rules and a way of resolving the conflicts that arise when several rules match simultaneously.
- A rule applier, which checks the capability of rule by matching the current state with the left hand side of the rule and finds the appropriate rule from database of rules.

Production System Rules

- We can represent knowledge in a production system as a set of rules of the form:
  - If (condition) THEN (action)
- along with a control system and a database.
- The control system serves as a rule interpreter and sequencer.
- The database acts as a context buffer, which records the conditions evaluated by the rules and information on which the rules act.
- The production rules are also known as condition – action, antecedent - consequent, pattern – action, situation – response, feedback – result pairs.

Example

- If (you have an exam tomorrow)
- THEN (study the whole night)

Features of Production System

- **Simplicity:** The structure of each sentence in a production system is unique and uniform as they use “IF-THEN” structure.

Features of Production System

- **Simplicity:** The structure of each sentence in a production system is unique and uniform as they use “IF-THEN” structure.
- **Modularity:** This means 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.

Features of Production System

- **Simplicity:** The structure of each sentence in a production system is unique and uniform as they use “IF-THEN” structure.
- **Modularity:** This means 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 of modifying rules. It allows the development of production rules in a skeletal form first and then it is accurate to suit a specific application.

Features of Production System

- **Simplicity:** The structure of each sentence in a production system is unique and uniform as they use “IF-THEN” structure.
- **Modularity:** This means 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 of 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 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.

Disadvantages of Production System

- **Opacity:** This problem is generated by the combination of production rules. The opacity is generated because of less prioritization of rules. More priority to a rule has the less opacity.

Disadvantages of Production System

- **Opacity:** This problem is generated by the combination of production rules. The opacity is generated because of less prioritization of rules. More priority to a rule has the less opacity.
- **Inefficiency:** During execution of a program, several rules may be active. As the rules of the production system are large in number and they are hardly written in hierarchical manner, it requires some forms of complex search through all the production rules for each cycle of control program.

Disadvantages of Production System

- **Opacity:** This problem is generated by the combination of production rules. The opacity is generated because of less prioritization of rules. More priority to a rule has the less opacity.
- **Inefficiency:** During execution of a program, several rules may be active. As the rules of the production system are large in number and they are hardly written in hierarchical manner, it requires some forms of complex search through all the production rules for each cycle of control program.
- **Absence of learning:** Rule-based production systems do not store the result of the problem for future use. Hence, it does not exhibit any type of learning capabilities. So for each time for a particular problem, some new solutions may come.

Disadvantages of Production System

- **Opacity:** This problem is generated by the combination of production rules. The opacity is generated because of less prioritization of rules. More priority to a rule has the less opacity.
- **Inefficiency:** During execution of a program, several rules may be active. As the rules of the production system are large in number and they are hardly written in hierarchical manner, it requires some forms of complex search through all the production rules for each cycle of control program.
- **Absence of learning:** Rule-based production systems do not store the result of the problem for future use. Hence, it does not exhibit any type of learning capabilities. So for each time for a particular problem, some new solutions may come.
- **Conflict resolution:** The rules in a production system should not have any type of conflict operations. When a new rule is added to a database, it should ensure that it does not have any conflicts with the existing rules.

SEARCH SPACE CONTROL

Search Strategies

- A strategy is defined by picking the order of node expansion.
- Strategies are evaluated along the following dimensions:
  - Completeness: Does it always find a solution if one exists?
  - Time complexity: Number of nodes generated/expanded
  - Space complexity: Maximum number of nodes in memory
  - Optimality: Does it always find a least-cost solution?

Search Strategies

- Time and space complexity are measured in terms of:
  - b: maximum branching factor of the search tree (actions per state).
  - d: depth of the solution
  - m: maximum depth of the state space (may be ∞) (also noted sometimes D).
- Two kinds of search: Uninformed and Informed.

Search Space Control

- Uninformed search
  - Breadth first search (BFS)
  - Depth first search (DFS)
  - Depth first search with iterative deepening

- Heuristic search
  - Simple hill climbing
  - Steepest ascent hill climbing
  - A* algorithm
  - AO* algorithm
  - Minimax search procedure for game playing
  - Alpha beta cut-offs